IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2018, accepted January 4, 2019, date of publication January 11, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891878

Categorical Variable Segmentation Model for Software Development Effort Estimation

PETR SILHAVY, RADEK SILHAVY, AND ZDENKA PROKOPOVA
Department of Computer and Communication Systems, Tomas Bata University in Zlín, 760 01 Zlín, Czech Republic

Corresponding author: Petr Silhavy (psilhavy@utb.cz)

This work was supported by the Faculty of Applied Informatics, Tomas Bata University in Zlín, under Project RO30186021025/2102.

ABSTRACT This paper proposes a new software development effort estimation model. The new model's design is based on the function point analysis, categorical variable segmentation (CVS), and stepwise regression. The stepwise regression method is used for the creation of the unique estimation model of each segment. The estimation accuracy of the proposed model is compared to clustering-based models and the international function point user group model. It is shown that the proposed model increases estimation accuracy when compared to baseline methods: non-clustered functional point analysis and clustering-based models. The new CVS model achieves a significantly higher accuracy than the baseline methods.

INDEX TERMS Estimation, function point analysis, software engineering, software measurement.

I. INTRODUCTION

The Software Engineering industry and research employ mathematical models used to design a Parametric Estimation Model (PEM). These PEMs are proposed in order to resolve Budgeting, Software Complexity (Size), or Development-time Planning [1], [2]. In Software Engineering Development Effort Estimations (SEDEE), a Use Case Points (UCP) [3], or Function Point Analysis (FPA) [4], may be used as a PEM. Improving PEM accuracy is the main aim in software development effort estimation research. During the past several years, research in software development effort has focused on improving the accuracy of the estimations. Models' improvements are mostly based on Computational Statistics [5], or Machine Learning [3], [6], methods. Improvements are focused on improving or new designs of algorithms – the counting process.

Some papers have investigated the influence of using historical data. Historical data is understood to mean past, previously-finished, software development projects. Historical data is used for Model Tuning or Design. Historical data can be used as a generic dataset, which is available from providers like the International Software Benchmarking Standards Group, (ISBSG), [7]. More accurate estimations are available from internal historical data - as was declared in [8]. The reason why historical data is better for estimation purposes lies in the similarity and consistency of the dataset. There are approaches which help one to find similar projects in the historical data; even if a generic dataset, (across a company), is used.

In this study, the International Function Point Users Group, (IFPUG), [4], [9], method is used as the basis for the research. The IFPUG method is used for obtaining model variables - (Categorical Variables, Dependent Variables, and Predictors). The original IFPUG method leads to complexity measurements, (size). If a software development effort - (time in person-hours) is needed, then a Productivity Factor (PF), has to be used for the transformation of the estimated size into the number of person-hours [10].

The authors demonstrate that segmentation using categorical variables improves the estimation accuracy of the software development effort more than when using clustering based on known approaches - (Spectral Clustering, Regression Clustering). This finding allows one to design a new approach where model training is supposed to be less time-consuming than training models using conventional clustering methods.

The rest of this study is structured as follows. Section 2 defines the Problem Statement. Section 3 describes the methods used. Section 4 is dedicated to the Experiment Design. The results are presented in Section 5 - and discussed in Section 6. Finally, the Conclusion is set out in Section 7.

A. RELATED WORK

In some publications, [3], [11], and [12], authors have declared that partitioning of the dataset, i.e. selection of a set of similar past projects, can significantly increase the accuracy of estimation models. Bardsiri *et al.* [13], introduce a method to increase the software development effort estimation accuracy, based on the combination of Fuzzy Clustering, Analogy and Artificial Neural Networks. In the method proposed herein, the effect of irrelevant and inconsistent projects on such estimates is decreased.

Idri *et al.* [11], provide a systematic mapping study of Analogy-based Software Development Effort Estimation (ASEE). In this paper, the authors investigate 65 studies from 1990 to 2012. Most of these studies are oriented on Subset Method Selection. The ASEE method looks for similarities in historical projects. Clustering helps to find analogies between projects and, is a broadly investigated method for reducing the number of historical data-points and selecting the most similar subsets. Nassif *et al.* [14], deal with setting the number of nearest projects. These authors recommend a method called Bisecting k-medoids.

Azzeh and Nassif [15] described a hybrid model that consists of classification and prediction stages using a Support Vector Machine and Radial Basis Neural Networks. They compare the hybrid model with k-medoids. Gallego *et al.* [16] develop a methods to estimation equations elicitation through the division of historical project datasets. In [17], the positive effect of the modified Expectation-Maximization algorithm [18] is presented. Hihn *et al.* [19], described that the nearest neighbour method has significantly more outliers than spectral clustering does.

Bardisiri *et al.* [13], declare that clustering as method of dataset segmentation has a significant effect on the accuracy of development effort estimation because it allows one to omit irrelevant projects from historical data-points.

Prokopová *et al.* [20], compare k-means, hierarchical and density-based clustering techniques with three different distance metrics. The results shows how important is to select the clustering type and distance metric properly. The authors show that hierarchical clustering has produced inappropriate distribution of clusters – and therefore, cannot be used. The k-means clustering technique appears to be the most appropriate method for segmentation.

Bardisiri *et al.* [21] introduce the combination of ASEE and the Particle Swarm Optimization (PSO) [22] algorithm. They introduce a weighting system in which the project attributes of different clusters are given different weights. This approach supports the comparison of a new project only with projects located in related clusters, based on the similarity measures.

Lokan and Mendes [23] investigations showed that moving windows are helpful as a subset selection technique, which demonstrates that the most recent projects are more important and makes estimation more accurate than using all available data points. Amasaki and Lokan [24], later compare moving windows for ASEE and regression models. Silhavy *et al.* [3] presented a study on the Moving Windows Segmentation Approach which is then compared to Clustering Approaches. Spectral Clustering was evaluated as the best option, when compared to the k-means or the moving windows approaches.

