IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 May 2022, accepted 12 June 2022, date of publication 22 June 2022, date of current version 11 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185393

Toward Improving the Efficiency of Software Development Effort Estimation via Clustering Analysis

VO VAN HAI, HO LE THI KIM NHUNG, ZDENKA PROKOPOVA, RADEK SILHAVY, AND PETR SILHAVY

Faculty of Applied Informatics, Tomas Bata University in Zlin, 75501 Zlin, Czech Republic

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

This work was supported by the Faculty of Applied Informatics, Tomas Bata University in Zlin, under Project IGA/CebiaTech/2022/001 and Project RVO/FAI/2021/002.

ABSTRACT *Introduction*: The precise estimation of software effort is a significant difficulty that project managers encounter during software development. Inaccurate forecasting leads to either overestimating or underestimating software effort, which can be detrimental for stakeholders. The International Function Point Users Group's Function Point Analysis (FPA) method is one of the most critical methods for software effort estimation. However, the practice of using the FPA method in the same fashion across all software areas needs to be reexamined. *Aim*: We propose a model for evaluating the influence of data clustering on software development effort estimation and then finding the best clustering method. We call this model the effort estimation using machine learning applied to the clusters (EEAC) model. *Method*: We cluster the dataset according to the clustering method and then apply the FPA and EEAC methods to these clusters for effort estimation. The clustering methods we use in this study include five categorical variable criteria (Development Platform, Industrial Sector, Language Type, Organization Type, and Relative Size) and the k-means clustering algorithm. *Results*: The experimental results show that the estimation accuracy obtaining with clustering consistently outperforms the accuracy without clustering for both the FPA and EEAC methods. Significantly, using the FPA method, the average improvement rate from using clustering as opposed to non-clustered was highest at 58.06%, according to the RMSE. With the EEAC method, this number reached 65.53%. The Industry Sector categorical variable achieves the best accuracy estimation compared to the other clustering criteria and k-means clustering. The improvement in accuracy in terms of the RMSE when applying this criterion is 63.68% for the FPA method and 72.02% for the EEAC method. *Conclusion*: Better results are obtained through dataset clustering compared to no clustering for both the FPA and EEAC methods. The Industry Sector is the most suitable clustering method among the tested clustering methods.

INDEX TERMS Software effort estimation, function point analysis, dataset clustering, K-means, categorical variables, machine learning.

I. INTRODUCTION

Estimating the resources a project will need has always been an essential step in project management, including software development management [1]. The ability to estimate the resources required for a software project (including effort, and cost) directly impacts the decision to launch, continue, or cancel any project.

There are three aspects of interest in any effort estimation process: underestimation, estimation accuracy, and overestimation. Overestimation will lead to wasted resources and perhaps customer rejection or a failed bid. In contrast, underestimation will lead to budget overruns, staffing shortages, and delivery delays. Both the above problems

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

can result in significant financial and contract losses [2]. However, the prediction of acceptable outcomes has never been manageable.

Many researchers have focused on designing models to improve effort estimation accuracy. Many models and techniques have been proposed, which can be broadly characterized into three categories [3], [4]: 1) non-algorithm approaches, 2) algorithm approaches, and 3) machine learning (ML) techniques. For the non-algorithmic approaches, experts typically use historical sample projects in project resource estimation. These approaches include the Analogy Technique [5], Expert Judgment [6], Pricing to Win, Wideband Delphi [7], Work Breakdown Structure (WBS) [8], [9], and Planning Poker [10], [11]. The algorithmic approaches are based on mathematical formulas. Representative approaches include Software Life Cycle Management (SLIM) [12], [13], Source Lines of Code (SLOC) [7], the Constructive Cost Model (COCOMO) [14], Use Case Points (UCP) [15], and Function Point Analysis (FPA) [16], [17]. Other methods of this type that are based on FPA include COSMIC [18], FiSMA [19], and MarkII [20], and NESMA [21]. ML techniques have been most frequently used in software effort estimation in recent years [22] and include Artificial Neural Networks, Fuzzy Logic, Neuro-Fuzzy Systems, Bayesian Networks, Regression Tree, Support Vector Machines, and the Genetic Algorithm.

