IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 September 2023, accepted 29 October 2023, date of publication 1 November 2023,
date of current version 15 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3329369

RESEARCH ARTICLE

Evaluating Kernel Functions in Software Effort Estimation: A Comparative Study of Moving Window and Spectral Clustering Models Across Diverse Datasets

PETR SILHAVY AND RADEK SILHAVY
Faculty of Applied Informatics, Tomas Bata University in Zlín, 76001 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 RVO/FAI/2021/002.

ABSTRACT This study embarks on an in-depth analysis of the performance of various kernel functions, namely uniform, epanechnikov, triangular, and gaussian, in window-based and spectral clustering-based models. Employing seven distinct datasets, our approach evaluated both window sizes (25%, 50%, 75%, and 100%) and clustering clusters (ranging from 1 to 4). The kernel functions served as weighting functions for regression models, leading to the creation of 192 window-based and 192 clustering-based models. Our analysis underscores the dominance of the uniform kernel function. In most models where the Pred(0.25) was maximal and the Mean Absolute Error was minimal, the uniform kernel function was predominantly utilized. Further, our results exhibit varying outcomes between moving windows and spectral clustering across datasets. For instance, in the fpa_china dataset, while moving windows with a 50% size displayed no significant superiority over spectral-clustering with 1 cluster, spectral-clustering (1 cluster) demonstrated a significantly enhanced performance. However, in datasets like fpa_kitchenham, neither approach proved to be significantly better. This comprehensive exploration into the efficiency of kernel functions in moving windows and spectral-clustering models provides valuable insights for future research and applications in data modelling and analysis.

INDEX TERMS Software effort estimation, kernel function, moving windows, spectral clustering, functional points, use case points.

I. INTRODUCTION

The effort estimation processes and algorithms are important for software project scheduling, resource allocation, or budgeting. Accurate estimation of efforts is crucial for software development management and strategic planning.

Essential topics under investigation include evaluating whether global or local cost functions are more accurate. The effort estimation algorithm can benefit from using the local cost function [1], [2] - the local data subset/segment is used for model training. One proposed approach is the moving window principle [3], [4], which is based on instances that represent recently completed software projects [5], [6]. Another assumption describes the influence of recent cases, which do not equally influence new estimations; therefore, weights are applied to those instances. The windows can be formed using a duration interval [7] or the number of recently completed projects. The move-in/move-out approach [8] can be applied to keep the window size uniform.

The investigations of Lokan and Mendes [9] represent different approaches to searching for similar projects. These researchers showed that moving windows are helpful for a subset selection technique. This approach is based on the idea that previous projects with similar completion times allow the creation of a better estimation model. In recent research data locality is investigated not only from the moving

The associate editor coordinating the review of this manuscript and approving it for publication was Hailong Sun.

windows or weighted moving windows perspective but also from clustering-based options.

Alqasrawi et al. [10] investigate local weighted regression to estimate software effort. The main approach is similar to moving windows; to create local data/segments for which an estimation algorithm is trained. A similar approach using stepwise regression is presented in [11]. Alqasrawi et al. present the effects of choosing kernels, polynomial degrees, and bandwidth parameters; non-uniform kernel methods and small polynomial degrees are considered best performers.

II. RESEARCH QUESTIONS

In this paper, we will investigate the impact of different moving windows (and weighted windows) approaches. Many studies used different datasets, different evaluation metrics, and research methodology. This paper aims to evaluate moving windows on seven datasets with four kernels (weighting functions) and help understand if moving windows impact effort estimation systematically. The research question that needs to be answered:

RQ1: Can a superior kernel function be identified for all datasets and predictor sets?

RQ2: Can moving windows be compared to spectral clustering in the ability to estimate the minimised error?

The statistical significance of the models is evaluated using the Wilcoxon rank sum test. The Wilcoxon test [5], [12] is used as a test of the null hypothesis that the means (μ) of two models' estimation errors are equal.

This paper compares the accuracy of the windows-based model with that of the spectral-clustering-based model using Wilcoxon rank sum test. The Wilcoxon test tests the null hypothesis that the mean value - median (μ) of two normally distributed populations are equal. The Wilcoxon test will be used for the evaluation of estimation errors of windows-based models (μ_{MW}) and spectral clustering-based model (μ_{SC}).

$H_0 : \mu_{MW} = \mu_{SC}$, there is no difference in prediction ability between windows-based and clustering-based models. There is no difference in mean error values.

Alternative hypothesis:

$H_1 : \mu_{MW} > \mu_{SC}$, there is a difference in prediction capability between windows-based and cluster-based models. There is statistically significant evidence that SC brings better accuracy and lower estimation error.

$H_2 : \mu_{SC} > \mu_{MW}$, there is a difference in prediction ability between windows-based and clustering-based models. There is statistically significant evidence that MW brings better accuracy and lower estimation error.

III. CONTRIBUTION

This study contributes to the effort estimation research community in the following aspects:

- Comprehensive Evaluation of Kernel Functions: The paper presents a systematic study of kernel functions' effectiveness in moving weighted windows (MW) across seven diverse datasets. This comprehensive evaluation gives readers valuable insights into which kernel functions might suit their respective datasets.
- Comparative Analysis with Spectral Clustering: By juxtaposing MW with spectral clustering (SC), the paper offers a comparative perspective, allowing readers to understand the relative merits and demerits of the two approaches.
- Window Size Analysis: The study analyzes the impact of varying window sizes on the effectiveness of the MW approach. By considering four distinct window sizes (25, 50, 75, and 100%), the research offers a granular understanding of how data utilization impacts the efficacy of the MW method.
- Preference for the Uniform Kernel: The research identifies a clear trend favouring the uniform kernel function in most evaluated models. This finding can guide future researchers and practitioners in selecting kernel functions for similar tasks.
- Foundational Insights for Future Work: The research provides a foundational understanding of the benefits, limitations, and best practices associated with kernel functions in MW. These insights can inform and guide future research and applications in this domain.

In summary, this paper advances the understanding of kernel functions in moving weighted windows, offers comparative insights with spectral clustering, and provides clear guidance on the optimal use of window sizes and kernel functions.

The present paper is organised as follows. Section IV presents related works. The background of the research and the approaches used are detailed in Section V. Research methods and design are described in Section VI. In Section VII the results are summarised and discussed. Finally, Section VIII is a conclusion.

IV. RELATED WORK–OTHER APPROACH FOR DATASET SEGMENTATION (DATA LOCALITY)

Idri et al. [13] investigate more than 60 papers published between 1990 and 2012. Most of these research papers are focused on creating data locality using clustering.

Clustering is based on looking at similarities in instances. Data locality is understood as the creation of a group of similar projects. The local cost function is then created on (usually) more consistent instances. Azzeh et al. addressed the issue of determining the number of projects nearby using a method known as bisecting k-medoids [14]. It is shown in [15] that grouping of varied projects into clusters can aid in accurate evaluation. In this paper, it is confirmed that clustering improves estimate accuracy. In [6], the authors compared moving windows and spectral clustering. As shown, spectral clustering excels compared to moving windows or the k-means approach. In [16], a hybrid model with classification and prediction phases is presented employing a support vector machine and radial basis neural networks. The authors

of [17] established a technique to estimate the elicitation of equations by dividing historical projects. Garre et al. [18] describe a beneficial effect of the improved expectation maximisation method (EM). The updated EM method was introduced by Dempster et al. [19]. Hihn et al. [20] proved that SC produces fewer outliners than the closest-neighbour approach.

The locality of the data omits inconsistent projects and improves the accuracy of the estimation, as described by Bardisiri et al. [21]. In [22] and [23] authors present the usage of the particle swarm optimization algorithm.

Prokopova et al. [24] showed the importance of selecting the type of clustering and the distance metric. Furthermore, the k-means algorithm exhibits better performance than the hierarchical clustering method.

Azzeh and Nassif [25], deal with setting the number of nearest projects. These authors recommend a method called Bisecting k-medoids Clustering and have claimed that this method is better than common ASEE methods.

Azzeh et al. in their paper [16], present a hybrid model that consists of classification and prediction stages using a Support Vector Machine and Radial Basis Neural Networks. They compare the said model with the k-medoids. They recommend that ECF be omitted from the estimation and to focus all estimation on the productivity factor, which represents the ratio between UCP, and development effort in person-hours.

Bardisiri et al. [22], declare that clustering significantly affects the accuracy of development effort estimation because it allows one to omit irrelevant projects from historical data points.

Prokopova et al. [24], compare k-means, hierarchical, and density-based clustering techniques with three different distance metrics. The results show that all tested clustering techniques improve estimation accuracy and that the number of clusters plays a significant role. It is important to select the clustering type and the distance metric correctly. The authors show that hierarchical clustering has produced an inappropriate distribution of clusters and therefore cannot be used.

In [26], Bardisiri et al. introduce the particle swarm optimisation (PSO) algorithm [23] in effort estimation. They introduced a weighting system in which the project attributes of different clusters are given different weights. This approach supports comparing a new project only with projects located in related clusters, based on similarity measures. Like other methods where a subset is selected, this method deals with setting the correct value of k of the nearest project. Hihn et al. [20], described that the nearest-neighbour method has significantly more outliers than spectral clustering does.

Lokan and Mendes [9], investigations showed that moving windows are helpful as a subset selection technique. Using 75 of the most recent projects for a new estimate makes this estimate more accurate than using all available data points.

V. BACKGROUND

Moving windows (MW) can be formed in two basic approaches. The instances (representing finished projects) in specific timeframe (e.g., last month, year, etc.) are included or a specific number of projects (6,30,75, etc.). [9]. The variable will be used depending mainly on the available instances. If there are a lot of finished projects in the portfolio, time duration windows can be used. Otherwise, setting a window size using the number of past projects is an option for small portfolios.