II. PROBLEM STATEMENT

When a dataset used for estimation training is large and inconsistent, then the estimation error is often very high. The clustering or segmentation of data helps to reduce estimation error. The authors understand the difference between segmentation and clustering as follows; segmentation is the process of grouping observations based on sharing the same value of the categorical variable, and clustering is the process of finding similarities in observations based on distance measurements.

The authors expect that the segmentation-based model may out-perform the clustering-based approach in its estimation accuracy. Furthermore, a segmentation-based model is supposed to be easy-to-use in the software industry than models using conventional clustering-based models.

Therefore, the authors present a new approach, the Categorical Variable Segmentation, (CVS), model. CVS is based on a FPA variant - called the IFPUG method, which is used to obtain model variables. The IFPUG method provides complexity measurements, (size), only. If a software development effort - (time in person-hours), is needed; a productivity factor (PF) has to be used to obtain the number of person-hours [10]. The proposed CVS model can be used for the estimation of the software development effort in person-hours.

The research goal of this study is to present the practical impact of the new proposed CVS model and to demonstrate its ability to decrease an estimation error when new software project development effort is estimated.

A. RESEARCH QUESTIONS

RQ1: Will a newCVS model out-perform IFPUG?
RQ2: Will a newCVS model out-perform Regression or Spectral clustering?
RQ3: Which of the tested categorical variables is the best option for segmentation?

B. EVALUATION CRITERIA

Estimation models can be evaluated using Mean Absolute Percentage Error, (MAPE; calculated by using (1), Mean Estimation Error, (MEE); or, calculated by using (2; or, using (3), to calculate the PRED (25).

MAPE was selected because - de Myttenaere *et al.* [25] prove that MAPE has practical and theoretical relevance for the evaluation of regression models and its intuitive interpretation regarding the relative error. Whereas MEE has a practical impact when person-hours are represented. PRED (l) describes the overall estimation quality within a selected level of percentage errors.

The formulas are given as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100 \tag{1}$$

$$MEE = median(y_i - \hat{y}_i) \tag{2}$$

$$PRED\,(l) = \frac{count\,\frac{|y_i - \hat{y}_i|}{y_i}}{n}, \quad if\ \frac{|y_i - \hat{y}_i|}{y_i} \le l \tag{3}$$

where n is the number of observations; y_i is the known observed value of the effort; $\hat{y}_i$ is the estimated value of the effort; and l is the threshold of percentage error. If $l = 0.25$, then the observation estimation error is less than 25%.

III. METHODS USED

Several methods were involved in this study. An FPA method, with an IFPUG variant, was used for variable determination. Stepwise Regression was used for model-training as a regression analysis method. Finally, Spectral Clustering and Regression Clustering are also presented in this text.

A. FUNCTION POINT ANALYSIS – IFPUG

Function Point Analysis, (FPA), was originally developed by Albrecht [26] in the late 1970s. The first point is that the FPA approach introduces three transaction function types; External Inputs (EI), External Outputs (EO) and External Inquiry (EC).

EI describes data-processing incoming to the application from outside a boundary. EO is used for accessing data or control processes outside an application boundary after the processing is done. EC - sends data or control processes outside an application boundary, but no further processing is performed.

The second point is that the FPA introduces two data functions, these are Internal Logical Files (ILF) and External Interface Files (EIF). ILF represents data-processing in the form of a relation; it should be a data table. EIF represents logically connected data and control information, which are maintained by the external system.

FPA distinguish three project types, [27]; a new project, existing software enhancement, and applications.

To count Function Points (FP) using FPA (IFPUG) means that all functionalities have to be identified and classified by using a complexity level. ILF and EIF complexity are based on two factors – Data Elements Tables (DET) and Record Elements Tables (RET). DET is used for user-recognisable fields (user interfaces), and RET is the data element, (subgroups).

$$\begin{aligned} UFP = {} & \sum EI \times weight + \sum EO \times weight \\ & + \sum EC \times weight + \sum EIF \times weight \\ & + \sum ILF \times weight \end{aligned} \tag{4}$$

EI, EO, EC, EIF and ILF are used for calculating the unadjusted FP - (UFP), but many system characteristics are not covered in this phase. Later, the FP are adjusted by using General System Characteristics (GSC), which are weighted in the interval of ⟨0-5⟩ by their Degree of Influence, (DI); and finally, the Value Adjusted Factor, (VAF), is determined. VAF is calculated as follows (5), and this factor can change a UFP by as much as 35% $(+/-)$.

$$VAF = (TDI \times 0.01) + 0.65 \tag{5}$$

Adjusted FPs (AFP) are obtained by multiplication of the UFP and VAF - as can be seen in (6). Finally, the work effort ($Effort_{Estimated}$) in person-hours have be calculated.