However, the categorization of effort estimation into different techniques is only relative. These methods are often combined with each other in the sense that each method performs a step in the effort estimation process. For example, in our previous study [25], we used a combination of ML and FPA in effort estimation: the ML algorithm was used to estimate the UFP value, and then the FPA method was used for effort computation.

FPA has played a significant role in the software industry. However, it has many limitations, as mentioned in our previous study [23]. In addition, applying the same formula to all software domains may not be the best choice. This study will investigate the use of FPA across different software domains in terms of accuracy. In addition, we will also investigate whether an ML algorithm yields better results than the FPA method within specific software domains.

A. PROBLEM FORMULATION

Many previous studies have attempted to improve the accuracy of the International Function Point Users Group (IFUFG)'s FPA method [18]–[21], [25]–[57]. Some of these efforts have focused on the essential issues surrounding FPA methods [18]–[21]. For example, the MarkII method was proposed to reflect the internal complexities of application systems, whereas the FiSMA method was designed as a service-oriented method to improve upon the process-oriented FPA method. The NESMA method uses the same rules as the IFPUG FPA method, but, depending on the degree of detail possible, it suggests one of three possible function point counts, namely, detailed, estimative, or indicative. Finally, the COSMIC method extracts the best features of the other listed methods.

In addition, ML algorithms have also been applied to this effort [25]–[57]. These approaches are designed to circumvent potential problems arising from applying the same computations to all fields and specialties, as in the FPA method. One such approach involves clustering by various criteria and has yielded specific improvements. Moreover, the FPA method was built on the local IBM datasets [24]. Thus, it cannot accurately reflect effort across the global software industry.

We recently proposed a new effort estimation method [25], along with a novel complexity weighting system for the IFPUG FPA model. This system includes two models, namely, the calibration of functional complexity weight (CFCW) and the calibration of functional complexity weight with optimization (CFCWO) models. The CFCW model is based on the Bayesian Ridge Regressor (BRR) model and calibrates the complexity weights for the FPA, while the CFCWO model uses a voting regressor model to optimize effort estimations obtained from the CFCW model.

Recently, ensemble models have been shown to be much better models than individual models [26]. In ensemble models, individual models are combined using specific combination rules to create a new technique. Many ensemble models have been proposed, such as voting [27] and stacking [28]. Rai *et al.* [29] proposed a hybrid model (HM) in which two models are combined into an ensemble model. This ensemble model averages the approaches of the participating models, creating an effective model. The participating models play equal roles in predicting the outcomes. The authors used the Multiple Layer Perceptron (MLP) model and the Generalized Linear Model (GLM) to evaluate this approach in their paper. The MLP includes three layers, where the input layer consists of seven neurons, the hidden layer consists of 64 hidden units, and the output layer contains one hidden unit. The activation function used in this algorithm is the sigmoid function. The GLM algorithm uses a random component variable from the dependent variables, the formula $\eta : \eta = X\beta$ is used as a systematic component, and the formula $E(y) = \mu = g^{-1}(\eta)$ is used as a link function. The last formula correlates the expected value of the response y to the linear component η. This study uses this HM model in comparisons with the proposed EEAC model on the clustered and non-clustered datasets. We choose the MLP and Linear Regression models as base estimators for the ensemble model. The initial parameter set in the authors' study did not yield good results for our study. Therefore, we tuned the parameters, as shown in TABLE 1.

Many previous studies have aimed at improving the accuracy of software estimation featuring data clustering, such as in [30]–[32] and [45]–[47]. However, the clustering methods selected in these studies were not comprehensive (few clustering methods per study). We surveyed the clustering methods used in recent studies and selected those that were most frequently used. We then clustered our dataset based

TABLE 1. The HM model's parameters.

Algorithm	Implementation	Parameters
Linear Regression	sklearn.linear_model.LinearRegression	*default*
MLP	klearn.neural_network.MLPRegressor	tol=0.00001, momentum=0.000001, random_state=1

on these chosen methods and estimated the software effort needed for the clusters using the FPA method. In addition, we also applied ML to estimate the effort per cluster as a counterweight to the FPA method. We named this ML model the Effort Estimation using machine learning Applied to the Clusters (EEAC) model for ease of comparison. We also wanted to identify the best clustering method among the selected methods. The determination of the best clustering method let us focus on investigating other comprehensive methods for effort estimation.