Another aspect is related to portfolio growth. In [8], the authors concluded that updating the training dataset by a recent project is beneficial. Therefore, only the recent project number corresponding to the size of duration-based windows or the size in the number of projects can be kept.

The question of window size was studied in several papers [3], [27]. In [3] windows, the size of 75 projects is recommended. The size of windows depends on the portfolio size, which, of course, varies. Therefore, dynamic sizing, based on proportion (perceptual) window size determination, is used.

A. WEIGHTED MOVING WINDOWS

Moving windows as described uses historical instances equally (concerning the window size or duration). The weighting approach can improve an estimation model by defining instances as more or less influencing estimation error [28], [29], [30]. Weighted moving windows can take several approaches to weight setting. A least square regression method is used in practice [6] and in [30], where the kernel functions are recommended. Kernel functions are triangular (1), epanechnikov (2), gaussian (3) and rectangle/uniform function (4). The rectangle function produces uniform weights.

$$W\left(n_i\right) = 1 - |x|, |x| < 1 \tag{1}$$

$$W\left(n_i\right) = 1 - x^2, |x| < 1 \tag{2}$$

$$W\left(n_i\right) = exp\left(-\frac{(2.5 * \mathrm{x})^2}{2}\right) \tag{3}$$

$$W\left(n_i\right) = 1, |x| < 1 \tag{4}$$

where $W\left(n_i\right)$ is an instance weight, n_i represents instance and x (5) represent order of an observation.

$$x = \frac{n_i}{n} \tag{5}$$

where n is a number of instances in windows (window size).

B. KERNEL FUNCTIONS

Kernel functions (1-4) are used, for instance, weight calculation. The weight is relative to the size of the windows. Instances are understood to be chronologically ordered in both duration- and size-based windows. Figure 1 shows an instance weight development for the used kernel function. It shows development in the window from recent to past.

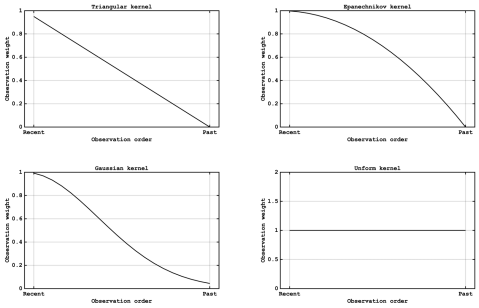

FIGURE 1. Kernel function and weights development.

C. STEPWISE REGRESSION

Stepwise regression (StepR) was used in [31]. StepR method was introduced in [11] and [32]. StepR is looking for the best combination of independent variables. The StepR approach can be described as follows:

1) Create an initial model by specifying the variables.
2) Set the constraints of the final model set the desired model complexity - linear, quadratic, interaction, etc.
3) Set the control threshold. The goal is to decide whether to remove or add another variable.
4) Re-test the model after adding or removing variables.
5) StepR stops when there is no further estimated progress.

Given a collection of independent variables, StepR generates many models. A model in the described approach is a multiple linear regression. The algebraic form is presented in (6).

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i \qquad (6)$$

where $i = 1, \ldots n$, y_i is the dependent variable; $X_{i1} \ldots X_{ip}$ are predictors; $\beta_1 \ldots \beta_n$ are regressors and β_0 is an intercept, and ε_i is a residual.

D. SPECTRAL CLUSTERING

Spectral Clustering (SC) is used as described in [31] and [33]. The technique is described in [6] and [33] and is based on a graphical representation in which each data point is a node, and the edges between the data points signify similarity. In SC, the k-nearest neighbour graph, ε - neighbourhood graph, and the fully connected graph are commonly employed [34]. The k-nearest neighbour graph connects the vertices v_i and v_j, where v_j is one of v_i'k-nearest neighbours. The ε - neighbourhood graph joins all data points with pairwise distances less than ε. The adjacency matrix W (7) is as follows:

$$W = \left(w_{ij}\right) \qquad (7)$$

where $i, j = 1..n$ and each cell in the matrix represents the edge weight between data points. There is no relationship between the edges if the weight is 0. After that, a Laplacian matrix (8) is computed:

$$L = D - W \qquad (8)$$

where the diagonal matrix D represents the vertex degree v_i. Spectrum calculation is a key stage in SC. It assumes the shape of the L matrix. There are two alternatives for the Normalised Laplacian algorithm: a symmetric matrix or a random matrix. The eigenvectors of a sorted list matrix are represented by the spectrum. An eigenvector represents a data point and an eigenvalue of a matrix. SC employs these eigenvectors as a feature. Finally, the spectrum is subjected to the clustering process. Although k-means are used in this paper, any clustering technique can be used.

VI. RESEARCH METHODS

A. DATASETS DESCRIPTION AND TYPICAL ISSUES

All the datasets that are used are known in the community and the majority of them are open to everyone. Research in the estimation of software effort is based on historical data [35]. Practical experience reveals that estimations frequently forecast a higher effort than has previously been experienced by a specific organisation. However, this does not always imply that the estimations utilised were inaccurate. Past data may be reported and recorded appropriately. To improve effort estimation in software projects (particularly in large corporations, you may need to record post-project maintenance and improvement operations throughout the software life. There may be significant errors in the estimate of the effort due to inconsistent reporting and recording of various parts of the overall effort in the dataset.

In this study, the following datasets were used:

- China Dataset (fpa_china) [36]
- Evidence-Based Software Portfolio Management Research Repository (fpa_EBSPM) [37]
- International Software Benchmarking Standards Group (fpa_isbsg2020) [38]
- Kitchenham dataset (fpa_kitchenham) [36]
- Maxwell dataset (fpa_maxwell) [36]
- Use Case Points 28 (ucp_28) [12]
- Use Case Points 71 (ucp_71) [11]

In fpa_isbsgdataset more than 8 thousand projects are recorded. The dataset was reduced to approximately 1,7 thousand instances. The only project described by the IFPUG features can be used in this study. Dataset noted as fpa_china contains 499 instances described with 14 features. In fpa_EBSPM 492 instances is recorded, but only 22 of them have been reported with actual effort. For instances where effort has been missing, the imputation of effort by the mean value of the productivity factor has been used. Another dataset called fpa_kitchenham contains 145 instances described by 10 features.

Finally, a fpa_maxwell dataset consists of 63 instances described by 25 features. In Table 1 dataset statistics based on effort features are described. Figure 2 shows the effort feature, log-transformed, for comparison of the effort size.

Datasets are chronologically ordered to keep or simulate order from past to news instances, which is important for moving windows approach. The following features are

TABLE 1. Dataset paparemetrs summary, based on effort.

Dataset	Sizing method	Instances	Features	Mean	Median	Min	Max
fpa_china	FP	499	14	3, 921	1,829	26	54,620
fpa_EBSPM	FP	492	37	5,889	2,048	31	279,976
fpa_isbsg	FP	1711	58	4,907	2,393	10	150,040
fpa_kitchenham	FP	145	10	3,113	1,557	219	113,930
fpa_maxwell	FP	63	25	8,109	5,100	583	63,964
ucp_28	UCP	28	7	1,210	1,115	191	2,652
ucp_71	UCP	71	18	3,856	3,884	2,602	5,350

used to keep data chronologically ordered (depending on availability in each of the datasets):

- fpa_china – feature Duration, no starting or ending date is available.
- fpa_EBSPM - feature Year_technical_go_live.
- fpa_isbsg– feature YearOfProject is used for simulated chronological ordering.
- fpa_kitchenham – feature Actualstartdate is used.
- fpa_maxwell – feature duration_months is used for simulated chronological ordering.
- ucp_28 – was used as already order by Project_No. No specific starting or ending date available.
- ucp_71 – chronological order already applied in Project_No. Instances ids represent ordering by project ending date.

Dataset ucp_28 was used in [12] and is based on previously published research papers [39] and by [40]. Another UCP based dataset – ucp_71 was first used in [11] and can be found in several following [6], [24], [32].

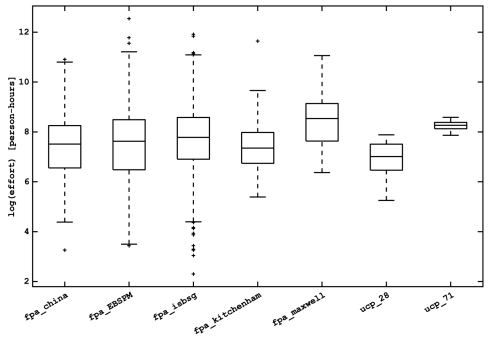

FIGURE 2. Dataset used in study (log-transformed effort).

Datasets are used as presented by the authors. Dataset processing was limited to choosing projects that are described by features that need experiments. This is mainly true for fpa_isbsg dataset where the IFPUG-only project was selected. In Table 2 a used feature is summarised, including types and dataset belongness.

