# Selecting Suitable Software Effort Estimation Method

Duygu Deniz Erhan[1], Ayça Kolukısa Tarhan[2] [0000-0003-1466-9605], Rana Özakıncı[3] [0000-0002-7803-453X]

[1,2,3] Hacettepe University, Department of Computer Engineering, Ankara, TURKEY
[1]duygudeniz06@gmail.com,
{[2]atarhan,[3]ranaozakinci}@hacettepe.edu.tr

**Abstract.** Effort estimation is one of the important factors affecting the success of software projects. In order to support this, many effort estimation methods have been developed from past to present. The reliability of the effort estimation of a project depends on the choice of the most appropriate method for the project characteristics and the estimation context. Even if a good performing method is used, the estimation results may remain to be inaccurate if an appropriate estimation method is not selected as appropriate to the project context. In this study, we proposed an approach for selecting the most suitable estimation method for a software project by considering the project characteristics and the stakeholder needs. An expert-opinion survey was prepared based on the key features of the commonly used estimation methods that have been frequently referred to in literature. The expert-opinion survey was answered by experts who carried out scientific studies in the field of software effort estimation, and a decision matrix was created in the light of their opinions. Then, a questionnaire was built for eliciting information about project characteristics from an estimator who wants to carry out effort estimation for his/her project. With the decision matrix, the estimator can select the most suitable method for his/her estimation by answering the questionnaire. A sample study was conducted and the questionnaire was answered using the ISBSG data set. At the end, the appropriateness of the proposed approach was discussed.

**Keywords:** Effort Estimation, Software Effort, Estimation Method, Method Selection, Decision Matrix, Expert Opinion.

## 1 Introduction

Software effort estimation (SEE) is the process of predicting the amount of effort required to build a software system. For effective planning, effort and schedule estimation is required for a project. In order to provide this benefit, estimation process must be accurate and reliable but this is a difficult task. In order to address this problem, many estimation methods have been proposed by researchers and many of the proposed methods have been shown to give successful results.

Nevertheless, there is no estimation method that makes the most accurate estimates in all projects [1,2]. Estimation methods make successful estimations for projects that provide certain characteristics (organizational structure, type of project, development environment, etc.). It is stated that the most successful effort estimation method can change for a given data set because different criteria are used [3]. Since the estimation method that gives accurate results will change even for different projects within the same organization, the selection of methods on the basis of the organization may not give correct results. Although methods focusing on specific project features are continually proposed for more accurate estimates in literature [4,5], it is necessary to perform analyses regarding the attributes of method, project and environment each time and even to use expert knowledge for determining which method is suitable. An accurate effort estimation can be achieved only by selecting an estimation method best matched to the estimation context.

Several studies on selecting the suitable estimation method have been proposed by examining the project properties and data sets to be used [4,5]. Although these studies have shown that the success of estimation methods can change according to the project characteristics and data set, they do not propose a general method for different types of projects and environments. The existing selection methods are not feasible for a new project. Since the method is chosen according to the MMRE success of the estimations in old projects, the characteristics of the new project are not considered, so it may not be suitable for the new project.

In this study, project characteristics that affect the success of estimation methods were examined. For this purpose, an expert-opinion survey was prepared and the relationship between the project characteristics and estimation methods was studied. The survey was realized by referring to the knowledge of the experts having published scientific studies on software effort estimation. It was aimed to determine the most suitable estimation method for given project characteristics and stakeholder needs by using the data obtained from the expert-opinion survey and answers from estimator questions. Multiple Criteria Decision Analysis (MCDA) method was used while processing the data from the expert-opinion survey into a decision matrix. Then, in an example estimation scenario based on ISBSG dataset, a user was asked to answer a set of estimator questions prepared, and the most suitable estimation method was selected with the information of a new project to be estimated.

The rest of this paper is organized as follows. Sec 2 provides background and related work on software effort estimation, method selection and MCDA. Sec 3 explains the proposed evaluation approach, alternative estimation methods used in this study and method selection criteria. Also, it is exemplified how decision matrix values are formed over the answers to expert-opinion survey questions. Sec 4 presents example evaluation using ISBSG data set and related estimation assumptions, and explains the feasibility of the proposal. Sec 5 discusses the weaknesses in the proposed approach. Finally, Sec 6 concludes the paper with a summary of this study and plans for future.

## 2 Background and Related Work

### 2.1 Software Effort Estimation and Method Selection

Since effort estimation method selection is a major factor in estimation, there are many studies in literature that examine and classify methods. At the same time, due to the importance of the criteria that affect the decision in method selection, there are many studies that investigate the criteria of various parameters that affect the methods. We chose the methods and the criteria we used to prepare the expert-opinion survey based on the studies that we describe below.

Wen et al. [6] made a systematic literature review on machine learning (ML) based software development effort estimation models. They analyzed ML based models from four aspects: type of ML technique, estimation accuracy, method comparison, and estimation context. After reviews, the authors found that eight types of ML techniques are mostly used, including Case-Based Reasoning, Artificial Neural Networks, Decision Trees, Bayesian Networks, Support Vector Regression, Genetic Algorithms, Genetic Programming, Association Rules. They suggested that ML methods are usually more accurate than non-ML methods. Also, they listed the strengths and weakness of the ML techniques used in software effort estimation.

Jorgenson et al. [7] prepared a basis for the studies about improvement of software development cost estimation. They explored the question “What are the most investigated estimation methods and how this changed over time?” and as a result, they showed the distribution of articles on different estimation approaches per period and in total. Also, they make recommendations for future estimation researches.

Marco et al. [8] made a systematic review on software effort estimation methods and reported that the number of studies on the subject is increasing. They also prepared a list of the best performing methods and the most used methods. The most active and influential researchers were also shown in their paper. We invited these researchers to answer our expert-opinion survey.

Idri et al. [9] analyzed analogy-based SEE techniques according to criteria and the studies from some perspectives (estimation context, accuracy comparison, estimation accuracy etc.) and they found that more estimation techniques should be developed. They also said that accuracy in effort estimation depends on several categories of parameters. These parameters are: Dataset characteristics used (size, missing value, etc.), analogy process configuration (adaptation formula, feature selection, etc.), evaluation method used (n-fold cross validation, disagreement, etc.).