The Productivity Factor (PF) is used as a constant that describes the relationship between one FP and the number of hours needed for its development. The PF value is derived in two different ways. First of all, the PF can be obtained as the mean of all PF from past projects. Secondly, a PF can be based on categorical values – it is therefore specific for certain types of projects [28], [29]. Productivity was studied in [6], [10], and [6].

$$AFP = UFP \times VAF \tag{6}$$

$$Effort_{Estimated} = AFP \times PF \tag{7}$$

Counting process can be summarised as follows:

1. Determine the type of count – EI, EO, EC, EIF, ILC
2. Identify the scope and boundary of the count
3. Determine the unadjusted FP count
4. Determine the VAF
5. Calculate the Adjusted FP Count
6. Effort calculation using AFP and corresponding PF value

B. STEPWISE REGRESSION

Stepwise Regression (SR) was adopted from [3], [5], and [30]. The Stepwise process of multiple linear regression is based on forward - and backward selection that involves an automatic process for the selection of independent variables; and can be briefly described as follows [3]:

1. Set a starting model that contains predefined terms (backward); or set a null-model (forward)
2. Set limits for the final model – determine the requested model complexity and which terms have to be included – linear, quadratic, interaction etc.
3. Set an evaluation threshold – the sum of residual errors is used to determine whether to remove or add a predictor
4. Adding or removing terms; re-testing the model
5. Stepwise regression halts when no further improvement in estimation occurs

Forward selection starts as a null-model and then iterates to add each variable which meets a given condition. When a non-significant variable is found, it is removed from the model. Backward selection works in a similar way, but removes variables when they are found to be non-significant. Therefore, stepwise regression requires two significance levels: the first - for adding variables; and the second - for removing variables.

SR is a method of building many models from a combination of predictors. Therefore, Multiple Linear Regression (MLR) assumptions have to be fulfilled. The MLR, is defined as:

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + \varepsilon_i \tag{8}$$

where, $i = 1, \ldots n$, y_i is the dependent variable; $X_{i1} \ldots X_{ip}$ are independent variables, (predictors); β_0 is an intercept; and, $\beta_1 \ldots \beta_n$ are regression coefficients. The value of ε_i

represents the residuals. The model is designed as a matrix - where each row represents a data-point.

When MLR is a polynomial regression, then the relationship between the dependent variables and the independent variable is modeled as a Math Degree Polynomial:

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}^2 + \ldots + \beta_p X_{ip}^m + \varepsilon_i \tag{9}$$

C. REGRESSION CLUSTERING

The regression clustering is based on SR. It uses SR for the elimination of observations that are unsuitable for forming a cluster. Clusters are determined according to Cook Distance [31]. Using the Cook Distance (10), a cluster cutoff is possible and influential data entries are identified for regression analysis. A similar approach was introduced by [32]:

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y_{j(i)}} \right)^2}{(p+1)\,\hat{\sigma}^2} \tag{10}$$

where D is the Cook Distance and is defined as a sum of all changes in the regression model when specific observation is removed from the regression model. $\hat{Y_{j(i)}}$ - stands for the fitted response. p is the number of predictors and $\hat{\sigma}^2$ is an estimated error variance for all observations.

D. SPECTRAL CLUSTERIN

The Spectral Clustering [33] algorithm is adopted from [3] and [33]–[35], based on graphical representation, where each data point is a node and the edges between data points represent similarity - see Graph G described by (11). This represents a Degree Diagonal Matrix in which a cell represents a sum of weights corresponding to each node from the graph - or respectively, a cell of matrix W:

$$G = (V, E) \tag{11}$$

where set V contains vertexes v_i and set E the edges - e_i, which represent data points. Two vertexes are connected if the similarity s_{ij} between the corresponding data points x_i and x_j are larger or equal to the threshold; and the edge is weighted by - s_{ij}. This means that part of the graph where edges with very low weights are found.

The k-nearest neighbour graph, ε-neighborhood graph and the fully-connected graph are typically used in Spectral Clustering [36]. The k-nearest neighbour graph connects v_i and v_j vertexes, where v_j is one of the k-nearest vertexes of v_i. The ε-neighborhood graph connects all data points where pairwise distances are smaller than ε. Later, the Adjacency Matrix W (12) is created:

$$W = (w_{ij}) \tag{12}$$

where, $i, j = 1..n$ and each cell in the matrix correspond to the edge weight between two data points. If the weight is 0, then there is no connection between the edges. Finally, a Laplacian Matrix (13) is calculated:

$$L = D - W \tag{13}$$

where D is the diagonal matrix of the degree of vertex v_i. The L matrix is used for spectrum calculation - which is a key point in spectral clustering algorithms. The L matrix is used in un-normalized algorithms; when a normalized Laplacian algorithm is used, there are two possibilities – a symmetric matrix (14a) and a random-walk (14b):

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{14a}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W \tag{14b}$$

The spectrum is a sorted list of the Eigen-vectors of a L, L_{sym} or L_{rw} matrix. In fact, the Eigen-vectors represent a data-point of a dataset and an Eigen value of a L, L_{sym} or L_{rw} matrix. Spectral clustering uses these Eigen vectors as a feature. The clustering of features can be performed by any known algorithm. In this paper, the k-means algorithm is used.

IV. EXPERIMENT DESIGN

A. DATA PRE-PROCESSING

In this study, an International Software Benchmarking Standards Group (ISBSG) dataset was adopted [7]. ISBSG dataset were pre-processed to filter only observations which met the following criteria:

1. IFPUG was used as the effort estimation methodology for the observation.
2. ISBSG data quality attribute were labeled as A or B.
3. Values are assigned to all involved variables (TABLE 1) - variables are not empty.
4. PF values for observations are within an interval $\langle Q_1 - 1.5 \times IQR; Q_3 + 1.5 \times IQR \rangle$.

All observations, where PF values were not in the interval of $\langle Q_1 - 1.5 \times IQR; Q_3 + 1.5 \times IQR \rangle$ were understood as outliners and therefore removed. A Q represents quartile and IQR stands for interquartile range (15).

$$IQR = Q_3 - Q_1 \tag{15}$$

TABLE 1. List of variables involved in experiments.