B. EVALUATION CRITERIA

The kernel functions are studied in a manner of estimation ability/accuracy. There are many accuracy measurements used in effort estimation studies. Model evaluation will be held using a Mean Absolute Errors (MAE,(9)) [10], Pred(l) [41] (10) and adjusted coefficient of determination $\left(R^2_{Adj}\right)$ (11). Common criteria such as Magnitude of Relative Errors (MRE) or related criteria are not used because of their bias towards higher estimates [42].

$$MAE = \frac{1}{2}\sum_{i=1}^{n} |y_i - \widehat{y_l}| \tag{9}$$

$$Pred(l) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1 \; if \; \dfrac{|y_i - \widehat{y_l}|}{y_i} \leq l \\ 0 \; if \; \dfrac{|y_i - \widehat{y_l}|}{y_i} > l \end{cases} \tag{10}$$

$$R^2_{Adj} = 1 - (1 - R^2) * \frac{n-1}{n-k-1} \tag{11}$$

where n is the number of observations, y_i is the effort value, $\widehat{y_l}$ is the estimated value, and l is the error boundary as a percentage. If $l = 0.25$ is considered, the estimation error is less than or equal to 25% of the known effort. $Pred(l)$ value is between 0 and 1, where the higher value represents a better model. $PreR^2$ is the coefficient of determination, k is the number of independent variables and n is the number of observations. R^2_{Adj} shows the effect size of a selected set of independent variables on the dependent variable. During the calculation, the number of instances and the number of independent variables is taken into consideration. The value R^2_{Adj} can acquire a value of between 0 and 1, when higher values represent better solutions.

C. RESEARCH DESIGN

The objective of this study is to investigate the effect of kernel functions in moving windows. According to previous research, moving windows size of 15 instances is the best performer [6]. Evaluation is performed on 7 datasets. Datasets can be divided using estimation methods:

- Functional points datasets (FPD) - fpa_china, fpa_EBSPM, fpa_isbsg2020, fpa_kitchenham, fpa_maxwell
- Use case points datasets (UCD) – ucp_25 and ucp_71

As a dependent variable, which is available in all datasets:

- Effort – in person-hours, is used in the original scale, to keep it in natural.

As independent variables are used in sizing estimation components:

- For FPD (a) – EI, EO, EQ, ILF, EIF – those are available in only fpa_china, fpa_isbsg

TABLE 2. List of used features.

Feature	Type	fpa_china	fpa_EBSPM	fpa_isbsg	fpa_kitchenham	fpa_maxwell	ucp_28	ucp_71
EI	Independent	X		X				
EO	Independent	X		X				
EQ	Independent	X		X				
ILF	Independent	X		X				
EIF	Independent	X		X				
UAW	Independent						X	X
UUCW	Independent						X	X
TCF	Independent						X	X
ECF	Independent						X	X
SimpleActors	Independent							X
AverageActors	Independent							X
ComplexActors	Independent							X
SimpleUC	Independent							X
AverageUC	Independent							X
ComplexUC	Independent							X
Effort	Dependant	X	X	X	X	X	X	X
Size	Independent	X	X	X	X	X	X	X
YearOfProject	Ordering			X				
Year_technical_go_live	Ordering		X					
Duration	Ordering	X						
Actualstartdate	Ordering				X			
YearOfProject	Ordering		X					
duration_months	Ordering					X		
Project_No	Ordering						X	X

- For FPD (b) – Size – size in FP is available in all datasets - fpa_china, fpa_EBSPM, fpa_isbsg2020, fpa_kitchenham, fpa_maxwell
- For UCD (a) – UAW, UUCW, TCF, ECF – available for both dataset – ucp_25, ucp_71
- For UCD (b) – Size – available in both datasets – ucp_25, ucp_71
- For UCD (c) – SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC – available only in ucp_71 dataset.

For all cases, where more than one independent variable is used, the rescaling was adopted using the Min-Max approach (12).

$$\bar{x} = \frac{x - min(X)}{max(X) - min(X))} \tag{12}$$

where $\bar{x}$ represents the rescaled value (in the range from 0 to 1) x is the value of the independent variable and X is the set of all independent variable values.

Experiments are utilised by using stepwise regression (StepR) [5], [6], [31], [43], [44] for effort estimation. StepR methods perform regression and feature selection using the added or removed predictors. In total, 48 experiments will be performed and evaluated. To address a comparison of moving windows with the kernel function with the other method - spectral clustering (SC) [6], [44] is presented.

All models will be trained using the hold-out data set (2)-fold cross-validation), where 80% will be a training fold and 20% will be a testing fold. To simulate a real-life scenario, all models will be retrained after performing 1 new estimation. All metrics and tests are on initially created testing fold, but the training fold is expanded, and models are retrained after adding 1 instance to the training fold. In brief, estimation models are retrained after 1 instance is estimated. In the case of SC, the training fold is reclustered when/after 1 instance is estimated. Training sets are being expanded after 5 instances are estimated. Those instances are appended to the end of the training set to keep the chronological order of the datasets.

Datasets are variable in size; therefore, the dynamic sizing based on proportion (percentual) windows size determination is used (25%, 50%, 70% and 100%). Clustering was used from 1 to 4 clusters. Kernel functions, as described, are used to set the weights of instances in StepR models. Weights are used identically for windows and for spectral clustering. In total, 192 models for windows and 192 for spectral clustering were tested and evaluated.

D. LIMITATIONS AND THREATS TO VALIDITY

The results of the study should be considered with respect to important measures to validity. Validity is discussed with respect to the approach introduced in [45] as internal, external, and construct. Internal validity refers to the bias of the instances available in datasets. It is not known whether all datasets represent a population fully. The data origin, same as the timeframe of data gathering, vary from approximately mid-1980 to 2017. Datasets also vary in project size, duration,

and overall complexity, in this study to project size or duration is not used as an estimation driver. The project problem domain is also not considered in this study. All datasets were used as they are available in public, with exception to ISBSG dataset. ISBSG dataset (heavily discounted), is available upon registration to the academic programme.

Generally, not much is known about how the dataset were collected and processed before publication. External validity is related to generalising the results of this study. All datasets vary in available variables, which allows one to use two variants of predictors. Project origins from various industries and problem domains. All are not understood as "within a company" data, but as a general data set, which may not be as accurate. Another question of external validity is whether the findings of the study are applicable to current software development processes, considering that the projects evaluated in this study (depending on the data sets) were completed between mid-1980s and 2017. Many fundamental changes in software techniques, tools, and technologies have occurred throughout the long time covered by this research. This was not considered in this study, because such changes were slow and dynamic at various periods. In addition, a special parameter was set to account for timing information. Selected factors will be significant in the future to explain the variance in effort and productivity caused by differences in the characteristics of software development.

During study construction, the experiments were unified as much as possible to be comparable. Windows size and cluster count were set identical for each of the data sets, since predictors were only selected identical variables that are available in datasets. The measurements are simplified to MAE and PRED, which is generally understood to be not biased.

VII. RESULTS AND DISCUSSION

In this study a weighted moving window was evaluated and compared to spectral clustering. Both approaches are understood as the data locality approach, which is considered for 7 tested datasets. The results are presented as partial for functional points and use case points datasets. In each group, results are presented per dataset – due to MAE keeping the original scale.

A. KERNEL FUNCTION EFFECTS ON FUNCTIONAL POINTS DATASETS – WINDOWS AND SPECTRAL CLUSTERING

The first run of experiments testing models, which utilise StepR methods and FP based predictors (EI, EO, EQ, ILF, EIF). These can be run for fpa_china and fpa_isbsg. In Figure 3 results for fpa_china and MW can be seen. In Figure 4 results for fpa_isbsg dataset can be seen. Pred(0.25) demonstrates that for both datasets the needed level of accuracy cannot be achieved. As can be seen, MAE leads to the recommendation of a lower size of windows for all tested kernel functions for fpa_china dataset, but for fpa_isbsg dataset, the higher sizes can be preferred.

Figure 5 and Figure 6 show results using spectral clustering. As can be seen, there is no favourite solution. For fpa_china dataset less clusters seem to be better solution, whereas for fpa_isbsg, it can be seen that there are more similarities in results. When we used Pred(0.25) and MAE independently to obtain the most accurate models, the results are summarised in Table 3. Independent variables EI,EO,EQ,ILF,EIF were used for StepR, where models were trained for repeatable after next instances were estimated (for windows), for clustering models are re-trained in each cluster after re-clustering. Re-clustering were performed after the next-instance estimation. FDP(a) models (using EI,EO,EQ,ILF,EIF for estimation and clustering) are weak in is performance, as can be seen, a uniform kernel is dominant in the best results.

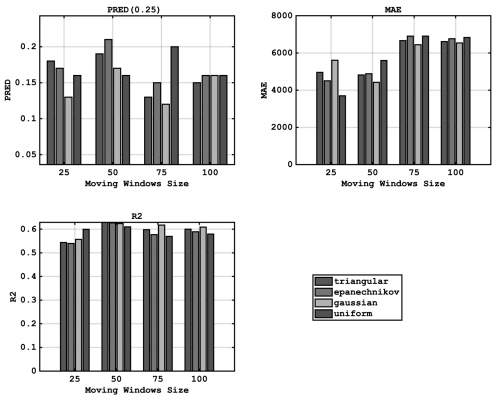

FIGURE 3. Kernel fucnction as weighting for moving windows for fpa_china dataset.

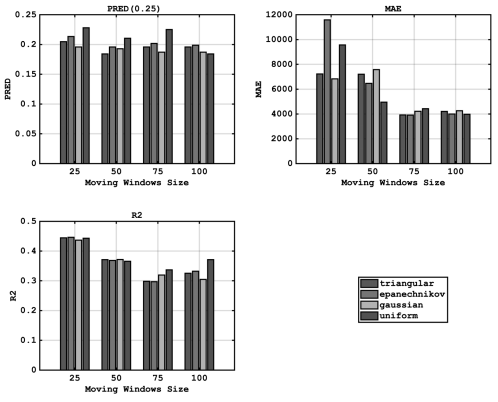

FIGURE 4. Kernel fucnction as weighting for moving windows for fpa_isbsg dataset.