Bilgaiyan et al. [10] made a review on software cost estimation in agile software development. They prepared a study in which different estimation methods are required to be successful, and discussed the difficulties of the methods.

Shekhar et al. [11] made a comprehensive review on software effort estimation methods. They explained the working principles of many methods. In addition, by listing the advantages and disadvantages of the methods, they shed light on the desired and to be avoided situations in the use of the methods.

Chirra et al. [12] tabulated all the methods in software cost estimation based on their type, amount of data required, validation methods used by them, weaknesses and strengths. They discussed the detailed results about the methods from several perspectives, including: type (algorithmic method, learning oriented method, etc.), strengths, weakness, accuracy, data (limited, extensive, etc.), and validation (cross validation method, Jackknife method, etc).

There are also studies that associate SEE method selection with various criteria and want to structure it. We also overview these studies below.

In 2012, Sehra et al. [4] proposed a method for selecting an effort estimation method based on the environment and the project type by using Fuzzy Analytic Hierarchy Process. They used reliability, mean magnitude of relative error (MMRE), prediction (Pred), and uncertainty criteria as input for their method. Their selected decision alternatives are Expert Judgement, COCOMO, and Fuzzy Neural Network based effort estimation methods.

In 2017, Bansal et al. [5] proposed fuzzy weighted distance-based approximation (WDBA) to solve selecting an effort estimation method problem based on MCDA. They found that WDBA is more effective than other MCDA solutions due to the lack of complex matrix operations. They used magnitude of relative error (MRE), root mean square (RMS), prediction (Pred), root mean square error (RMSE), mean absolute relative error (MARE), variance absolute relative error (VARE), value accounted for (VAF), accuracy, reliability, uncertainty, and mean absolute error (MAE) as input to their method. They selected eleven algorithmic effort estimation methods as decision alternatives.

In 2015, Nayebi et al. [3] proposed an approach for selecting a machine learning effort estimation method for specific datasets. They selected nine machine learning methods as decision alternatives. They used prediction (Pred), correlation coefficient (CRR), and Bayesian Information Criterion (BIC) as input to their approach. They compared SEE methods based on these criteria by evaluating nine different datasets.

Ozakinci and Tarhan [13] aimed to identify software defect prediction methods in the early stages of the project, which would give the most accurate result in defect prediction. In this study, the authors determined the criteria based on the project, data, and method features considering the related studies in literature. Then, they sent a survey to the experts and asked them to evaluate the criteria against the prediction methods. At the end, using the MCDA tool, they prepared a questionnaire for users to choose the appropriate method for software defect prediction. Our study employed a similar approach as specific to software effort estimation.

### 2.2 Multi Criteria Decision Analysis (MCDA)

Multi-Criteria Decision Analysis is a structure used to resolve important and complex decision-making situations of decision makers [14]. MCDA is an "*umbrella term to describe a collection of formal approaches which seek to take explicit account of multiple criteria in helping individuals or groups explore decisions that matter*" [15].

Many MCDA methods have been proposed in literature. The most well-known of them are: AHP (Analytic Hierarchical Process) [16], TOPSIS (Ordering Simulation Technique in Ideal Solution) [17], PROMETHEE (The Preference Ranking Organization METHod for Enrichment of Evaluations) [18] and ELECTRE (Elimination and Choice Expressing the Reality) [19].

An MCDA decision making mechanism works with the following steps [20]: 1) Define the Decision Opportunity, 2) Identify Stakeholder Interests, 3) Build a Decision Framework, 4) Rate the Alternatives, 5) Weight Stakeholder Interests, 6) Score the Alternatives, 7) Discuss Results, Re-Score, Discuss Again, and Decide.

# 3 Evaluation Approach

The aim of this study was to provide estimators with a vehicle in selecting the best fit software effort estimation method to enable more accurate effort estimation of software projects, which is an important step in software project planning. Accordingly, an evaluation approach based on Multi-Criteria Decision Analysis was created to select the most suitable software effort estimation method. While applying the MCDA, the core elements were determined as follows:

- *Problem*: Estimating software project effort accurately
- *Requirements*: Developing software effort estimation model considering project requirements, data and environmental dynamics
- *Goal*: Selecting a SEE method that can best meet the requirements
- *Criteria*: Various aspects required to develop a software effort estimation model in relation to project requirements
- *Alternatives*: Software effort estimation methods that can meet the requirements in accordance with the determined criteria
- *MCDA Tool*: An excel based decision matrix prepared using expert opinions.

## 3.1 Alternatives and Criteria

**Alternatives.** There are many different classifications of estimation methods in the literature [21]. In this study, we tried to select the most common effort estimation methods in classification and review studies. Although many review studies only examine the methods of one classification, we selected our alternatives by choosing methods from different classifications. While choosing our alternatives, we paid attention to be the most applied methods according to literature review studies. The methods we have chosen as an alternative in our study are as follows:

- Neural Networks (NN) [6,8,11]
- Case-Base Reasoning (CBR) [6,8]
- Linear Regression (LR) [7,8,10]
- Analogy Based (AB) [7,11]
- Expert Judgement (EJ) [7,10,11]
- Support Vector Regression (SVR) [6,8]
- Decision Trees (DT) [6,8]
- Bayesian Networks (BN) [6,8]

**Criteria.** While preparing the questionnaire, the criteria that distinguish the SEE methods to evaluate were determined. These criteria play a role in determining how well the requirements match the methods. It is also aimed to determine the basic properties of the methods and to determine their compatibility with the project dynamics [13]. Criteria and related questions are shown in Table 1. The headings of criteria used in evaluation are explained below.

a) *Approach to construct method*: This criterion defines the method's approach to data dependency when configuring the SEE method. Methods estimate effort using historical data or estimation is done with different inputs independent of data.

b) *Data characteristics*: When creating the SEE method, the characteristics of data are decisive to choose the method to be successful. Addressing the limitations of the data will help in choosing the right method. The sub-criteria determined for data characteristics are as follows: type of input data, dataset size, and number of parameters.