Variable	Abbreviation	Description
External Inputs	EI	Independent Variable
External Outputs	EO	Independent Variable
External Inquiry	EQ	Independent Variable
Internal Logical Files	ILF	Independent Variable
External Interface File	EIF	Independent Variable
Normalized Work Effort (person-hours)	Effort	Dependent Variable
Relative Size	Size	Categorical Variable
Industry Sector	Sector	Categorical Variable
Business Area Type	Business Area	Categorical Variable

Using a PF value in IQR filtering approach allows to remove all observations which PF is outside of selected interval (range). Reducing PF range allows obtaining the dataset, which simulate an in-house dataset.

Dataset pre-processing criteria allow to obtain a more consistent dataset which consists of 612 observations.

B. DATASET DESCRIPTION

The dataset, which is used in this study consist of 11 variables, which describe all observations (TABLE 1). Categorical variables (Size, Sector and Business Area) were used segmentation parameter in CVS model.

The dataset obtained was split into training and testing parts using the Hold-out Approach and a 2:1 ratio. TABLE 2, presents descriptive statistics of training and testing datasets (for the Effort Variable). As can be seen, the median value for the testing part of the data-set is higher than for the training part. The range is lower for the testing part.

TABLE 2. **Descriptive statistic of training and testing datasets.**

Dataset	n	Median Effort	SD Effort	Min Effort	Max Effort	Range Effort
Training	428	1,868	5,692	51	70,035	69,984
Testing	184	4,912	4,912	140	37,760	32,620

TABLE 3. **Intervals for categorical variable relative size.**

Categorical Value	Size Interval (in FP)
XXS	≥ 0 < 10
XS	≥ 10 < 30
S	≥ 30 < 100
M1	≥ 100 < 300
M2	≥ 300 < 1,000
L	≥ 1,000 < 3,000
XL	≥ 3,000 < 9,000
XXL	≥ 9,000 < 18,000

The categorical variables are described with more details since these variables are used for data-set segmentation and will be evaluated in the proposed algorithm design.

The Relative Size variables intervals are summarised in TABLE 3. Observations from XXS to XXL are also involved in the experiments. Originally, the ISBSG dataset projects contains more than 18,000 functional points, which were then eliminated during the data-cleaning procedure. The XXL and XXS sized projects are not available in the training data-set. This fact influences the design approach to the CVS model, where training for unavailable sizes is performed using a spare model based on the whole training data-set.

Figure 1 depicts a histogram of the Relative Size Distribution in the data-set.

The Industry Sector is the second categorical variable used for datasets segmenting. Sector values are depicted in Figure 2. As can be seen, some values – Defence, Medical & Health Care are representing by 2, respectively 5 observations. Therefore, the experiment procedure - when SR models are trained on the Training data-set, have to be dealt with. In this study, if the model for the category is not trained, then the model for the whole training data-set is used instead.

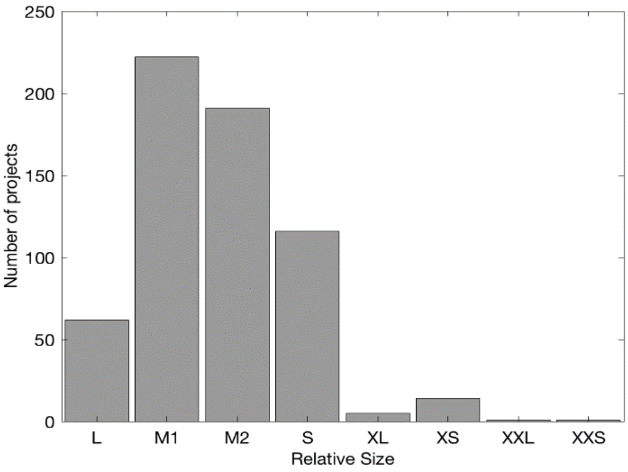

FIGURE 1. **Histogram of relative size variables.**

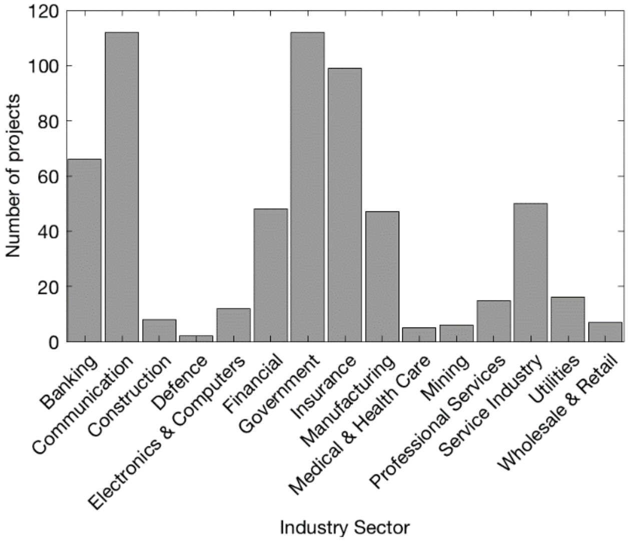

FIGURE 2. **Histogram of industry sector variable.**

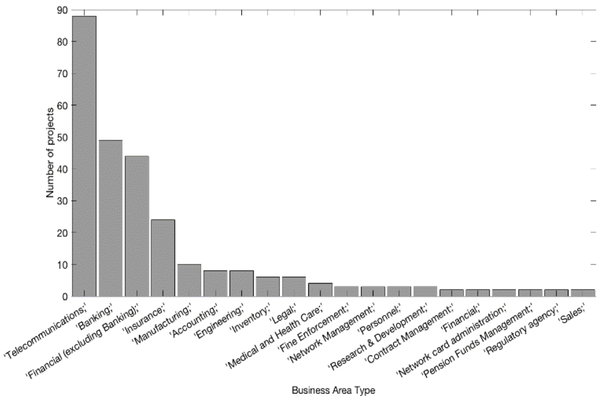

FIGURE 3. **Histogram of business area type variables.**

Finally, the third categorical variable – the Business Area Type is described. The Business Area Type allows detailed segmentation, (more categories as compared to Sector). Figure 3 presents the most frequent categories of the Business Area Type.