The experimental group FDP(b) consists of all 7 datasets, because of independent variable size is used. Performance varies in each dataset (Table 4). For fpa_china (Figure 7) and fpa_isbsg (Figure 8) the overall performance is not dramatically different from the models, which are based on fp variables. Similar for clustering models (Figure 14 and Figure 13).

TABLE 3. Solution overview for FPD(a) models, where MAE is minimal or Pred(0.25) maximal.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2
fpa_china	uniform	25 %	EI,EO,EQ,ILF,EIF	3,697.80	0.16	0.60
fpa_china	epanechnikov	50 %	EI,EO,EQ,ILF,EIF	4,879,20	0.21	0.62
fpa_isbsg	epanechnikov	75 %	EI,EO,EQ,ILF,EIF	3,905.70	0.20	0.30
fpa_isbsg	uniform	25 %	EI,EO,EQ,ILF,EIF	9,566.60	0.22	0.44
fpa_china	epanechnikov	2 c	EI,EO,EQ,ILF,EIF	3,100.60	0.20	0.29
fpa_china	triangular	3 c	EI,EO,EQ,ILF,EIF	6,743.30	0.21	0.35
fpa_china	uniform	1 c	EI,EO,EQ,ILF,EIF	5,870.40	0.21	0.64
fpa_isbsg	uniform	2 c	EI,EO,EQ,ILF,EIF	3,589.90	0.19	0.23
fpa_isbsg	triangular	3 c	EI,EO,EQ,ILF,EIF	4,024.40	0.20	0.21

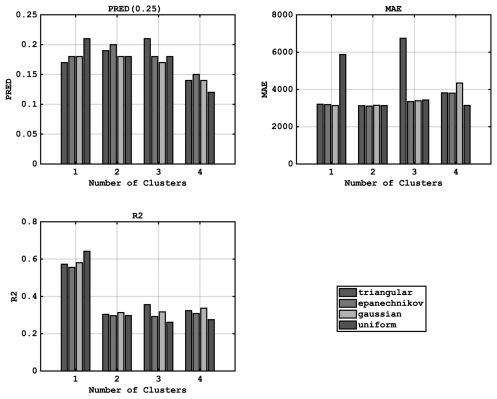

FIGURE 5. Spectral clustering for 1 to 4 clusters, kernel functions used in StepR for fpa_china.

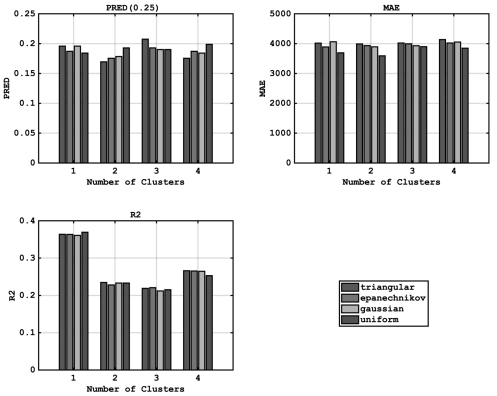

FIGURE 6. Spectral clustering for 1 to 4 clusters, kernel functions used in StepR for fpa_isbsg.

The model trained on fpa_EBSPM (Figure 9) shows similar performance for Pred and MAE, both choose different kernel function. When these are compared to SC models (Figure 14), then be seen to be lower (4,821 vs 7,407). Interestingly, using MAE 4 clusters excels, but when Pred (0.25) is considered, then 1 cluster (means no clustering) is better solution. On fpa_kitchenham dataset (Figure 10) windows size of 25% is recommended, by MAE and Pred(0.25). As can be seen in this data set, there are two solutions with identical Pred(0.25) value (0.51). When results clustering results for fpa_kitchenham (Figure 13) are evaluated using MAE and Pred(0.25) it can be seen that this dataset in only one, where both criteria recommend the same kernel function and the same number of clusters (no clustering).

Finally, an fpa_maxwell dataset (Figure 11) shows that data locality is beneficial here. For moving windows, a 25% (windows size) is a best option with a similar MAE value (as 100% windows size). When models for fpa_maxwell dataset are trained for clustering-based model (Figure 17), there can be seen that 4 cluster off the best solution and performance of triangular, epanechnikov and gaussian kernel functions. In all cases Pred(0.25) equals 0.46 and MAE is approx. 4,150.

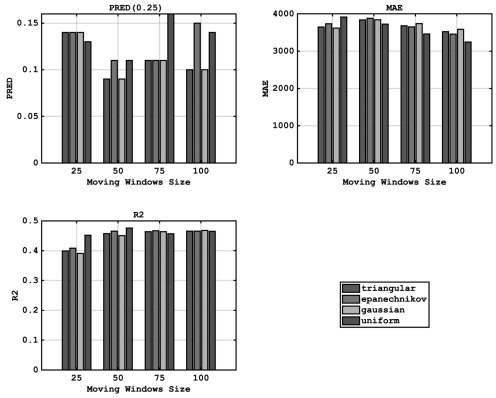

FIGURE 7. Kernel function (size as independent) for fpa_china.

B. KERNEL FUNCTION EFFECTS ON USE CASE POINTS DATASETS – WINDOWS AND SPECTRAL CLUSTERING

Two data sets based on use case points are written with the same StepR approach as the functional point dataset. Results are described and discussed for UCP parameters – UCD(a), for size as independent - UCD(b), and for number of actors and use cases (grouped by complexity level) – UCD(c). The variant UCD(c) is only presented for ucp_71, which contains the number of actors and use cases.

TABLE 4. Solution overview for FDP(b) models, where MAE is minimal or Pred(0.25) maximal.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2
fpa_china	uniform	100 %	size	3,245.10	0.14	0.46
fpa_china	uniform	75 %	size	3,459.50	0.16	0.45
fpa_isbsg	uniform	100 %	size	4,150.80	0.19	0.17
fpa_isbsg	triangular	50 %	size	4,353.10	0.20	0.21
fpa_isbsg	triangular	100 %	size	4,242.00	0.20	0.15
fpa_isbsg	epanechnikov	50 %	size	4,348.80	0.20	0.22
fpa_isbsg	gaussian	100 %	size	4,260.20	0.20	0.16
fpa_EBSPM	gaussian	25 %	size	7,407.80	0.36	0.51
fpa_EBSPM	epanechnikov	25 %	size	7,734.70	0.38	0.52
fpa_kitchenham	uniform	25 %	size	856.14	0.48	0.58
fpa_kitchenham	triangular	25 %	size	897.24	0.51	0.54
fpa_kitchenham	gaussian	25 %	size	899.57	0.51	0.53
fpa_maxwell	uniform	100 %	size	4,859.70	0.15	0.70
fpa_maxwell	uniform	25 %	size	5,454.20	0.30	0.72
fpa_china	uniform	2 c	size	2,781.60	0.19	0.33
fpa_china	uniform	2 c	size	2,781.60	0.19	0.33
fpa_china	uniform	3 c	size	2,799.40	0.19	0.23
fpa_isbsg	uniform	4 c	size	3,850.20	0.18	0.33
fpa_isbsg	gaussian	4 c	size	3,878.20	0.19	0.23
fpa_EBSPM	uniform	4 c	size	4,861.20	0.16	0.16
fpa_EBSPM	uniform	1 c	size	8,249.70	0.49	0.30
fpa_kitchenham	triangular	1 c	size	813.22	0.55	0.64
fpa_kitchenham	triangular	1 c	size	813.22	0.55	0.64
fpa_maxwell	uniform	4 c	size	3,935.10	0.31	0.40
fpa_maxwell	triangular	4 c	size	4,188.60	0.46	0.39
fpa_maxwell	epanechnikov	4 c	size	4,141.50	0.46	0.39
fpa_maxwell	gaussian	4 c	size	4,249.30	0.46	0.38

TABLE 5. Solution overview for UCD(a) models, where MAE is minimal or Pred(0.25) maximal.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2
ucp_28	uniform	75 %	UAW,UUCW,TCF,ECF	568.17	0.33	0.76
ucp_28	gaussian	25 %	UAW,UUCW,TCF,ECF	2017.90	0.67	0.94
ucp_28	gaussian	50 %	UAW,UUCW,TCF,ECF	947.50	0.67	0.75
ucp_71	uniform	75 %	UAW,UUCW,TCF,ECF	548.65	0.85	0.03
ucp_71	uniform	100 %	UAW,UUCW,TCF,ECF	570.84	0.92	0.03
ucp_28	gaussian	4 c	UAW,UUCW,TCF,ECF	350.77	0.50	0.52
ucp_28	gaussian	1 c	UAW,UUCW,TCF,ECF	524.03	0.67	0.78
ucp_71	epanechnikov	3 c	UAW,UUCW,TCF,ECF	488.35	0.93	0.40
ucp_71	epanechnikov	2 c	UAW,UUCW,TCF,ECF	545.58	1.00	0.14

Two datasets were used for these experiments. In Table 5 results where MAE is minimal or Pred(0.25) is maximal. For ucp_28 (Figure 17) dataset is the best option with gaussian kernel function for windows size equal to 50% (MAE = 947.5 and Pred(0.25) = 0.67). Clustered (SC) solutions (Figure 19) recommends 4 clusters (MAE = 350.77 and Pred(0.25) = 0.5). If both criteria are evaluated individually, then there are several selected results.

Models, which are trained on ucp_71 weighted moving windows (Figure 18) achieve the best performance for uniform kernel function and windows size 75% (if Mae is considered) and for a size of 100% if Pred is considered. Epanechnikov kernel for weighting in StepR models is selected for ucp_71, where spectral clustering is used. Using MAE criterion 3 clusters can be selected and using Pred 2 clusters. Clustered models achieved better performance for both datasets (ucp_28, ucp_71).