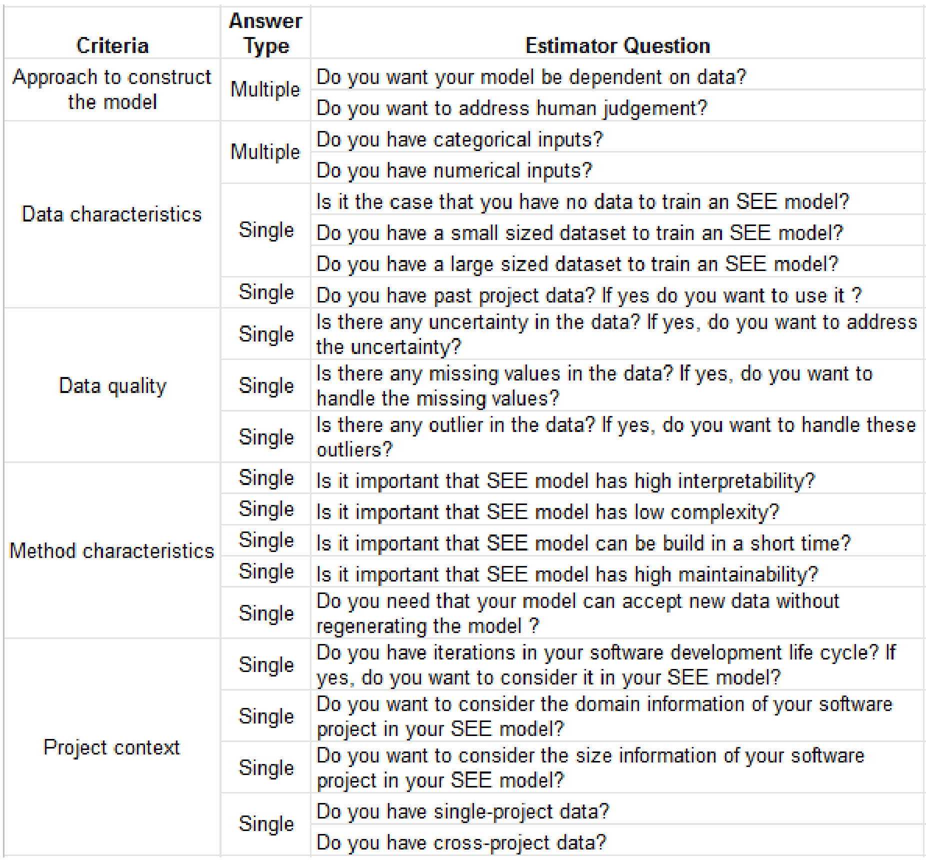
c) *Data quality*: This criterion indicates the quality features of the data that will be used to construct the SEE method. These are uncertainty, missing values, and outliers. Uncertain data means that the data may be inaccurate, imprecise, untrusted or unknown. Besides, missing data for certain variables leads to poor estimations in some sensitive methods. Also, the outlier data can affect choosing the suitable method. An outlier is an observation that lies an abnormal distance from other values in a dataset.

d) *Method characteristics*: This criterion defines the characteristics of the methods to use to construct the SEE method. The method should be interpretable, easy to use (not complex), speedy, maintainable, and adaptive. Interpretability indicates that the user can understand the cause of any result. Ease of use (not being complex) is the degree of which the method is not complicated in design. Speed is the degree of which the method is built in a short time and performs fast in general. Maintainability is the degree of which the method is easy to manage in time. Being adaptive means that the method can accept new data without re-running the SEE method.

e) *Project context*: This criterion indicates the factors related to the context information of the project subject to SEE. The factors are iteration, domain, size, and project type. Software development life cycle is an affecting factor to build the SEE method. Domain information is the expertise in the project area. Project size information is considered as the size criterion. Project data type information represents cross-project or single-project options. There are differences between these types in terms of project management and obtaining project information. Project data type has been added as a criterion for information on whether this affects the method selection.

In addition, the experts who answered the expert-opinion survey were asked to add the criteria that they thought would affect the choice of the method and to add further methods that should be considered if any. They suggested that personnel parameters and project parameters should be added to the evaluation criteria. Fuzzy logic, soft computing methods, and sequential model optimization were suggested as the additional methods that should be considered in evaluation. Also, the experts advised that we should study with criteria and methods from the industry users' perspective, and not only the researchers' perspective.

**Table 1.** Criteria and related estimator questionnaire



| Criteria | Answer Type | Estimator Question |
|---|---|---|
| Approach to construct the model | Multiple | Do you want your model be dependent on data? |
| | | Do you want to address human judgement? |
| Data characteristics | Multiple | Do you have categorical inputs? |
| | | Do you have numerical inputs? |
| | Single | Is it the case that you have no data to train an SEE model? |
| | | Do you have a small sized dataset to train an SEE model? |
| | | Do you have a large sized dataset to train an SEE model? |
| | Single | Do you have past project data? If yes do you want to use it ? |
| Data quality | Single | Is there any uncertainty in the data? If yes, do you want to address the uncertainty? |
| | Single | Is there any missing values in the data? If yes, do you want to handle the missing values? |
| | Single | Is there any outlier in the data? If yes, do you want to handle these outliers? |
| Method characteristics | Single | Is it important that SEE model has high interpretability? |
| | Single | Is it important that SEE model has low complexity? |
| | Single | Is it important that SEE model can be build in a short time? |
| | Single | Is it important that SEE model has high maintainability? |
| | Single | Do you need that your model can accept new data without regenerating the model ? |
| Project context | Single | Do you have iterations in your software development life cycle? If yes, do you want to consider it in your SEE model? |
| | Single | Do you want to consider the domain information of your software project in your SEE model? |
| | Single | Do you want to consider the size information of your software project in your SEE model? |
| | Single | Do you have single-project data? |
| | | Do you have cross-project data? |

### 3.2 Expert-Opinion Survey and Estimator Questionnaire

A well-defined expert-opinion survey that collects the necessary data to specify the characteristics of the effort estimation methods was designed and conducted. The survey consisted of questions that allowed us to determine the weight of criteria defined above for the estimation methods. A group of experts having published studies on software effort estimation was selected and asked to participate in the survey. The experts have been doing academic studies for a long time in the field of effort estimation as seen in Figure 1. The expert-opinion survey resulted in answers by eight experts for three different question types; List selection (QT1), Ranking on Likert scale (QT2), Yes/No selection (QT3). The first type is list selection, for which possible answers are A, B, both A and B. The Likert scale used has the following answer options: very low, low, average, high, very high. The last one is Yes/No choice. As an example, the answers to these three question types for the Expert Judgment estimation method are given in Table 2.