C. BASELINE MODELS

The CVS model is compared to three baseline models:

- The IFPUG Model
- The Regression Clustering-based Model, and …
- The Spectral Clustering-based Model

1) THE IFPUG MODEL

The IFPUG formulas - (4)-(7), are used to obtain new estimations. In this study, the PF Mean and PF Median values were tested. The PF values are used - based on the training part of the data-set.

To obtain a new estimation, do as follows:

1. Create training and testing sets by using the Hold-out (2:1) Method
2. Estimate the work effort in hours using the IFPUG Counting Procedure
3. Multiply the AFP by the PF value
4. Estimate the Error (EE) for projects in the testing set and compute them
5. The MAPE, MEE, and PRED(25) criteria are computed

2) THE REGRESSION CLUSTERING-BASED MODEL

The procedure for the Regression Clustering-based application was as follows:

1. Creating training and testing sets by using the Hold-out (2:1) method
2. Setting a feature list - EI, EO, EC, ILF, EIF. The dependent variable was set with regard to effort
3. Applying an SR to all projects in the training set
4. Removing all observations where $D_i > 3 \times median\ D$
5. Computing then SR model again
6. Repeating all processes as many times until no outliners are identified
7. If no more outliners are identified, then the rest of the observations form a cluster
8. Repeating the process with observations for which no cluster have been assigned yet

To obtain a new estimation, the procedure is as follows:

1. Classification of observations in the testing data-set into clusters, using Discriminant Analysis
2. Estimation of work effort in hours is performed by using a Cluster-specific model
3. Estimation Error (EE) for projects in the testing set is then computed
4. MAPE, MEE, and PRED (25) criteria are computed

3) THE SPECTRAL CLUSTERING-BASED MODEL

Spectral clustering is used with k-means. This means that the number of clusters have to be predefined before an algorithm can start. As can be seen in Step 3, a maximum number of clusters is derivated. This method is used instead of hyper-parameter tuning or other known methods for selecting the proper number of clusters. This was done because the correct number of clusters is evaluated according to MAPE, MEE and PRED (25), and no standard procedure for obtaining the number of clusters is able to handle these evaluation criteria.

The application procedure of Spectral Clustering was as follows:

1. Create training and testing sets by using the Hold-out (2:1) method
2. Set a feature list - EI, EO, EC, ILF, EIF. The dependent variable was set to Effort
3. Apply Spectral Clustering to the Training Data-set using a precondition of 15 observations in a cluster - which is used for the predefinition of the maximum number of clusters.
4. Computing SR models for all defined clusters

Obtain a new estimation - as follows:

1. Classify observations in the testing data-set into clusters, using Discriminant Analysis.
2. Estimate the work effort in hours by using a Cluster-specific model
3. Estimation Error (EE) for projects in the testing set is then computed
4. The MAPE, MEE, PRED (25) criteria are then computed.

D. THE CATEGORICAL VARIABLE SEGMENTATION MODEL

The proposed algorithm employs Segmentation by Categorical Variables, Stepwise Regression and the IFPUG Method. The algorithm design is based on the assumption that observation described by using an Identical Categorical Variable are more similar - and they therefore form segments that can be fitted into the SR Estimation Model. It is expected that each observation is described by a list of features obtained by the IFPUG Method - (EI, EO, EC, ILF and EIF). Observations in the training data-set are segmented using a Categorical Variable Value depending on which is used (Size, Sector or Business Area). Then, observations where the Categorical Variable has an identical value form a segment. The segment is used to perform an SR in order to obtain an estimation model. The model training procedure can be performed as follows:

1. Creating training and testing sets by using the Hold-out (2:1) method
2. Setting a feature list - EI, EO, EC, ILF, EIF. The dependent variable was set to Effort
3. Performing segmentation based on Selected Categorical Variables
4. Performing SR for each set of observations that equal the selected value of the Categorical Variable
5. Performing SR on all observations from the Training data-set - (resulting in the creation of the spare model)

The proposed model is then evaluated in relationship to these three categorical variables:

- Relative Size
- Industry Sector
- Business Area

When models are trained, a new estimation can be performed. Models for each of these segments are used. Before the estimation can be performed, the user has to set the value of the Categorical Variable. If the value that was planned to be used was missing in the Training data-set, then a spare model is used for the estimation. Without a spare model, the estimation of such a project cannot be performed.

Obtain a new estimation as follows:

1. Classification of observations in the Testing data-set into segments, using Categorical Variable values
2. Estimation of software development effort in person-hours is performed by using a segment-specific model – based on a Categorical Variable or using the spare model if a specific model is not pre-trained
3. Estimation Error (EE) for projects in the testing set is then computed
4. MAPE, MEE, and PRED (25) criteria are computed (for process evaluation estimation purposes)

V. RESULTS

A. BASELINE MODELS

1) THE IFPUG MODEL

The estimated effort in person-hours is compared to known Effort values in the testing data-set. TABLE 4 presents the evaluation of estimation ability of the IFPUG.

TABLE 4. Results of IFPUG model.