Using size as an independent variable produces several solutions in a similar performance for both ucp datasets. In the case of ucp_28 there can be seen (in Table 6) three solutions where Pred(0.25) equals 1 when an MAE is counted, then a uniform kernel for windows size 100% can be selected (Figure 21). Clustering achieves (Figure 23)

TABLE 6. Solution overview for UCD(b) models, where MAE is minimal or Pred(0.25) maximal.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2
ucp_28	uniform	100 %	size	200.61	1.00	0.92
ucp_28	epanechnikov	100 %	size	219.87	1.00	0.94
ucp_28	uniform	50 %	size	215.51	1.00	0.95
ucp_28	uniform	100 %	size	200.61	1.00	0.92
ucp_71	uniform	75 %	size	483.4	0.92	0.29
ucp_71	triangular	75 %	size	502.92	0.92	0.26
ucp_71	triangular	100 %	size	498.39	0.92	0.30
ucp_71	epanechnikov	75 %	size	494.43	0.92	0.25
ucp_71	epanechnikov	100 %	size	501.47	0.92	0.30
ucp_71	gaussian	75 %	size	512.46	0.92	0.27
ucp_71	gaussian	100 %	size	499.89	0.92	0.28
ucp_71	uniform	50 %	size	522.2	0.92	0.17
ucp_71	uniform	75 %	size	483.4	0.92	0.29
ucp_28	uniform	2 c	size	189.74	0.83	0.75
ucp_28	triangular	3 c	size	215.12	1.00	0.60
ucp_28	epanechnikov	1 c	size	215.47	1.00	0.94
ucp_28	epanechnikov	3 c	size	218.01	1.00	0.63
ucp_28	gaussian	3 c	size	215.9	1.00	0.61
ucp_28	uniform	1 c	size	193.08	1.00	0.91
ucp_28	uniform	3 c	size	212.42	1.00	0.67
ucp_71	epanechnikov	3 c	size	450	0.92	0.02
ucp_71	triangular	1 c	size	504.69	1.00	0.29
ucp_71	epanechnikov	1 c	size	504.44	1.00	0.30
ucp_71	gaussian	1 c	size	506.51	1.00	0.26
ucp_71	uniform	3 c	size	459.92	1.00	0.03

TABLE 7. Solution overview for UCD(c) models, where MAE is minimal or Pred(0.25) maximal.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2
ucp_71	uniform	100 %	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	463.00	0.92	0.06
ucp_71	triangular	100 %	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	543.12	0.92	0.03
ucp_71	uniform	100 %	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	463.00	0.92	0.06
ucp_71	uniform	1 c	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	479.58	0.92	0.05
ucp_71	uniform	1 c	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	479.58	0.92	0.05
ucp_71	uniform	2 c	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	517.23	0.92	0.12
ucp_71	uniform	4 c	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	633.35	0.92	0.49

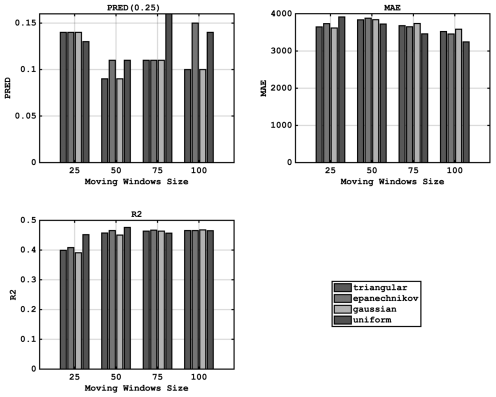

FIGURE 8. Kernel function (size as independent) for fpa_isbsg.

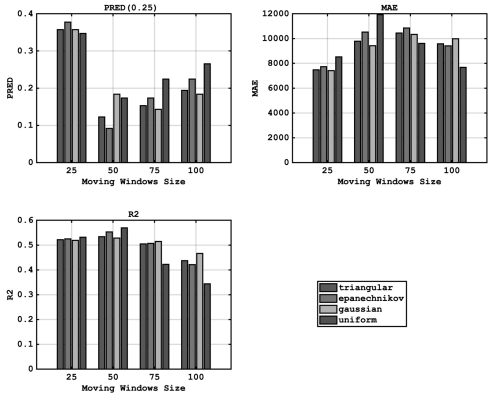

FIGURE 9. Kernel function (size as independent) for fpa_EBSPM.

similar performance for a uniform kernel (2 clusters) or for a triangular kernel (3 clusters).

Dataset ucp_71 shows similar performance for various combinations of window size and kernel function (Figure 22). The best seems to be the uniform kernel function for windows size of 75%. Clustering for ucp_71 brings an increase of Pred(0.25) and similar MAE levels (Figure 24). The most prominent appears to be 3 clusters solution and uniform kernel function.

Dataset ucp_71 allows to perform the last expiration, which uses the number of actors (grouped by complexity) and number of use cases (grouped by complexity) as independent

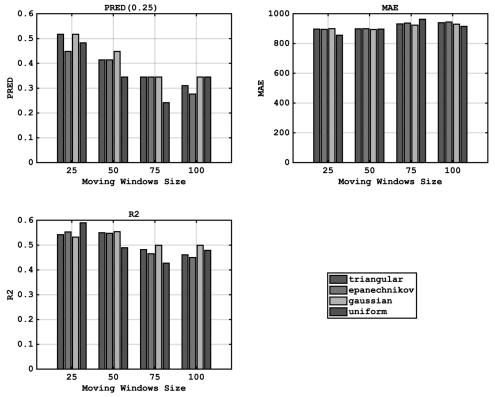

FIGURE 10. Kernel function (size as independent) for fpa_kitchenham.

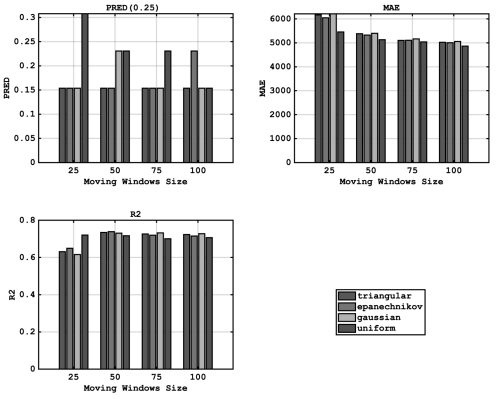

FIGURE 11. Kernel function (size as independent) for fpa_maxwell.

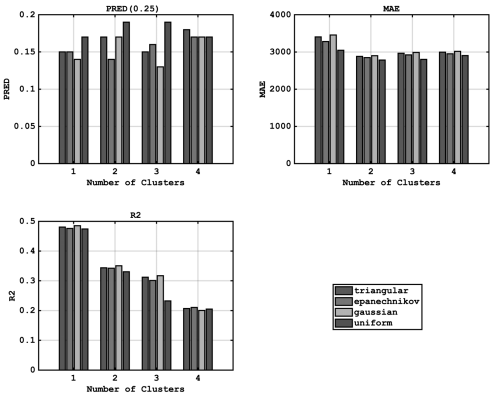

FIGURE 12. Spectral Clustering (size as independent and for clustering) for fpa_china.

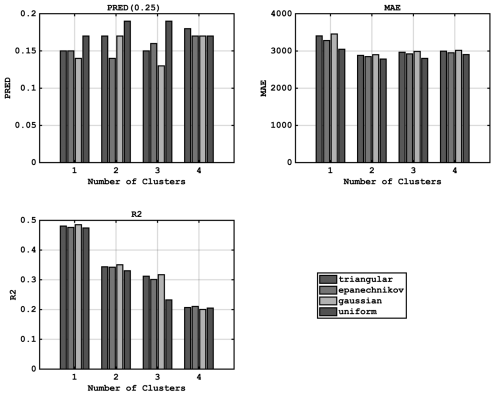

FIGURE 13. Spectral Clustering (size as independent and for clustering) for fpa_isbsg.

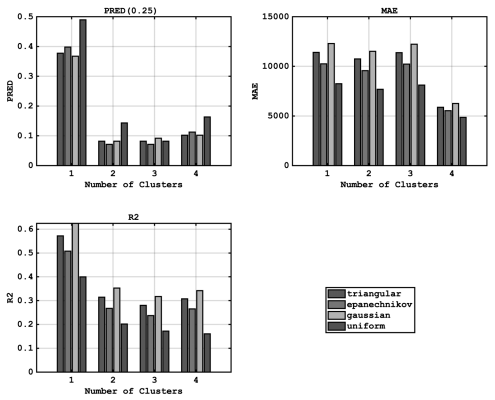

FIGURE 14. Spectral Clustering (size as independent and for clustering) for fpa_EBSPM.

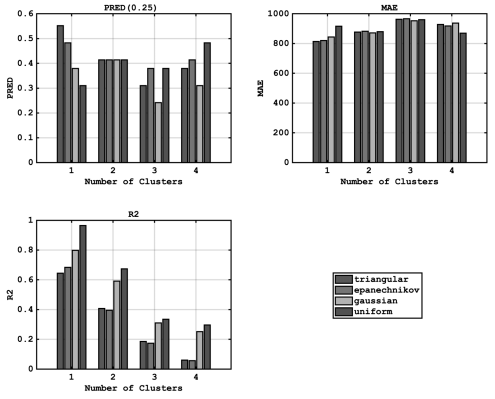

FIGURE 15. Spectral Clustering (size as independent and for clustering) for fpa_kitchenham.

variables for StepR models. In Table 7 it can be seen that Pred(0.25) is 0.92 for all selected models, which means that only MAE can be considered. Interestingly, weighted moving windows are not considered an optimal solution, windows size 100% (Figure 26) is recommended (MAE = 463.00). Clustering (Figure 25) does not improve the accuracy of the estimation. Overall, 4 clusters can be favourites, because of the higher R^2_{adj} value.