The dataset used in this study is the International Software Benchmarking Standards Group (ISBSG) dataset [33], which contains contributions from many companies worldwide, thus circumventing locality issues. Moreover, diversity was considered in the selection of the clustering methods for this study.

Based on the details provided above, we formulated the following research questions:

RQ1: How does data clustering affect the FPA method's estimation accuracy? Are there significant improvements in accuracy?

RQ2: Does the EEAC model outperform the FPA and HM models on the dataset with and without clustering?

RQ3: Which clustering method leads to the best accuracy among the studied methods?

We conducted an experimental investigation to answer these research issues. In addition, we performed pairwise t-testing to compare method accuracies [34], [35]. These comparisons include: 1) the FPA method's estimation accuracy on datasets with and without clustering, and 2) the EEAC method's estimation accuracy compared to the FPA and HM methods on datasets with and without clustering. The hypotheses for these comparisons are expressed as follows:

- H1: There is a significant difference in the estimation capabilities of the FPA method under clustering and no clustering. This statement implies that the FPA method estimation accuracy with clustering differs significantly from that without clustering.
- H2: There is a significant difference in the estimation capabilities of the EEAC method and the FPA, or HM methods with and without clustering. This statement implies that the EEAC method's estimation accuracy differs significantly from those of the other two methods with and without clustering.

B. CONTRIBUTIONS

The essential contributions of this study are as follows:

1) This study shows that clustering according to a specific method will lead to better accuracy than simply applying FPA across all fields and specialties. The effort estimation results for the clusters using the FPA method is better than the effort estimation of the FPA method without clustering. The clustering is based on the categorical variables Development Platform (DP), Industrial Sector (IS), Language Type (LT), Organization Type (OT), and Relative Size (RS), plus the k-means clustering algorithm.
2) The best-performing clustering method, i.e., IS, is identified. This criterion was obtained via the following evaluation criteria: MAE, MAPE, RMSE, MBRE, and MIBRE.
3) The EEAC method with IS clusters achieves higher accuracy than using the FPA model with such clusters.

The remaining sections of this paper are organized as follows: Section 2 presents the related work. The background for our work, including an overview of the FPA method, and k-means clustering, is present in Section 3. Section 4 describes the research methodology, including the experimental setup, data preprocessing, and evaluation criteria. The results and a discussion are presented in Section 5. In Section 6, we examine validity. The conclusion and future work are presented in Section 7.

II. RELATED WORK

We conducted searches on the IEEE Explore, Google Scholar, and Web-of-Science websites with keywords related to "ISBSG", "clustering", and "effort estimation" for the period 2016-2020. This survey investigated the frequency of grouping with some variables from the ISBSG dataset. Using the results, we decided upon the clustering methods for our research. Some of the results from the survey are found below.

In their systematic literature review, González *et al.* [36] reported on the frequencies with which fields in the ISBSG dataset were used from 2000 to 2013. According to the research results, the four most frequently used categorical variables for clustering were Development Type, Organization Type, Application Type, and Business Area Type. However, in [13], the author proposed replacing the two features Application Type and Organization Type with the derived features Application Group and Industry Sector, respectively.

Huang *et al.* [45] used seven categorial features, i.e., Development Platform, Development Type, Organization Type, Business Area Type, Application Type, and Primary Programming Language, to demonstrate how powerful preprocessing data techniques and their combinations can be.

Meridji *et al.* [46] selected three clustering attributes, namely, Organization Type, Application Type, and Development Type, to test hypotheses about the relationship between team size and work effort and between productivity and project function size. They showed that the correlations between these two pairs of factors are proportional.

Prokopová *et al.* [47] selected four independent variables, i.e., Count Approach, Business Area Type, Industry Sector, and Relative Size, to determine their influence on produc-

tivity and Normalized Work Effort. The experimental result showed that productivity is weakly dependent on the tested factors.

According to Kaewbanjong and Intakosum [48], the best 15 independent variables for clustering data for predicting user software satisfaction were: Client-Server, Personnel Changes, Total Defects Delivered, Inactive Project Time, Industry Sector, Application Type, Development Type, How Methodology Acquired, Development Techniques, Decision-Making Process, Intended Market, Size Estimate Approach, Size Estimate Method, Cost Recording Method, and Effort Estimate Method.

In a systematic review, Pillai *et al.* [49] singled out two factors, i.e., Development Type and Industry Sector, for future research.