PF Method	MAPE	MEE	PRED(25)
Mean	130.79	-2286.90	0.03
Median	44.95	-435.55	0.49

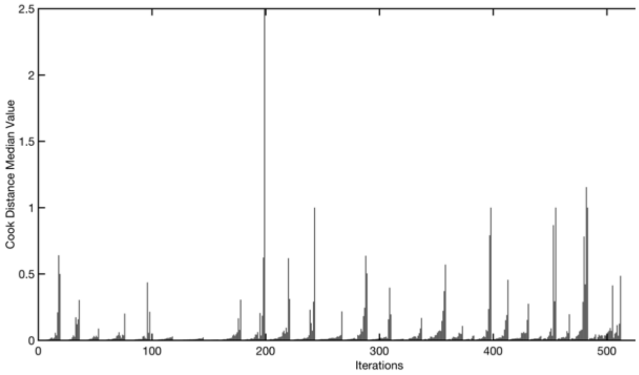

FIGURE 4. Cook distance medians over iterations.

2) THE REGRESSION CLUSTERING-BASED MODEL

Regression Clustering allows the formation of an apriori unknown number of clusters. All observations are clustered into as many classes as needed. This approach allows one to create nearly optimal MLR models for each cluster - based on the Training data-set. All conditions are fulfilled, including Residual Normal Distribution. 33 clusters were identified in 511 iterations of the algorithm. In Figure 4, the Cook Distance progress over iteration can be seen.

TABLE 5, shows the results obtained when Method 1 is used. The Median Estimation Error value is low, but PRED (25) shows that only 36% of new estimations have an error below the 25% threshold.

3) THE SPECTRAL CLUSTERING-BASED MODEL

Spectral Clustering is a modern clustering method, which is under consideration in many applications including Software Effort Estimation. This method uses k-means for clustering itself, which means it has similar disadvantages [35]. The number of clusters have to set in advance. In this study, a solution in the interval from 2 to 29 clusters was evaluated, with a condition of 15 observation in a cluster. TABLE 6, shows that clustering - up to a maximum of 19 clusters can be obtained. This means that the Testing data-set was classified into 19 clusters - but not all of the 19 clusters need to be used.

TABLE 5. Results of the regression clustering-based model.

Number of Clusters	MAPE	MEE	PRED(25)
33	92.13	18.72	0.36

TABLE 6. Results of the spectral clustering-based model.

Number of Clusters	MAPE	MEE	PRED(25)
2	43.50	-134.81	0.45
3	47.11	-67.93	0.48
4	43.87	-74.17	0.46
5	41.31	42.05	0.47
6	43.76	-105.65	0.44
7	59.34	-138.61	0.48
8	67.88	-206.11	0.38
9	73.46	-201.78	0.39
10	101.03	87.75	0.46
11	95.36	182.03	0.42
12	94.40	112.04	0.40
13	118.53	155.11	0.46
14	160.73	-52.46	0.42
15	119.74	-38.55	0.45
16	127.61	88.74	0.40
17	222.62	-154.56	0.38
18	325.00	-242.87	0.35
19	217.66	-65.53	0.41

TABLE 7. Results of CVS model.

Categorical	MAPE	MEE	PRED(25)
Relative Size	34.35	-15.16	0.53
Industry Sector	51.72	-44,30	0.43
Business Area Type	48.37	-173.20	0.46

B. THE CATEGORICAL VARIABLE SEGMENTATION MODEL

The CVS model is tested with the following Categorical Variables - Relative Size, Industry Sector and Business Area as a segmentation attribute, (see TABLE 3). TABLE 7 presents a comparison of Categorical Variable performance - as can be seen, Relative Size is the best option for model evaluation.

VI. DISCUSSION

This study compares the new CVS model to the IFPUG model as well as to selected clustering-based models. Those clustering-based models are - Regression Clustering (Cook Distance Elimination) and Spectral Clustering (k-means).

In the Problem Statement, three research questions were addressed: Will a new CVS model outperform IFPUG? (RQ1); Will a new CVS model outperform Regression or

Spectral Clustering? (RQ2; which of the tested categorical variables is the best option for segmentation? (RQ3).

TABLE 8 depicts a comparison between the CVS and Baseline methods. The Parameter Column shows the method configuration. Configuration includes the number of clusters for the Clustering-based method, Categorical Variable names or PF settings.

TABLE 8. Comparison between CVS and baseline models.

Method	Parameter	MAPE	MEE	PRED(25)	Overestimate	Underest
IFPUG	PF Median	44.96	-435.55	0.49	42	142
Regression Clustering	33	92.13	18.72	0.36	93	91
Spectral Clustering	5	41.31	42.05	0.47	95	89
CVS	Relative Size	34.35	-15.16	0.53	88	96

The IFPUG method was tested with PF based on mean or median value. The median-based PF allows one to achieve higher estimation accuracy - (see TABLE 4). When the Regression Clustering-based model is applied, then 33 clusters was the most accurate option - (see TABLE 5). Spectral Clustering worked best when 5 clusters were used - (see TABLE 6).

Finally, the CVS model is used with Relative Size as the segmentation parameter, this allows the most accurate estimation, (see TABLE 7).

When discussing the RQ1, it can be said that the CVS model outperforms the IFPUG method. As can be seen from TABLE 8, the IFPUG method produces a MAPE of cca. 45 % and PRED (25) of 0.49. When compared to the CVS, this means that the estimation capability is increased by 4 % (PRED). When MAPE is compared to the new CVS model, it reduces its value by nearly 11 %. Another interesting aspect is the tendency to over/under estimate. The IFPUG method underestimates the majority of observations in the Testing dataset (142 of 184); whereas the CVS is not biased as regards overestimation or underestimation.

RQ2 asked if the CVS model performs better than Clustering methods. As can be seen from TABLE 8, when clustering is used in an estimation process, it decreases estimation errors, but the CVS still produces a more accurate estimation. The CVS model achieves a lower MAPE value for 34 % vs 41 % for Spectral Clustering. The same behavior is observed when Regression Clustering is used. There is a lower MEE - but MAPE (92 %), shows that models do not perform well. This is confirmed by the third criterion – PRED (0.25) = 0.36 for Regression Clustering shows that the CVS model is more accurate.