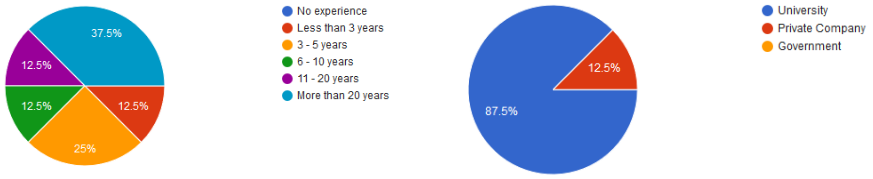


**Fig. 1.** Year of expertise in SEE and organization types of the experts

**Table 2.** Answers to three types of questions for Expert Judgment estimation method

| **Expert** | **QT1:** Please select the convenient option on “Approach to Construct the SEE method” with the below methods. | **QT2:** To what extent do you think the following methods are “interpretable” by its users in SEE? | **QT3:** Do you think that iteration in software development life cycle is an affecting factor in SEE with the following methods? |
|---|---|---|---|
| E1 | Based on human judgement | Low | Yes |
| E2 | Based on human judgement | High | Yes |
| E3 | Can address both | Very Low | Yes |
| E4 | Based on human judgement | Average | Yes |
| E5 | Based on human judgement | Very High | Yes |
| E6 | Based on human judgement | | |
| E7 | Based on human judgement | High | Yes |
| E8 | Based on human judgement | High | No |

The decision matrix in Table 3 was created using the answers to the expert-opinion survey from eight experts. Estimator questions and weights in the decision matrix were derived from the expert-opinion survey results. We explain below the steps for generating and weighting three sample estimator questions (EQ) with respect to the three types of survey questions.

**QT1.** “Do you want your method be dependent on data?” (EQ1) and “Do you want to address human judgement?” (EQ2) questions were created of QT1 from the expert-opinion survey result. While determining the weight of EQ1 ($W_{EQ1}$), the number of “Dependent on data” and “Can address both” answers given was divided by the number of all answers to EQ1. Similarly, weight of EQ2 ($W_{EQ2}$) was determined by dividing the number of “Based on human judgment” and “Can address both” responses by the number of all responses to EQ2.

- $W_{EQ1}$ = Count (Dependent on data) + Count (Can address both) / Count (All EQ1 answers)
  $W_{EQ1}$ = (0 + 1) / 8
  $W_{EQ1}$ = 0.13

- $W_{EQ2}$ = (Count (Based on human judgement) + Count (Can address both)) / Count (All EQ2 answers)
  $W_{EQ2}$ = (7 + 1) / 8
  $W_{EQ2}$ = 1

**QT2.** The question “Is it important that SEE method has high interpretability?” (EQ12) was created of QT2 from the expert-opinion survey result. When determining the weight of EQ12, values in range [1-5] were assigned for the answers in range [Low-Very High]. Total weight of EQ12 ($W_{Total\text{-}EQ12}$) was calculated by summing the product of each answer (in [1-5]) by the determined weight value which was taken as the number of that answer given. Weighted sum of EQ12 ($W_{EQ12}$) was determined by dividing the total weight of EQ12 ($W_{Total\text{-}EQ12}$) by the sum of all EQ12 responses multiplied by the maximum weight value of 5.

- $W_{Total\text{-}EQ12}$ = 1 x Count (Very Low) + 2 x Count (Low) + 3 x Count (Average) + 4 x Count (High) + 5 x Count (Very High)  
  $W_{Total\text{-}EQ12}$ = 1 x 1 + 2 x 1 + 3 x 1 + 4 x 3 + 5 x 1  
  $W_{Total\text{-}EQ12}$ = 23

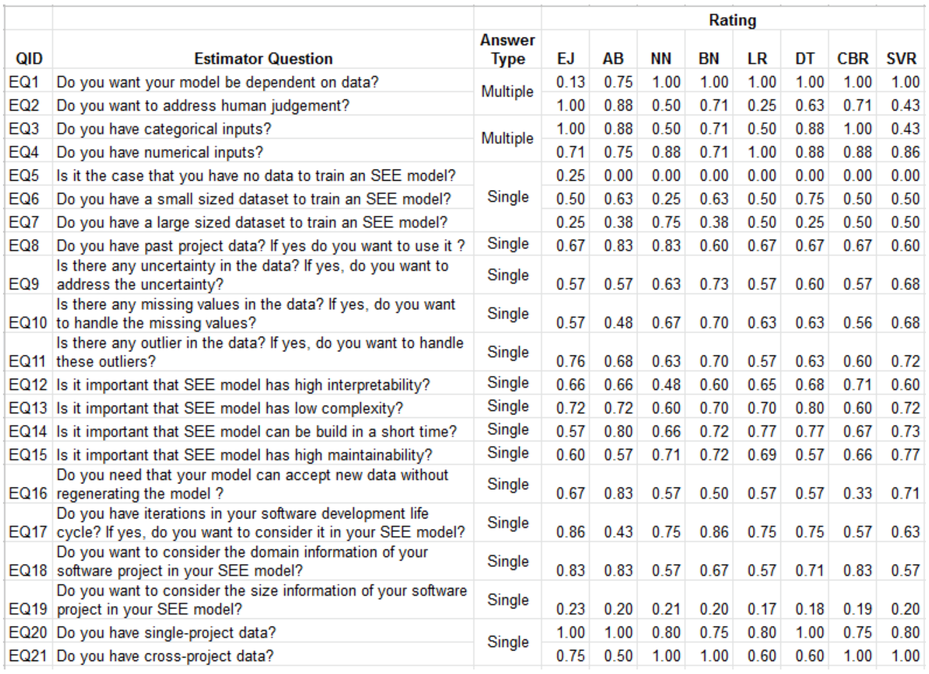
- $W_{EQ12}$ = $W_{Total\text{-}EQ12}$ / (Count (All EQ12 answers) x 5)  
  $W_{EQ12}$ = 23 / (7 x 5)  
  $W_{EQ12}$ = 0.66

**QT3.** The question “Do you prefer iteration in software development life cycle?” (EQ17) was created of QT3 from the expert-opinion survey result and its weight was determined by dividing the number of “Yes” answers by the number of all EQ13 answers.