López-Martín [50] used multilayer feedforward neural networks on the Development Platform and Language Type variables from the ISBSG dataset to confirm that the MLP model can be used to predict the duration for which developed software will be maintained.

Li *et al.* [51] used Language Type to examine the correlation between programming language and productivity. Their research comprehensively analyzed productivity across languages.

Fernández and González [52] examined usage of Application Group, 1st Data Base System, Development Platform, Development Type, Industry Sector, Language Type, Primary Programming Language, and Used Methodology as categorical criteria. The results showed that Development Type, Language Type, and Development Platform were most commonly used in categorical analyses. However, they also recommended increased usage of the Application Group, Project Elapsed Time, and 1st Data Base System variables.

In their paper, Silhavy *et al.* [31] proposed a new categorical variable segmentation model. The model was based on dataset segmentation via three categorical variables: Relative Size, Industry Sector, and Business Area Type. The proposed model with Relative Size increased the estimation accuracy compared to clustering-based models and the IFPUG FPA.

In their survey, Usharani *et al.* [53] found that the selective classification approach classifying projects based on essential attributes, such as Organization Type, Project Type, and Development Platform, will improve effort prediction.

In [54], the authors proposed a learning-based adjustment model for the FPA = using a Genetic Algorithm and analogy-based estimation. They also used Language Type, Development Platform, Development Type, and 1st Database System as the crucial factors. As a result, their proposed research method can improve both the usability and accuracy of the FPA method.

Pospieszny *et al.* [55] used the attributes Industry Sector, Application Type, Development Type, Development Platform, Language Type, Package Customization, Relative Size, Architecture, Agile, Used Methodology, and Resource Level as input parameters to determine effort and duration. They used an ensemble model with three ML algorithms (Support Vector Machines, Neural Networks, and Generalized Linear Model) to create a decision support tool for organizations developing or implementing software systems.

TABLE 2. Lists the fields used in the studies mentioned above. TABLE 2. Categorical variables identified by the survey.

No.	Clustering Method	References
1	Development Type (DT)	[45], [46], [48], [49], [52], [54], [55], [57]
2	Development Platform (DP)	[45], [50], [52], [53], [54], [55], [57]
3	Language Type (LT)	[50], [51], [52], [54], [55], [56], [57]
4	Industry Sector (IS)	[25], [36], [47], [48], [52], [31], [55]
5	Organization Type (OT)	[45], [46], [53], [57]
6	Relative Size (RS)	[47], [31], [55], [56]
7	Application Type (AT)	[45], [46], [48] [55]
8	Business Area Type (BAT)	[45], [47], [31]
9	Primary Programming Language (PPL)	[49], [52]
10	Application Group (AG)	[36], [52]
11	1st Database System (1DB)	[52], [54]
12	Used Methodology (UM)	[52], [55]
13	Count Approach (CA)	[47], [56]
14	Project Type (PT)	[53]
15	Resources Level (RL)	[55]
16	1st Operation System (1OS)	[56]
17	K-Means Algorithm	[37], [38], [39], [40], [30], [41], [42], [43], [44]

Saavedra *et al.* [56] developed an automated estimation-model generator system that uses ML techniques to analyze the accuracy of these models, comparing them to the traditional estimation methods using an international database and the internal database of a company. The authors used Relative Size, Count Approach, 1st Operation System, and Language Type as data partitioning criteria.

Song *et al.* [57] used Organization Type as the main criteria in partitioning their dataset. They also used Development Type, Language Type, and Development Platform as features in their experiments. As a result, they proposed a novel estimator with greater accuracy.

In the Sinaga K. P. and Yang M. study [44], an unsupervised learning schema for the k-means algorithm (U-k-means) was proposed so that it is free of initializations and parameter selection while simultaneously finding an optimal number of clusters. Experiments show that the U-k-means algorithm has some outstanding aspects compared to other clustering methods.

Based on TABLE 2 and FIGURE 1, the top seven methods were identified as DT, DP, LT, IS, OT, RS, and AT.