Answering RQ3, it can be declared that CVS works the best when Relative Size is used for data-set segmentation (see TABLE 7). Relative Size outperforms the Industry Sector by 17 % and the Business Area Type by 14 %, when MAPE is considered. This finding is derived by MEE, which is significantly lower for the Relative Size parameter than for others (see TABLE 7).

A. THREATS TO VALIDITY

The main treats to validity relate to the ISBSG data-set usage. The ISBSG is the only data-set available in which categorical variables are included. The Holdout Approach and Data-set Cleaning Methods were used to decrease this threat. In the ISBSG data-set, all entries are marked with quality labels (A-D grades). In this study, only the A and B labels were used. The quality labels evaluate the counting process and the data submission procedure. In this study, the authors expected that standard IFPUG procedures would be used for counting. This may limit the results when another data-set is used.

The Spectral Clustering method is used in variants, which are based on k-means. This leads to the question of setting the number of clusters. In k-means, the number of clusters have to be pre-defined before clustering begins. There are several hyper-parametric tuning methods that can be used for the pre-definition of the number of clusters. In this study no such method was applied because the number of clusters was selected according to evaluation criteria that deal with estimation accuracy.

VII. CONCLUSION

The new CVS model was introduced in this study. This algorithm is based on data-set segmentation, where Relative Size is used as a segmentation parameter. This approach allows one to estimate the software development effort by using a specific model, trained on a specific data segment. This approach outperforms all of the tested baseline methods and leads to the simplification of the estimation process. This study showed that clustering-based models are outperformed by the new proposed CVS model.

To conclude, research results based on RQs, it can be said that all tested methods perform better than the IFPUG approach itself. The CVS model works best when a Relative Size variable is used for segmentation. Its MAPE value is 34 % and its PRED (25) is 0.53. Both values demonstrate that the proposed model outperforms all of the baseline methods and confirms its practical applicability in the software industry.

In future research, Clustering and Segmentation methods will be subject to further investigation. Regression Clustering was expected to perform well in conjunction with SR Model Estimation, but it does not. Therefore - in future research, a new method of elimination observation will be under investigation.

VIII. DATA AVAILABILITY

The ISBSG data used to support the findings of this study may be released upon application to the International Software Benchmarking Standards Group (ISBSG) [7], which can be contacted at admin@isbsg.org or http://isbsg.org/academic-subsidy

REFERENCES

[1] F. González-Ladrón-de-Guevara, M. Fernández-Diego, and C. Lokan, "The usage of ISBSG data fields in software effort estimation: A systematic mapping study," *J. Syst. Softw.*, vol. 113, pp. 188–215, Mar. 2016.

[2] T. Xia, J. Chen, G. Mathew, X. Shen, and T. Menzies. (2018). "Why software effort estimation needs SBSE." [Online]. Available: https://arxiv.org/abs/1804.00626

[3] R. Silhavy, P. Silhavy, and Z. Prokopova, "Evaluating subset selection methods for use case points estimation," *Inf. Softw. Technol.*, vol. 97, pp. 1–9, May 2018.

[4] A. Z. Abualkishik and L. Lavazza, "IFPUG function Points to COSMIC function Points convertibility: A fine-grained statistical approach," (in English), *Inf. Softw. Technol.*, vol. 97, pp. 179–191, May 2018.

[5] R. Silhavy, P. Silhavy, and Z. Prokopova, "Analysis and selection of a regression model for the use case points method using a stepwise approach," *J. Syst. Softw.*, vol. 125, pp. 1–14, Mar. 2017.

[6] M. Azzeh, A. B. Nassif, and S. Banitaan, "Comparative analysis of soft computing techniques for predicting software effort based use case points," (in English), *IET Softw.*, vol. 12, no. 1, pp. 19–29, Feb. 2018.

[7] ISBSG. *ISBSG Development & Enhancement Repository Release 13*. Accessed: Feb. 2015. [Online]. Available: http://isbsg.org

[8] L. L. Minku and X. Yao, "Can cross-company data improve performance in software effort estimation?" in *Proc. 8th Int. Conf. Predictive Models Softw. Eng.*, 2012, pp. 69–78.

[9] A. Z. Abualkishik *et al.*, "A study on the statistical convertibility of IFPUG function point, COSMIC function point and simple function point," (in English), *Inf. Softw. Technol.*, vol. 86, pp. 1–19, Jun. 2017.

[10] M. Azzeh and A. B. Nassif, "Analyzing the relationship between project productivity and environment factors in the use case points method," (in English), *J. Softw., Evolution Process*, vol. 29, no. 9, p. e1882, Sep. 2017.

[11] A. Idri, F. A. Amazal, and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," (in English), *Inf. Softw. Technol.*, vol. 58, pp. 206–230, Feb. 2015.

[12] C. Lokan and E. Mendes, "Investigating the use of duration-based moving windows to improve software effort prediction," (in English), in *Proc. 19th Asia–Pacific Softw. Eng. Conf. (Apsec)*, vol. 1, Dec. 2012, pp. 818–827.

[13] V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "Increasing the accuracy of software development effort estimation using projects clustering," (in English), *IET Softw.*, vol. 6, no. 6, pp. 461–473, Dec. 2012.

[14] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "Neural network models for software development effort estimation: A comparative study," (in English), *Neural Comput. Appl.*, vol. 27, no. 8, pp. 2369–2381, Nov. 2015.