- $W_{EQ17}$ = Count (Yes) / Count (All QT3 answers)  
  $W_{EQ17}$ = 6 / 7  
  $W_{EQ17}$ = 0.86

The weights of the estimator questions were interpolated to the range [0-1] to ensure that no criteria dominate other criteria during selection of an estimation method. In the calculation, the total number of answers given to the questions was used to eliminate the effect of the questions that were not answered by the experts. In this way, it is aimed to determine the estimation method selection not from the weight difference between the criteria, but from the weight difference between the key features of the methods.

**Table 3.** Decision Matrix



| | | | Rating | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| QID | Estimator Question | Answer Type | EJ | AB | NN | BN | LR | DT | CBR | SVR |
| EQ1 | Do you want your model be dependent on data? | Multiple | 0.13 | 0.75 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| EQ2 | Do you want to address human judgement? | | 1.00 | 0.88 | 0.50 | 0.71 | 0.25 | 0.63 | 0.71 | 0.43 |
| EQ3 | Do you have categorical inputs? | Multiple | 1.00 | 0.88 | 0.50 | 0.71 | 0.50 | 0.88 | 1.00 | 0.43 |
| EQ4 | Do you have numerical inputs? | | 0.71 | 0.75 | 0.88 | 0.71 | 1.00 | 0.88 | 0.88 | 0.86 |
| EQ5 | Is it the case that you have no data to train an SEE model? | Single | 0.25 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ6 | Do you have a small sized dataset to train an SEE model? | | 0.50 | 0.63 | 0.25 | 0.63 | 0.50 | 0.75 | 0.50 | 0.50 |
| EQ7 | Do you have a large sized dataset to train an SEE model? | | 0.25 | 0.38 | 0.75 | 0.38 | 0.50 | 0.25 | 0.50 | 0.50 |
| EQ8 | Do you have past project data? If yes do you want to use it ? | Single | 0.67 | 0.83 | 0.83 | 0.60 | 0.67 | 0.67 | 0.67 | 0.60 |
| EQ9 | Is there any uncertainty in the data? If yes, do you want to address the uncertainty? | Single | 0.57 | 0.57 | 0.63 | 0.73 | 0.57 | 0.60 | 0.57 | 0.68 |
| EQ10 | Is there any missing values in the data? If yes, do you want to handle the missing values? | Single | 0.57 | 0.48 | 0.67 | 0.70 | 0.63 | 0.63 | 0.56 | 0.68 |
| EQ11 | Is there any outlier in the data? If yes, do you want to handle these outliers? | Single | 0.76 | 0.68 | 0.63 | 0.70 | 0.57 | 0.63 | 0.60 | 0.72 |
| EQ12 | Is it important that SEE model has high interpretability? | Single | 0.66 | 0.66 | 0.48 | 0.60 | 0.65 | 0.68 | 0.71 | 0.60 |
| EQ13 | Is it important that SEE model has low complexity? | Single | 0.72 | 0.72 | 0.60 | 0.70 | 0.70 | 0.80 | 0.60 | 0.72 |
| EQ14 | Is it important that SEE model can be build in a short time? | Single | 0.57 | 0.80 | 0.66 | 0.72 | 0.77 | 0.77 | 0.67 | 0.73 |
| EQ15 | Is it important that SEE model has high maintainability? | Single | 0.60 | 0.57 | 0.71 | 0.72 | 0.69 | 0.57 | 0.66 | 0.77 |
| EQ16 | Do you need that your model can accept new data without regenerating the model ? | Single | 0.67 | 0.83 | 0.57 | 0.50 | 0.57 | 0.57 | 0.33 | 0.71 |
| EQ17 | Do you have iterations in your software development life cycle? If yes, do you want to consider it in your SEE model? | Single | 0.86 | 0.43 | 0.75 | 0.86 | 0.75 | 0.75 | 0.57 | 0.63 |
| EQ18 | Do you want to consider the domain information of your software project in your SEE model? | Single | 0.83 | 0.83 | 0.57 | 0.67 | 0.57 | 0.71 | 0.83 | 0.57 |
| EQ19 | Do you want to consider the size information of your software project in your SEE model? | Single | 0.23 | 0.20 | 0.21 | 0.20 | 0.17 | 0.18 | 0.19 | 0.20 |
| EQ20 | Do you have single-project data? | Single | 1.00 | 1.00 | 0.80 | 0.75 | 0.80 | 1.00 | 0.75 | 0.80 |
| EQ21 | Do you have cross-project data? | | 0.75 | 0.50 | 1.00 | 1.00 | 0.60 | 0.60 | 1.00 | 1.00 |

An estimator questionnaire was derived from expert-opinion survey answers. The questionnaire is intended for use by a project staff who holds the role of an estimator and wants to carry out effort estimation in his/her project accurately. The expert-opinion survey was filled once by the experts and a decision matrix was prepared from it. After having the decision matrix prepared, the estimator can use this matrix in order to select the most suitable estimation method for his/her need by answering a number of estimator questions (EQ).

In the decision matrix shown in Table 3, the first column (QID) refers to the identifier of the estimator question, the second column (Estimator Question) refers to the description of the estimator question, and the third column (Answer Type) refers to the way the question is answered. In that column 'Multiple' value is used for the criteria elicited by answering more than one question, and 'Single' value is used for the criteria elicited by answering only one question. In the other columns (Rating), the weights calculated from the expert-opinion survey as detailed above according to the estimator question types for the relevant estimation methods are given.