Development Type is an essential criterion, as it determines the approach to the project. Moreover, Gonzalez *et al.* [36] found that 57.9% of the articles they reviewed used Development Type as a categorical variable. We only wished to study

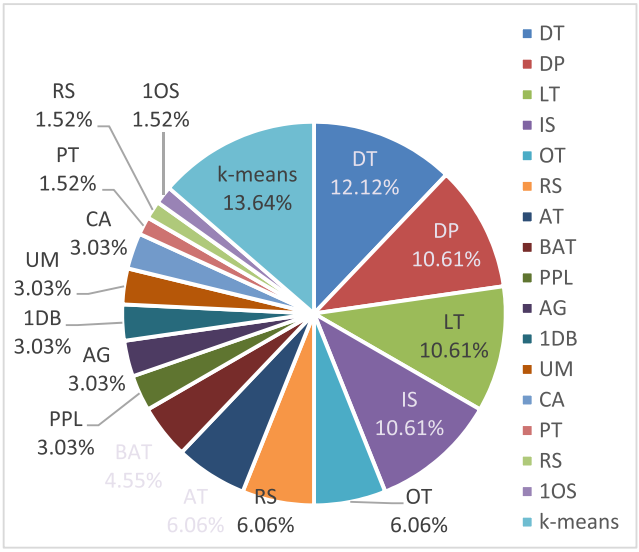

FIGURE 1. Distribution of clustering methods used.

projects that were considered new developments. Therefore, Development Type was included in the data filtering stage as the most basic type of categorical variable.

In addition, many studies have used the k-means algorithm for clustering to improve software effort estimation, see, for example, [30], [37]–[43]. Therefore, this study also used the k-means algorithm for clustering.

AT clustering was not included in this study, as the number of clusters obtained with this criterion was too large. We also considered BAT clustering but, after filtering the data, there were too few fields left with more than 20 records (for just new development types: Don't Know – 224, Banking – 76, Telecommunications – 49, Financial – 28, Engineering – 21, and (blank) – 81). Therefore, we decided to remove BAT from our experiment.

Thus, we retained six variables, namely, DT, DP, LT, IS, OT, and RS, as our clustering criteria. In addition, the k-means algorithm was used for clustering.

III. BACKGROUND

A. FPA METHOD OVERVIEW

In the late 1970s, the FPA method was first introduced by Albrecht [24], who proposed it as a metric for measuring the functionality of a project. IFPUG [17] has been the governing body for FPA since 1986, making it responsible for improving and developing counting rules and other related matters. With the creation of IFPUG, the original FPA method became known as IFPU FPA. In this study, the FPA method will always refer to the IFPUG FPA method. FPA is currently standardized by ISO/IEC 20926:2010 [16]. This standard defines a set of definitions, rules, and steps for the application of this standard [58].

The FPA method uses three transactional functions (External Input (EI), External Output (EO), and External Inquiry (EQ)), and two data functions (Internal Logic Files (ILF), and

TABLE 3. Functional complexity weights.

		Component				
		EI	EO	EQ	EIF	ILF
Functional Complexity Weight	Low	3	4	3	5	7
	Average	4	5	4	7	10
	High	6	7	6	10	15

External Interface Files (EIF)), all of which are called Base Functional Components.

- An EI processes data or controls the information sent outside the boundary.
- An EQ conveys or controls information outside the boundary. The processing logic contains no mathematical formulas or computations and creates no derived data.
- An EO sends data or control information outside the application's boundary and conducts processing beyond that of an EQ. At least one mathematical formula or computation must be included in the processing logic. It must have the ability to create derived data, maintain one or more ILFs, and/or change the system's behavior.
- An ILF is a group of users with recognizable logically-related data or control information maintained within the application's measured boundary.
- An EIF is a detectable group of data or control information connected to user logic, referenced by the application being measured but kept within the bounds of another program.

Each of these Base Functional Components is judged as simple, average, or complex and assigned a weight accordingly. TABLE 3 shows the functional complexity weights of these functions.

To count the Unadjusted Function Points (UFP), we use Eq. (1) below:

$$UFP = \sum EI \times weight + \sum EO \times weight + \sum EQ \times weight + \sum ILF \times weight + \sum EIF \times weight \quad (1)$$

The effects of 14 factors (Fs) on the counting process should be identified in this phase, and the General System Characteristics (GSCs) should also be determined. Further, the Value Adjustment Factor (VAF) should be computed (see Eq. (2)) before counting the Function Points (FPs) for the Adjusted Function Points (AFP) (see eq. (3)). The GSCs are given ratings in the interval 0 to 5 according to their degree of influence.

$$VAF = 0.65 + 0.01 \times \sum_{i=1}^{14} (F_i \times rated) \quad (2)$$

$$AFP = UFP \times VAF \quad (3)$$

The FP counting process can be summarized in the following five steps:

1) Determine the counting scope and boundary.
2) Measure the data and transactional functions.