[15] M. Azzeh and A. B. Nassif, "A hybrid model for estimating software project effort from use case points," *Appl. Soft Comput.*, vol. 49, pp. 981–989, Dec. 2016.

[16] J. J. C. Gallego, D. Rodríguez, M. A. Sicilia, M. G. Rubio, and A. G. Crespo, "Software project effort estimation based on multiple parametric models generated through data clustering," (in English), *J. Comput. Sci. Technol.*, vol. 22, no. 3, pp. 371–378, 2007.

[17] M. Garre, J. J. Cuadrado, M. A. Sicilia, M. Charro, and D. Rodríguez, "Segmented parametric software estimation models: Using the EM algorithm with the ISBSG 8 database," in *Proc. 27th Int. Conf. Inf. Technol. Interfaces*, 2005, pp. 181–187.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[19] J. Hihn, L. Juster, J. Johnson, T. Menzies, and G. Michael, "Improving and expanding NASA software cost estimation methods," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–12.

[20] Z. Prokopová, R. Silhavy, and P. Silhavy, "The effects of clustering to software size estimation for the use case points methods," in *Software Engineering Trends and Techniques in Intelligent Systems* (Advances in Intelligent Systems and Computing), vol. 575. Springer, Apr. 2017, pp. 479–490.

[21] V. Khatibi Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons," (in English), *Empirical Softw. Eng.*, vol. 19, no. 4, pp. 857–884, Aug. 2014.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," (in English), in *Proc. Int. Conf. Neural Netw.*, vol. 4, pp. 1942–1948, Nov./Dec. 1995.

[23] C. Lokan and E. Mendes, "Applying moving windows to software effort estimation," (in English), in *Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, 2009, pp. 111–122.

[24] S. Amasaki and C. Lokan, "The effect of moving windows on software effort estimation: Comparative study with CART," (in English), in *Proc. 6th Int. Workshop Empirical Softw. Eng. Pract. (IWESEP)*, 2014, pp. 1–6.

[25] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, Jun. 2016.

[26] A. J. Albrecht, "Measuring application development productivity," in *Proc. IBM Appl. Develop. Joint SHARE/GUIDE Symp.*, Monterey, CA, USA, 1979, pp. 83–92.

[27] M. Bundschuh and C. Dekkers, *The IT Measurement Compendium: Estimating and Benchmarking Success With Functional Size Measurement*. Springer, 2008.

[28] S. Ezghari and A. Zahi, "Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method," *Appl. Soft Comput.*, vol. 67, pp. 540–557, Jun. 2018.

[29] F. Sarro and A. Petrozziello, "Linear programming as a baseline for software effort estimation," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, p. 12, 2018.

[30] P. Silhavy, R. Silhavy, and Z. Prokopová, "Evaluation of data clustering for stepwise linear regression on use case points estimation," in *Software Engineering Trends and Techniques in Intelligent Systems* (Advances in Intelligent Systems and Computing), vol. 575. Springer, Apr. 2017, pp. 491–496.

[31] J. P. Stevens, "Outliers and influential data points in regression analysis," *Psychol. Bull.*, vol. 95, no. 2, p. 334, 1984.

[32] D. S. Jayakumar and A. Sulthan, "A new procedure of regression clustering based on Cook's D," *Electron. J. Appl. Stat. Anal.*, vol. 8, no. 1, pp. 13–27, 2015.

[33] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[34] R. Silhavy, P. Silhavy, and Z. Prokopová, "Improving algorithmic optimisation method by spectral clustering," in *Software Engineering Trends and Techniques in Intelligent Systems* (Advances in Intelligent Systems and Computing), vol. 575. Springer, Apr. 2017, pp. 1–10.

[35] J. Hihn, L. Juster, J. Johnson, T. Menzies, and G. Michael, "Improving and expanding NASA software cost estimation methods," in *Proc. IEEE Aerosp. Conf.*, New York, NY, USA, Mar. 2016, pp. 1–12.

[36] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès, "Robust subspace clustering," *Ann. Statist.*, vol. 42, no. 2, pp. 669–699, 2014.

PETR SILHAVY was born in Vsetín, Czech Republic, in 1980. He received the B.Sc., M.Sc., and Ph.D. degrees in engineering informatics from the Systems Department, Faculty of Applied Informatics, Tomas Bata University in Zlín, in 2004, 2006, and 2009, respectively. From 1999 to 2018, he was a CTO in a company specialized on database systems development. Since 2009, he has been a Senior Lecturer with the Tomas Bata University in Zlín.

His major research interests include software engineering, empirical software engineering, system engineering, data mining, and database systems.

RADEK SILHAVY was born in Vsetín, Czech Republic, in 1980. He received the B.Sc., M.Sc., and Ph.D. degrees in engineering informatics from the Faculty of Applied Informatics, Tomas Bata University in Zlín, in 2004, 2006, and 2009, respectively.

He is a Senior Lecturer and a Researcher with the Computer and Communication Systems Department. His major research interests include effort estimation in software engineering, empirical software engineering, and system engineering.

ZDENKA PROKOPOVA was born in Rimavská Sobota, Slovakia, in 1965. She received the master's degree in automatic control theory and the Technical Cybernetics Doctoral degree from Slovak Technical University, in 1988 and 1993, respectively.

From 1988 to 1993, she was an Assistant with Slovak Technical University. From 1993 to 1995, she was a Programmer of database systems with Datalock business firm. From 1995 to 2000, she was a Lecturer with the Brno University of Technology. Since 2001, she has been with the Faculty of Applied Informatics, Tomas Bata University in Zlín. She is currently an Associate Professor with the Department of Computer and Communication Systems. Her research interests include programming and applications of database systems, mathematical modeling, computer simulation, and the control of technological systems.

• • •