In the estimation process, an estimator answers the estimator questions by giving a value of 1 or 0, suitable for the question in each row. The answers are multiplied by the relevant method ratings, and the calculated scores for all questions are summed for each method to find the method scores. The method with a higher score is more suitable for estimation. Details of using the decision matrix in estimation process is explained in the next section.

# 4 Example Evaluation

## 4.1 Data Set

The proposed method was operated using the ISBSG [22] dataset as an example. According to the study [9], ISBSG dataset is widely used for software project estimations. The International Software Benchmarking Standards Group (ISBSG) maintains a data repository containing software project data from many organizations. The ISBSG aims to provide a wide range of project data from many sectors to organizations. These data can be used for awareness of trends, effort estimation, productivity benchmarking and comparing platforms and languages. The version of the ISBSG dataset that we used is Release 2016 R1.1.

## 4.2 Evaluation

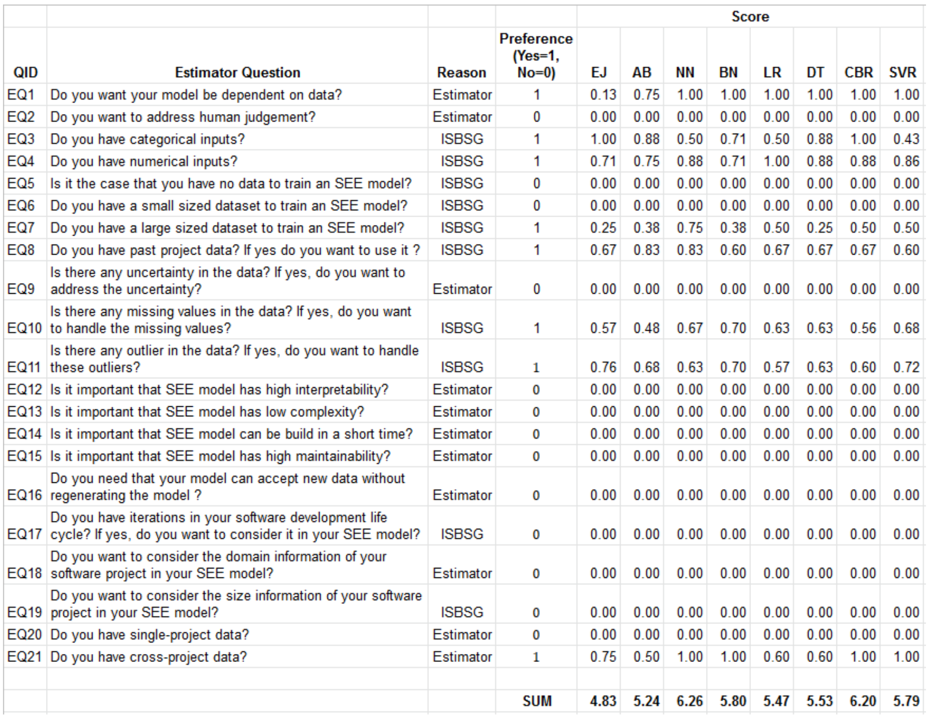
The decision matrix described in Table 3 was detailed with an example evaluation in Table 4. The estimator questions were answered using the ISBSG dataset and a number of assumptions regarding the example estimation. In order to answer the questions, a project of a company was selected from the ISBSG dataset and its information was examined. While some of the answers were answered directly by using the dataset, some of them were answered by the first author according to the hypothetical estimation needs, considering the project information of the company. This information is shown in the Reason column in Table 4. The questions were answered as 1 for yes and 0 for no, in accordance with the estimation context.

The reasons for answering the questions are as follows. The ISBSG dataset is used for the example estimation (EQ1) and the estimation model is preferred not to be dependent on human judgment (EQ2). That is why the answer to EQ1 is Yes, while that of EQ2 is No. Since there are categorical and numerical inputs in the ISBSG dataset, the answers given to EQ3 and EQ4 are Yes. The size of the dataset that can be used for training in the dataset is large, so the answer to EQ7 is Yes while the answers to EQ5 and EQ6 are No. The answer to EQ8 is Yes since other projects' information can be used. The uncertainty information will not be addressed in the estimation, so the answer given to EQ9 is No. There are missing data in the dataset and this information will be handled in the estimation process (EQ10). There is an abnormal distance between the values in the dataset, so the preference is Yes for EQ11. The estimator does not need the estimation model to have high interpretability, low complexity, high maintainability and short built time. Therefore, the preferences are No for EQ12, EQ13, EQ14, and EQ15. The estimator does not need that the model can accept new data without regenerating so the answer for EQ16 is No. The iteration information from the dataset will not be handled in the estimation process (No for EQ17). The domain information will not be used in estimation, so the answer for EQ18 is No. The size information can be found in the dataset (Yes for EQ19). Finally, the estimator considers the project is a cross-project (No for EQ20 and Yes for EQ21).

After entering estimator responses, method scores were calculated using the rating values in the decision matrix. Summing all the scores in the relevant method column, the total score for each method was obtained in the last (SUM) row of the table. The answers and the total scores for each estimation method in our example evaluation can be seen in Table 4. The answers were given with respect to the characteristics of ISBSG dataset and the estimator's assumptions.

According to the decision matrix prepared with our approach, the most suitable effort estimation model with a score of 6.26 is the Neural Network (NN) method and then the Case Base Reasoning (CBR) method with a score of 6.20.