VIII. CONCLUSION

In this study, an effect of kernel function in moving weighted windows has been tested on 7 different datasets. The window sizes 25,50,75 and 100% have been tested. The percentage window size has been selected to allow reproducibility on datasets, which vary in size. Windows approach has been compared to spectral clustering, which has been evaluated

TABLE 8. Selected models for fpa_china.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
fpa_china	epanechnikov	50 %	EIEOEQILFEIF	4,879.20	0.21	0.62	0.99
fpa_china	uniform	1 c	EIEOEQILFEIF	5,870.40	0.21	0.64	0.01
fpa_china	uniform	75 %	size	3,459.50	0.16	0.45	0.99
fpa_china	uniform	2 c	size	2,781.60	0.19	0.33	0.00

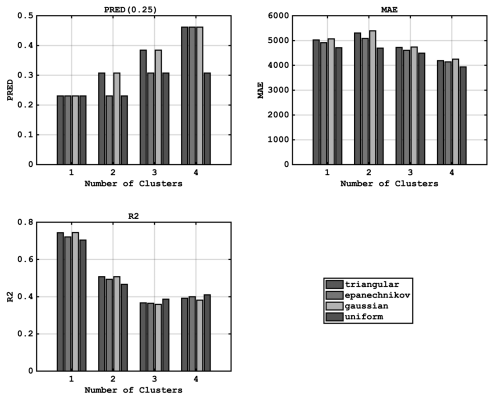

FIGURE 16. Spectral Clustering (size as independent and for clustering) for fpa_maxwell.

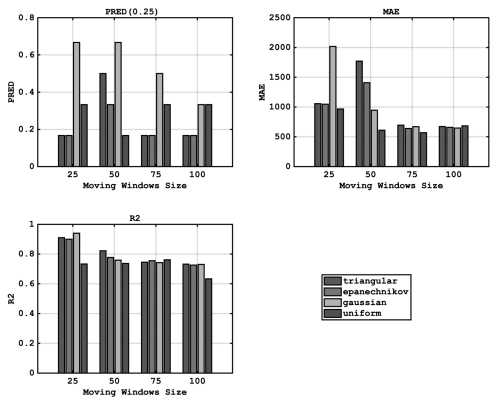

FIGURE 17. Kernel functions, UCP parameters for ucp_28 dataset.

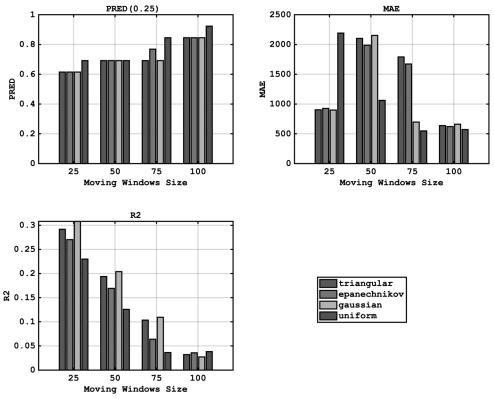

FIGURE 18. Kernel functions, UCP parameters for ucp_71 dataset.

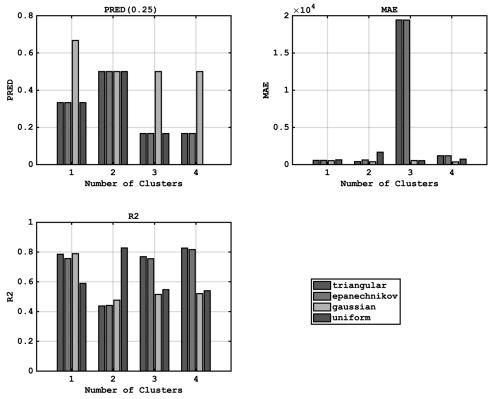

FIGURE 19. Spectral Clustering (UCP parameters as independent and for clustering) for ucp_28 dataset.

for number of clusters from 1 to 4, whereas 1 is considered a nonclustered option. The kernel function has been used as a weighting function regression model. In total, there were 192 windows-based models and 192 clustering-based models tested.

A. RQ1: CAN A SUPERIOR KERNEL FUNCTION BE IDENTIFIED FOR ALL DATASETS AND PREDICTOR SETS?

Addressing RQ1 the analysis, how frequently is each kernel function used. Comparison of models (Table 3 - Table 7) where Pred(0.25) is maximal or MAE is minimal can be seen In Figure 27, part (a). The comparison of the frequency of the kernel functions, where Pred(0.25) is maximal and then MAE is minimal (Table 8 - Table 14) can be seen in Figure 27, part (b).

In both views, the dominancy of uniform kernel function can be seen. In the majority of models, where Pred(0.25) is maximal and MAE minimal, we are using a uniform kernel function for weights in the MW and SC approaches too. If we compare kernel functions using Pred(0.25), it can be seen that uniform function can be understood as better than other kernel function (Figure 28), wherein part (a) boxplots illustrate all models where Pred(0.25) is maximal and MAE minimal (for MW). In part (b), all SC models can be seen. In parts (c) and (d) only the best-selected models for MW, respectively, SC can be seen. Only the top 10 models

TABLE 9. Selected models for fpa_isbsg.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
fpa_isbsg	uniform	25 %	EIEOEQILFEIF	9,566.60	0.23	0.44	0.97
fpa_isbsg	triangular	3 c	EIEOEQILFEIF	4,024.40	0.21	0.21	0.02
fpa_isbsg	triangular	100 %	size	4,242.00	0.20	0.15	1.00
fpa_isbsg	gaussian	4 c	size	3,878.20	0.19	0.23	0.00

TABLE 10. Selected models for fpa_EBSPM.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
fpa_EBSPM	epanechnikov	25 %	size	7,734.70	0.38	0.52	1.00
fpa_EBSPM	uniform	1 c	size	8,249.70	0.49	0.40	0.00

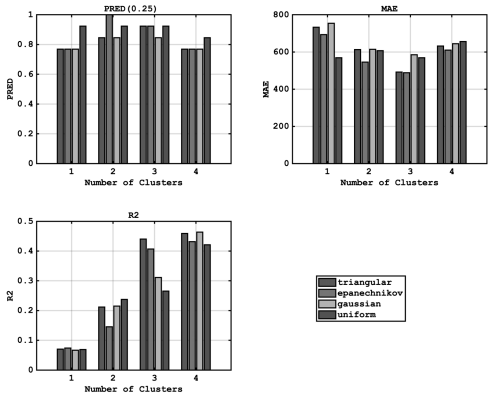

FIGURE 20. Spectral Clustering (UCP parameters as independent and for clustering) for ucp_71 dataset.

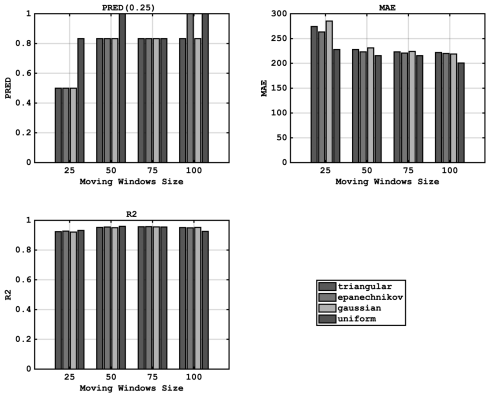

FIGURE 21. Kernel functions (size as independent) for ucp_28 dataset.

using MW and SC are used. In part (c) only one model has been selected for epanechnikov and triangular kernel functions.

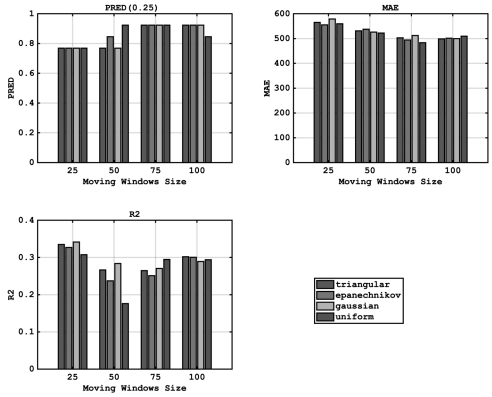

FIGURE 22. Kernel functions (size as independent) for ucp_71 dataset.

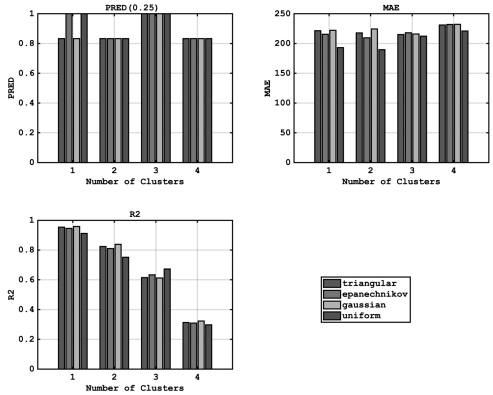

FIGURE 23. Spectral Clustering (size as independent and for clustering) for ucp_28 dataset.