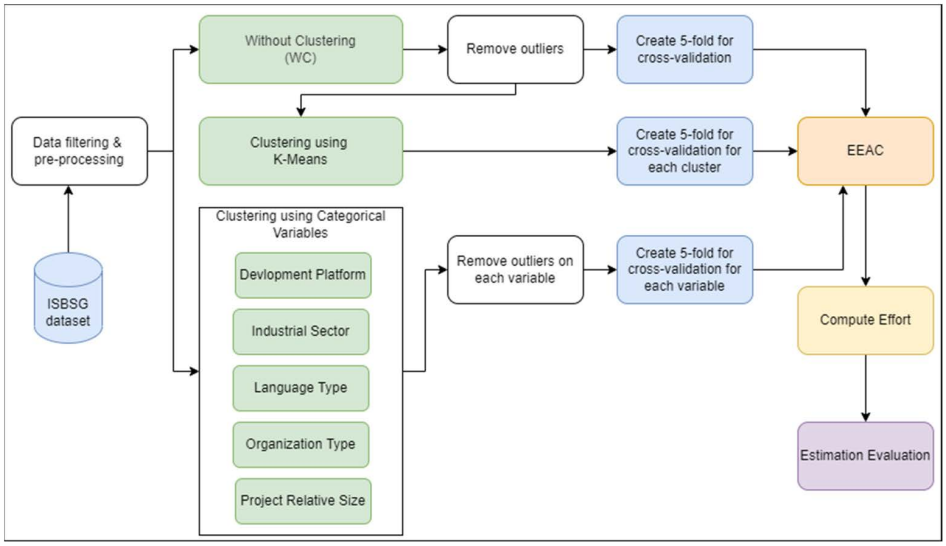

FIGURE 2. Experimental process.

3) Compute the U FP.
4) Determine the GSCs and calculate the VAF.
5) Calculate the AFP.

To estimate the effort after AFP counting, we should have another parameter, namely, the Productivity Factor (PF). This factor explains the relationship between one FP and the number of hours needed for its development by one person. Productivity was studied in [59] and [60]. According to the ISBSG, the Productivity Delivery Rate (PDR) is used to measure efficiency in person-hours per FP. Thus, we can derive the PF from the PDR by inverting it, and vice versa [48]. Eq. (4) below can be used to calculate the effort:

$$Effort = AFP \times PDR \tag{4}$$

B. K-MEANS CLUSTERING

K-means clustering [61] is an unsupervised ML algorithm used to cluster given objects into k clusters, where k is pre-specified. In clustering k-means, each cluster is represented by its center (centroid), i.e., the mean of the points assigned to the cluster [62]. We can summarize the k-means algorithm as follows:

1) Specify the number of clusters k.
2) Randomly select k points from the central data set (centroids) for the k clusters.
3) Calculate the distances between the points and the centers.
4) Assign the points to the centroids nearest them to form the initial clusters.
5) Define new centers for the clusters by calculating the means for the data points in the respective clusters.
6) Repeat step 3 until there no points change clusters.

In this study, the k-means algorithm was implemented with the *sklearn.cluster.KMeans* package, where the number of clusters is obtained with the Elbow method (next section), and other parameters are at their default settings. In this package, the distances between the points and the centers are calculated using the Euclidean Distance algorithm. When a new project requires effort estimation, the Euclid distances will be calculated, and the cluster to which the new project belongs will be determined. For an effort estimate, the selected cluster's corresponding model will be applied.

1) DETERMINATION OF k

The Elbow approach is used to identify the number of clusters (k) for the k-means algorithm and does so based on the visualization graph. In particular, it looks at the attenuation of the distortion function and selects the elbow point, which is the point at which the reduced rate of the distortion function will change the most. That is, after this point, increasing the number of clusters does not significantly reduce the distortion function [63], [64].

The essence of the Elbow method is the sum of squares of errors (SSE). This quantity sums the squared Euclidean distances from all points to their centroids:

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 , \tag{5}$$

where C_i is the i^{th} cluster, p is a sample point in C_i, and m_i is the mean of the points in C_i (centroid of C_i).