**Table 4.** Example Evaluation using the Decision Matrix



| | | | | Score | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| QID | Estimator Question | Reason | Preference (Yes=1, No=0) | EJ | AB | NN | BN | LR | DT | CBR | SVR |
| EQ1 | Do you want your model be dependent on data? | Estimator | 1 | 0.13 | 0.75 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| EQ2 | Do you want to address human judgement? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ3 | Do you have categorical inputs? | ISBSG | 1 | 1.00 | 0.88 | 0.50 | 0.71 | 0.50 | 0.88 | 1.00 | 0.43 |
| EQ4 | Do you have numerical inputs? | ISBSG | 1 | 0.71 | 0.75 | 0.88 | 0.71 | 1.00 | 0.88 | 0.88 | 0.86 |
| EQ5 | Is it the case that you have no data to train an SEE model? | ISBSG | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ6 | Do you have a small sized dataset to train an SEE model? | ISBSG | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ7 | Do you have a large sized dataset to train an SEE model? | ISBSG | 1 | 0.25 | 0.38 | 0.75 | 0.38 | 0.50 | 0.25 | 0.50 | 0.50 |
| EQ8 | Do you have past project data? If yes do you want to use it ? | ISBSG | 1 | 0.67 | 0.83 | 0.83 | 0.60 | 0.67 | 0.67 | 0.67 | 0.60 |
| EQ9 | Is there any uncertainty in the data? If yes, do you want to address the uncertainty? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ10 | Is there any missing values in the data? If yes, do you want to handle the missing values? | ISBSG | 1 | 0.57 | 0.48 | 0.67 | 0.70 | 0.63 | 0.63 | 0.56 | 0.68 |
| EQ11 | Is there any outlier in the data? If yes, do you want to handle these outliers? | ISBSG | 1 | 0.76 | 0.68 | 0.63 | 0.70 | 0.57 | 0.63 | 0.60 | 0.72 |
| EQ12 | Is it important that SEE model has high interpretability? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ13 | Is it important that SEE model has low complexity? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ14 | Is it important that SEE model can be build in a short time? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ15 | Is it important that SEE model has high maintainability? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ16 | Do you need that your model can accept new data without regenerating the model ? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ17 | Do you have iterations in your software development life cycle? If yes, do you want to consider it in your SEE model? | ISBSG | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ18 | Do you want to consider the domain information of your software project in your SEE model? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ19 | Do you want to consider the size information of your software project in your SEE model? | ISBSG | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ20 | Do you have single-project data? | Estimator | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EQ21 | Do you have cross-project data? | Estimator | 1 | 0.75 | 0.50 | 1.00 | 1.00 | 0.60 | 0.60 | 1.00 | 1.00 |
| | | | | | | | | | | | |
| | | | **SUM** | **4.83** | **5.24** | **6.26** | **5.80** | **5.47** | **5.53** | **6.20** | **5.79** |

Wen et al. [6] showed that NN and CBR methods with the usage of the ISBSG data set are the most frequently used ones. They prepared a list for "distribution of the studies over the types of ML techniques". CBR and NN are at the top of the list. The research interest in CBR and NN methods have increased over the years compared to other methods. Also, these methods are more accurate than others when working with the ISBSG data set. According to the mean magnitude of relative error (MMRE) values examined in the study, NN performed better than all other methods.

Marco et al. [8] systematically gathered the information of many studies that examined estimation methods in terms of accuracy. According to the results, the two best MMRE values of the studies performed with the NN method for the ISBSG dataset were calculated as 9.5 and 49. The two best MMRE values for CBR method with the same dataset were obtained as 53 and 52.32. It is seen from these results that NN achieves better estimation performance with ISBSG dataset. As in our study, NN is more accurate when a choice is made between NN and CBR.

Venkataiah et al. [23] examined which data set and which methods were studied together in the literature. As a result of his analysis, he stated that one of the most worked methods with the ISBSG data set is NN.

The above studies [6,8] review and list the MMRE values of the estimation methods from multiple studies. Although in some of these studies it is reported that methods other than NN and CBR give more accurate results (e.g. [24,25]), there are also studies that contain results that support the selection of these methods as suggested by our study (e.g. [26,27]). Therefore, we can say that the results obtained by our proposed approach in the sample evaluation is partially supported with the results and suggestions of studies in the latter group.

Nevertheless, comparing the selection of estimation methods in the studies based on the resulting MMRE values only might remain incomplete since the estimation process includes many requirements and assumptions other than the ones related to the dataset, as also considered in our evaluation approach. Accordingly, we need to create further estimation cases, or repeat the past estimation cases by applying our questionnaire when possible, to make more meaningful comparisons and to discuss the reliability of our evaluation approach. This is left for future work at the moment.

## 5 Discussion

The proposed approach addresses the problem of selecting a suitable software effort estimation method through structuring the information and suggestions of the studies in literature into a decision matrix. It will be beneficial to expand the scope of the work by including the gains in the field of effort estimation in software industry. For this purpose, it will be beneficial to include the opinions of the experts working in this field in the industry to the expert opinions analyzed within the scope of our study. In this way, in addition to the observed effects in academy, the effects experienced in industry can be reflected to the process of selecting a suitable estimation method. In addition, eight effort estimation methods, which are widely referenced in the literature, are analyzed within the scope of this study. Similarly, the scope of the study can be expanded by analyzing the effort estimation methods and features that are commonly used in the industry.

The most important factor affecting the selection of the appropriate estimation method is the answers to the estimator questions. Therefore, in order to answer the questionnaire, it is necessary to have sufficient knowledge of the characteristics of the project and related data to be included in the estimation. Failure to reflect the project characteristic to the decision matrix through estimator questions will negatively affect the selection of the estimation method. This situation may lead to poor estimation by choosing an unsuitable method. Our approach does not control whether the project characteristics are accurately reflected in the decision matrix. The responsibility in this matter is on the person who will perform the estimation.

The majority of the experts who answered the expert-opinion survey are from the university. In future studies, reaching the experience of the people working in this field in the industry will increase the value of the expert-opinion survey results.

## 6 Conclusion

In this study, an approach has been proposed for selecting the most suitable software effort estimation method considering the project characteristics and the needs of the stakeholders. The approach aims to assist users in choosing the most suitable estimation method in the targeted estimation context.

We started this study by identifying the distinctive criteria for software effort estimation methods. To identify these criteria, we used findings of several literature reviews and followed the approach of a similar study in [13]. Then, using the criteria, we prepared an expert-opinion survey to take the opinions of the experts in SEE. The aim of the expert-opinion survey was to enable us to establish a relationship between methods and criteria, in the form of a decision matrix. We calculated the rating values for the questions derived from the criteria using the answers given by eight experts to the survey.