B. RQ2: CAN MOVING WINDOWS BE COMPARED TO SPECTRAL CLUSTERING IN THE ABILITY TO ESTIMATE THE MINIZINE ERROR?

To address the effect of MW and SC the most accurate models were selected from each dataset for windows-based

TABLE 11. Selected models for fpa_kithchenham.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
fpa_kitchenham	triangular	25 %	size	897.24	0.52	0.54	0.94
fpa_kitchenham	triangular	1 c	size	813.22	0.55	0.64	0.06

TABLE 12. Selected models for fpa_maxwell.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
fpa_maxwell	uniform	25 %	size	5,454.20	0.30	0.54	0.94
fpa_maxwell	epanechnikov	4 c	size	4,141.50	0.46	0.64	0.06

TABLE 13. Selected models for ucp_28.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
ucp_28	gaussian	50 %	UAWUUCWTCFECF	947.50	0.66	0.76	0.56
ucp_28	gaussian	1 c	UAWUUCWTCFECF	524.03	0.66	0.79	0.46
ucp_28	uniform	100 %	size	200.61	1	0.92	0.56
ucp_28	uniform	1 c	size	193.08	1	0.91	0.42

TABLE 14. Selected models for ucp_71.

Dataset	Kernel Function	Windows size/Number of clusters	Independent	MAE	Pred(0.25)	R2	p
ucp_71	uniform	100 %	UAW,UUCW, TCF, ECF	570.84	0.92	0.03	0.62
ucp_71	epanechnikov	2 c	UAW,UUCW, TCF, ECF	545.58	1.00	0.14	0.39
ucp_71	uniform	75 %	size	483.4	0.92	0.29	0.26
ucp_71	uniform	3 c	size	459.92	1.00	0.02	0.74
ucp_71	uniform	100 %	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	463.00	0.92	0.063	0.48
ucp_71	uniform	1	SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC	479.58	0.92308	0.054877	0.61

and clustering-based solutions. In Table 8 the best performing MW and SC models are compared in fpa_china dataset.

Using EI,EO,EQ,ILF,EIF independent are used, the value μ_{MW} of MW size 50% is not significantly lower than for SC for 1 cluster ($p = 0.99, \alpha = 0.05$), while SC (1 cluster) is significantly better ($p = 0.01, \alpha = 0.05$).

If size is used as independent MW is not significantly better than clustering for 2 clusters ($p = 0.99, \alpha = 0.05$) and clustering error mean values is significantly lower ($p = 0.00, \alpha = 0.05$). For fpa_china dataset H_0 cannot be rejected, but H_1 can, because SC brings statistically significant improvements in lower a mean error value. H_2 can be confirmed.

Another dataset, where both types of predictors are available, is fpa_isbsg. Selected models that perform best are summarised in Table 9.

Experiments on fpa_isbsg dataset for EI,EO,EQ,ILF,EIF independent variables shows that, the value μ_{MW} of MW size 25% is not significantly lower than for SC for 3 clusters ($p = 0.99, \alpha = 0.05$), while SC (3 cluster) is significantly better ($p = 0.02, \alpha = 0.05$).

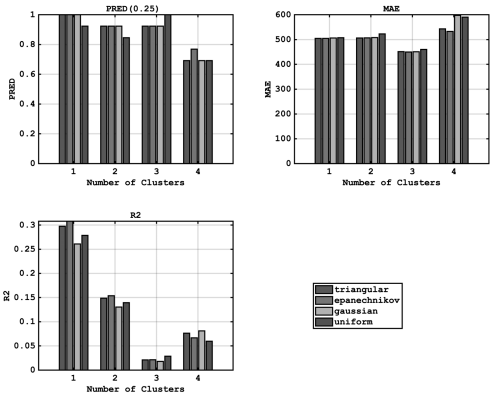

FIGURE 24. Spectral Clustering (size as independent and for clustering) for ucp_71 dataset.

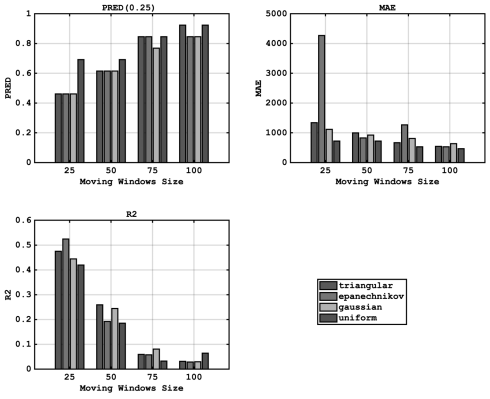

FIGURE 25. Kernel functions (SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC as independent) for ucp_71 dataset.

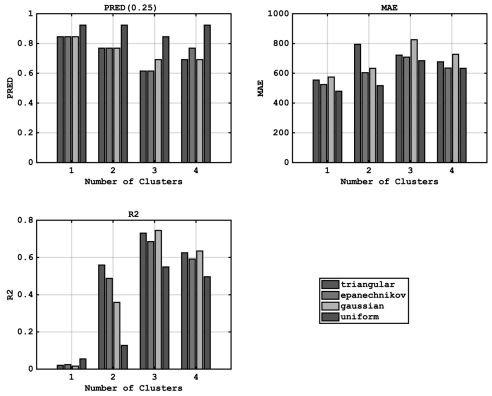

FIGURE 26. Spectral Clustering (SimpleActors, AverageActors, ComplexActors, SimpleUC, AverageUC, ComplexUC as independent and for clustering) for ucp_71 dataset.

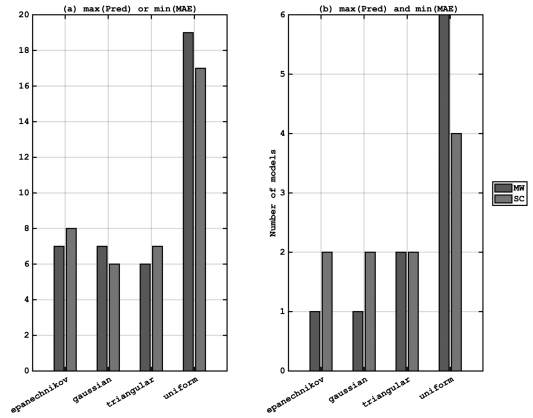

FIGURE 27. Kernel function frequency in best performing models.

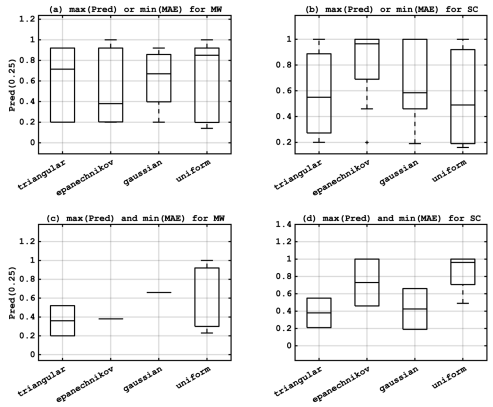

FIGURE 28. Pred(0.25) comparison for kernel functions.

If size is used as independent MW are not significantly better than clustering for 4 clusters ($p = 1.00, \alpha = 0.05$). Otherwise, SC for 4 clusters brings a significantly lower error mean ($p = 0.00, \alpha = 0.05$). For fpa_china dataset H_0 cannot be rejected, but H_1 can, because SC brings statistically significant improvements in a lower mean error value. In case of fpa_isbsg H_0 and H_1 can be rejected, but H_2 cannot be rejected. It cannot be confirmed that FP parameters bring a significantly lower estimation error mean than size option. Next dataset – fpa_EBSPM is only tested using size as an independent variable (as for fpa_kitchenham or fpa_maxwell).

The results (Table 10) for fpa_EBSPM shows the value μ_{MW} of MW size 25% is not significantly lower than for SC for 1 cluster ($p = 1.00, \alpha = 0.05$), while SC (1 cluster) is significant ($p = 0.00, \alpha = 0.05$). H_0 can be rejected, the error means are not equal. H_1 can be rejected, because SC brings significant improvements in lowering a mean error value, resulting in accepting of H_2.

MW and SC do not outperform each other (Table 11) for fpa_kitchenham dataset, H_0 cannot be rejected, but H_1 and H_2 can be rejected. MW (size 25%) versus SC (1 group) are not significantly different ($p = 0.94, \alpha = 0.05$). The same as the opposite comparison ($p = 0.06, \alpha = 0.05$).

The last FP dataset (Table 12), fpa_maxwell, shows again that H_0 cannot be rejected, but H_1 and H_2 can be rejected. MW (size 25%) versus SC (4 clusters) are not significantly different ($p = 0.064, \alpha = 0.05$). Same as the opposite comparison ($p = 0.37, \alpha = 0.05$).

Use case points-based dataset – ucp_28 and ucp_71 are evaluated using the two (ucp_28) or three (ucp_71) predictors set. The selected results for the ucp_28 dataset are summarised in Table 13. As can be seen, no model can be statically confirmed. MW and SC do not outperform each other. H_0 cannot be rejected, but H_1 and H_2 can be rejected. MW (size 50%) versus SC (1 cluster) are not significantly different ($p = 0.56, \alpha = 0.05$) for UCP variables, as for the opposite comparison ($p = 0.46, \alpha = 0.05$). Identically for the size-independent variable - MW (size 100%) versus SC (1 cluster) are not significantly different in both directions ($p = 0.56, \alpha = 0.05$ respectively $p = 0.42, \alpha = 0.05$).

MW and SC do not outperform each other in ucp_71 dataset. As can be seen in Table 14, none of the best performing models (using any of the predictors set) is significantly better than others. H_0 cannot be rejected for models, but H_1 and H_2 can be rejected. Using UCP variables as independent lead to MW (size 100%) vs SC (2 cluster) where $p = 0.56, \alpha = 0.05$, for SC (2 clusters) vs MW (size 100%), there were $p = 0.46, \alpha = 0.05$.

Similarly, when the independent variable of size – MW (size 75%) versus SC (1 cluster) resulting in $p = 0.26, \alpha = 0.05$ respectively $p = 0.74, \alpha = 0.05$. (H_1 and H_2) Finally for models based on the number of actors and use cases again H_1 and H_2 can be rejected and H_0 cannot be rejected. In this case, the best-performing models were MW (size 100%) and SC (1 cluster). MW vs SC brings $p = 0.48, \alpha = 0.05$ and SC vs MW results in $p = 0.61, \alpha = 0.05$. To conclude RQ1, as can be seen, the uniform kernel function can be understood as beneficial. In practice, it rejects the weighting approach as useful for increasing accuracy. In total, 6 of 10 selected models in the moving windows approach are using uniform kernel function.

To summarise RQ2, the move windows approach brings benefits in 6 of 10 MW models, 4 of them received the best result for using windows size of 100%. In practice, all historical instances are used for model training. Statistical significance shows that FP datasets can benefit from using clustering. Moving windows do not outperform SC in either of the cases. All models tested on the UCP dataset achieve similar performance and differences in estimation errors are insignificant.

In future research, the effect of project order will be addressed, and the effect of categorical variables and other data locally approaches will be investigated for software effort or size estimation.

REFERENCES

[1] L. L. Minku and X. Yao, "Using unreliable data for creating more reliable online learners," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Birmingham, U.K., Jun. 2012, pp. 1–8.

[2] L. L. Minku and X. Yao, "Ensembles and locality: Insight on improving software effort estimation," *Inf. Softw. Technol.*, vol. 55, no. 8, pp. 1512–1528, Aug. 2013.

[3] S. Amasaki and C. Lokan, "The effects of moving windows to software estimation: Comparative study on linear regression and estimation by analogy," in *Proc. Joint Conf. 22nd Int. Workshop Softw. Meas. 7th Int. Conf. Softw. Process Product Meas.*, Oct. 2012, pp. 23–32.

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

[5] C. Lokan and E. Mendes, "Investigating the use of duration-based moving windows to improve software effort prediction: A replicated study," *Inf. Softw. Technol.*, vol. 56, no. 9, pp. 1063–1075, Sep. 2014.

[6] 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.

[7] C. Lokan and E. Mendes, "Investigating the use of chronological split for software effort estimation," *IET Softw.*, vol. 3, no. 5, p. 422, 2009.

[8] B. Kitchenham, S. L. Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," *J. Syst. Softw.*, vol. 64, no. 1, pp. 57–77, Oct. 2002.

[9] C. Lokan and E. Mendes, "Investigating the use of moving windows to improve software effort prediction: A replicated study," *Empirical Softw. Eng.*, vol. 22, no. 2, pp. 716–767, Apr. 2017.

[10] Y. Alqasrawi, M. Azzeh, and Y. Elsheikh, "Locally weighted regression with different kernel smoothers for software effort estimation," *Sci. Comput. Program.*, vol. 214, Feb. 2022, Art. no. 102744.

[11] 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.

[12] R. Silhavy, P. Silhavy, and Z. Prokopova, "Algorithmic optimisation method for improving use case points estimation," *PLoS ONE*, vol. 10, no. 11, Nov. 2015, Art. no. e0141887.

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

[14] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "Neural network models for software development effort estimation: A comparative study," *Neural Comput. Appl.*, vol. 27, no. 8, pp. 2369–2381, Nov. 2016.

[15] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "Improved effort and cost estimation model using artificial neural networks and Taguchi method with different activation functions," *Entropy*, vol. 23, no. 7, p. 854, Jul. 2021.

[16] 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.

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

[18] M. Garre, J. J. Cuadrado, M. A. Sicilia, M. Charro, and D. Rodriguez, "Segmented parametric software estimation models: Using the EM algorithm with the ISBSG 8 database," in *Proc. 27th Int. Conf. Inf. Technol. Interface*, Madrid, Spain, Jun. 2005, pp. 193–199.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., Ser. B Methodolog.*, vol. 39, no. 1, pp. 1–22, Sep. 1977.

[20] 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.

[21] V. K. Bardsiri, D. N. A. Jawawi, A. K. Bardsiri, and E. Khatibi, "LMES: A localized multi-estimator model to estimate software development effort," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2624–2640, Nov. 2013.

[22] 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," *IET Softw.*, vol. 6, no. 6, p. 461, 2012.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948.

[24] Z. Prokopova, R. Silhavy, and P. Silhavy, "The effects of clustering to software size estimation for the use case points methods," in *Proc. Comput. Sci. On-Line Conf.*, vol. 575, 2017, pp. 479–490.

[25] M. Azzeh and A. B. Nassif, "Analogy-based effort estimation: A new method to discover set of analogies from dataset characteristics," *IET Softw.*, vol. 9, no. 2, pp. 39–50, Apr. 2015.

[26] V. K. 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," *Empirical Softw. Eng.*, vol. 19, no. 4, pp. 857–884, Aug. 2014.

[27] B. Kitchenham, S. L. Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," *J. Syst. Softw.*, vol. 74, no. 2, p. 227, 2005.
[28] S. Amasaki and C. Lokan, "The evaluation of weighted moving windows for software effort estimation," in *Proc. Int. Conf. Product Focused Softw. Process Improvement*, vol. 7983, 2013, pp. 214–228.
[29] S. Amasaki and C. Lokan, "The effect of moving windows on software effort estimation: Comparative study with CART," in *Proc. 6th Int. Workshop Empirical Softw. Eng. Pract.*, Nov. 2014, pp. 1–6.
[30] S. Amasaki and C. Lokan, "The effects of gradual weighting on duration-based moving windows for software effort estimation," in *Product-Focused Software Process Improvement*, A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, and M. Raatikainen, Eds. Cham, Switzerland: Springer, 2014, pp. 63–77.
[31] P. Silhavy, R. Silhavy, and Z. Prokopova, "Categorical variable segmentation model for software development effort estimation," *IEEE Access*, vol. 7, pp. 9618–9626, 2019.
[32] P. Silhavy, R. Silhavy, and Z. Prokopova, "Evaluation of data clustering for stepwise linear regression on use case points estimation," in *Proc. Comput. Sci. On-Line Conf.*, vol. 575, 2017, pp. 491–496.
[33] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
[34] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès, "Robust subspace clustering," *Ann. Statist.*, vol. 42, no. 2, pp. 669–699, Apr. 2014.
[35] *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia*, document GAO/IMTEC-92-26, 1992.
[36] J. S. Shirabad and T. Menzies. (2005). *The Promise Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada*. [Online]. Available: http://promise.site.uottawa.ca/SERepository
[37] H. Huijgens, A. van Deursen, and R. van Solingen, "The effects of perceived value and stakeholder satisfaction on software project impact," *Inf. Softw. Technol.*, vol. 89, pp. 19–36, Sep. 2017.
[38] ISBSG. (Feb. 2, 2015). *ISBSG Development & Enhancement Repository Release 13*. [Online]. Available: http://isbsg.org
[39] M. Ochodek, B. Alchimowicz, J. Jurkiewicz, and J. Nawrocki, "Improving the reliability of transaction identification in use cases," *Inf. Softw. Technol.*, vol. 53, no. 8, pp. 885–897, Aug. 2011.
[40] A. Subriadi and P. Ningrum, "Critical review of the effort rate value in use case point method for estimating software development effort," *J. Theroretical Appl. Inf. Technol.*, vol. 59, no. 3, pp. 735–744, 2014.
[41] A. Idri, I. Abnane, and A. Abran, "Evaluating Pred(p) and standardized accuracy criteria in software development effort estimation," *J. Softw., Evol. Process*, vol. 30, no. 4, Apr. 2018, Art. no. e1925.
[42] E. Stensrud, T. Foss, B. Kitchenham, and I. Myrtveit, "An empirical validation of the relationship between the magnitude of relative error and project size," in *Proc. 8th IEEE Symp. Softw. Metrics*, Jun. 2002, pp. 3–12.
[43] E. Mendes, C. Lokan, R. Harrison, and C. Triggs, "A replicated comparison of cross-company and within-company effort estimation models using the ISBSG database," in *Proc. 11th IEEE Int. Softw. Metrics Symp. (METRICS)*, Auckland, New Zealand, Jun. 9201, pp. 331–340.
[44] P. Silhavy, R. Silhavy, and Z. Prokopova, "Spectral clustering effect in software development effort estimation," *Symmetry*, vol. 13, no. 11, p. 2119, Nov. 2021.
[45] C. Wohlin, *Experimentation in Software Engineering*. New York: Springer, 2012.

PETR SILHAVY received the Ph.D. degree in engineering informatics from the Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic, in 2009. He has expertise as a CTO and a Software Developer in database programming, database design, data management, and data science. He is currently an Associate Professor with the Faculty of Applied Informatics, Tomas Bata University in Zlín. He is also a Senior Research and an Associate Professor of system engineering and informatics with a demonstrated history of working in research and higher education. His research interests include prediction and empirical methods for software engineering.

RADEK SILHAVY received the Ph.D. degree in engineering informatics from the Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic, in 2009. He is currently an Associate Professor and a Senior Researcher with the Faculty of Applied Informatics, Tomas Bata University in Zlín. He is also an Associate Professor of system engineering and informatics with a demonstrated history of working in research, higher education, project management, and software analysis. His research interests include predictive analytics for software engineering, empirical methods in software engineering, or prediction models focused on cost, size, and effort estimations in system/software engineering. He is also involved in academic publishing as the Editor-in-Chief, an editor, or a reviewer.