Then, a questionnaire was prepared to be answered by the user (estimator) who wants to perform the estimation. The user would be able to see the accuracy scores of the estimation methods on the decision matrix by answering the questionnaire.

To make our work understandable, we explained it through an example evaluation based on the ISBSG data set, and found that Neural Network and Case Based Reasoning are the most suitable methods in our estimation context. This method selection is partially supported with the results of the studies in the literature, and there is a need for further studies to validate the results of the evaluation approach.

We think that one of the most important factors that determine the success of our study is the number of experts who answer the expert-opinion survey. As future work, we plan to send our survey to the experts in industry. We think that the expert-opinion survey answered by more experts will increase the reliability of the decision matrix.

## References

1. Shepperd, Martin & Cartwright, Michelle. Predicting with sparse data. IEEE Trans. Softw.Eng., 27(11):987–998, 2001
2. Idri, Ali & Mbarki, Samir & Abran, Alain. (2004). Validating and understanding software cost estimation models based on neural networks. 433 - 434. 10.1109/ICTTA.2004.1307817.
3. Nayebi, Fatih & Abran, Alain & Desharnais, Jean-Marc. (2015). Automated selection of a software effort estimation model based on accuracy and uncertainty. Artificial Intelligence Research. 4. 10.5430/air.v4n2p45.

4. Sehra, Sumeet Kaur & Brar, Yadwinder & Kaur, Navdeep. (2012). Multi Criteria Decision Making Approach for Selecting Effort Estimation Model. International Journal of Computer applications. 39. 975-8887. 10.5120/4783-6989.
5. Bansal, Ashu & Kumar, Brijesh & Garg, Rakesh. (2017). Multi-criteria decision making approach for the selection of software effort estimation model. Management Science Letters. 7. 285-296. 10.5267/j.msl.2017.3.003.
6. Wen, Jianfeng & Li, Shixian & Lin, Zhiyong & Hu, Yong & Huang, Changqin. (2012). Systematic literature review of machine learning based software development effort estimation models. Information & Software Technology. 54. 41-59. 10.1016/j.infsof.2011.09.002.
7. Jorgensen, M. & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, 33, 33--53. doi: 10.1109/TSE.2007.256943.
8. Marco, R. & Suryana, N. & Ahmad, S.S.S. (2019). A systematic literature review on methods for software effort estimation. Journal of Theoretical and Applied Information Technology. 97. 434-464.
9. Idri, A., Amazal, F. A. & Abran, A. (2015). Analogy-based software development effort estimation: A systematic mapping and review. Inf. Softw. Technol., 58, 206-230.
10. Bilgaiyan, Saurabh & Sagnika, Santwana & Mishra, Samaresh & Das, M.N. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. JOURNAL OF ENGINEERING SCIENCE AND TECHNOLOGY REVIEW. 10. 51-64. 10.25103/jestr.104.08.
11. Shekhar, Shivangi & Kumar, Umesh. (2016). Review of Various Software Cost Estimation Techniques. International Journal of Computer Applications. 141. 31-34. 10.5120/ijca2016909867.
12. Chirra, Sai Mohan Reddy & Reza, Hassan. (2019). A Survey on Software Cost Estimation Techniques. Journal of Software Engineering and Applications. 12. 10.4236/jsea.2019.126014.
13. Ozakinci, Rana & Tarhan, Ayça. (2019). An Evaluation Approach for Selecting Suitable Defect Prediction Method at Early Phases. 10.1109/SEAA.2019.00040.
14. Figueira, José & Greco, Salvatore & Ehrgott, Matthias. (2005). Multiple Criteria Decision Analysis, State of the Art Surveys. Multiple Criteria Decision Analysis: State of the Art Surveys. 78. 10.1007/b100605.
15. Belton, Valerie & Stewart, Theodor. (2002). Multiple Criteria Decision Analysis: An Integrated Approach. 10.1007/978-1-4615-1495-4.
16. Saaty, R.W. (1987) The Analytic Hierarchy Process—What It Is and How It Is Used. Mathematical Modelling, 9, 161-176. http://dx.doi.org/10.1016/0270-0255(87)90473-8.
17. Hwang, C.L. & Yoon, K. (1981). Multiple Attribute Decision Making: Methods and Applications. Springer-Verlag, New York. 10.1007/978-3-642-48318-9.
18. Brans, J.P. & Mareschal, Bertrand. (2005). Chapter 5: PROMETHEE methods. Multiple Criteria Decision Analysis: State of the Art Surveys. 164-189.
19. Figueira, J. & Mousseau, V. & Roy, B. (2005) Electre Methods. In: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science, vol 78. Springer, New York, NY.
20. Multi-Criteria Decision Analysis, https://projects.ncsu.edu/nrli/decision-making/MCDA.php.
21. Vera, Tomas & Ochoa, Sergio & Perovich, Daniel. (2018). Survey of Software Development Effort Estimation Taxonomies. 10.13140/RG.2.2.14599.29601.
22. ISBSG, International Software Benchmarking standards Group, http://www.isbsg.org.

23. Venkataiah, V. & Mohanty, Rama & Nagaratna, M. (2017). Review on intelligent and soft computing techniques to predict software cost estimation. International Journal of Applied Engineering Research. 12. 12665-12681.
24. Moosavi, S. H. Samareh & Bardsiri, V. Khatibi (2017).Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. Eng. Appl. Artif. Intell., vol. 60. https://doi.org/10.1016/j.engappai.2017.01.006.
25. Pospieszny, Przemyslaw & Czarnacka-Chrobot, Beata & Kobyliński, Andrzej. (2017). An effective approach for software project effort and duration estimation with machine learning algorithms. Journal of Systems and Software. 137. 10.1016/j.jss.2017.11.066.
26. Azzeh, Mohammad & Neagu, Daniel & Cowling, Peter. (2011). Analogy-based software effort estimation using Fuzzy numbers. Journal of Systems and Software. 84. 270-284. 10.1016/j.jss.2010.09.028.
27. Azzeh, Mohammad & Neagu, Daniel & Cowling, Peter. (2010). Fuzzy grey relational analysis for software effort estimation. Empirical Software Engineering. 15. 60-90. 10.1007/s10664-009-9113-0.