In this study, we used the Elbow method implemented by the Yellowbrick organization [65]. The KElbowVisualizer object with the Distortion metric was used to detect the optimal value of k.

IV. RESEARCH METHODOLOGY

A. EXPERIMENTAL SETUP

In this section, we explain the experimental setup, which is outlined in FIGURE 2.

First, the data from the ISBSG dataset was filtered and preprocessed to create the dataset for this experiment (see the

next section). Next, we formed the three experimental groups described below:

1) Experimental group 1: one experiment on the whole dataset without clustering (WC)
   a. Remove outliers from the entire dataset.
   b. Create a 5-fold cross-validation.
   c. Compute effort using the IFPUG FPA method for the whole dataset.
   d. Compute effort using the BRR algorithm for the whole dataset.
   e. Evaluation.
2) Experimental group 2: one experiment on k clusters
   a. Remove outliers from the entire dataset.
   b. Find the optimal k for the dataset using the Elbow method.
   c. Cluster the dataset into k clusters using the k-means algorithm.
   d. Create a 5-fold cross-validation for each cluster.
   e. Compute effort using the IFPUG FPA method for each cluster.
   f. Compute effort using the BRR algorithm for each cluster.
   g. Evaluation of the clusters and the computation of the mean value.
3) Experimental group 3: five experiments corresponding to the five categorical variables
   a. Cluster the dataset with the categorical variables.
   b. Remove the outliers from each cluster.
   c. Create a 5-fold cross-validation for each cluster.
   d. Compute effort using the IFPUG FPA method for each cluster.
   e. Compute effort using the BRR algorithm for each cluster.
   f. Evaluation.

Finally, we conduct a comparison of the experimental results.

B. TESTED MODELS

The proposed EEAC model was tested on the whole dataset without clustering and on the dataset clustered according to the categorical variables (DP, IS, LT, OT, and RS) and the k-means algorithm. The models that were compared are described briefly below:

- EEAC - the effort was computed using the EEAC approach. This method infers the complexity weights system from each cluster's five base functional components of the FPA counting process by using Bayesian Ridge Regression (BRR). It then uses this complexity weight system in the procedure of effort estimation with the PF (PDR) parameter from the dataset.
- IFPUG FPA – the effort was computed using the IFPUG FPA approach [16] to find the UFP, VAF, and PF (PDR) from the dataset (mean value from all sectors or values based on each sector).
- HM – the effort was computed using the UFP obtained with the average ensemble approach developed by Rai *et al.* [29], and the VAF, and PF (PDR) came from the dataset (mean value from all sectors or values based on each sector).

C. DATA PREPROCESSING

The dataset we used in our experiment came from the ISBSG repository for August 2020 R1 [33]. The data filtering criteria in our study was as follows:

1) Record data quality was A or B.
2) We only selected records using the IFPUG the counting approaches (including IFPUG Old and IFPUG 4+).
3) Development Type was New Development only.
4) Rows with empty Base Functional Component values were removed.
5) Rows with empty values in Normalized Productivity Delivery Rate (PDR) and Summary Work Effort (SWE) were also erased.
6) Fill in the blank VAF cells with the values obtained via Eq. (3).

According to Lichtenberg [66], the number of records in the dataset should be large enough for a given training set size to attain the most satisfactory results. In addition, Hammad [67] proved that some algorithms learn perfectly once the training set is large enough. In our study, each categorical variable cluster needed to contain more than 20 records. Any cluster unable to satisfy this condition was put into a cluster named ''Others''.

Additionally, we observed several AFP and SWE values that were too far from the mean group, implying that the data was potentially noisy. We found and removed outliers based on the interquartile range (IQR) method [34], which produced a lower bound of 0.15 and an upper bound of 0.85. After clustering using one of the categorical variables or the k-means algorithm, we proceeded to remove the outliers before continuing the experiments.

1) DEVELOPMENT PLATFORM

The DP is determined by the operating system used [68]. Each project is classified as having been developed on a PC, developed on a mid-range (MR) computer, developed on a mainframe (MF), or as multi-platform (Multi). In our case, records without a development platform listed were categorized into a cluster named ''Others.'' FIGURE 3 shows the histogram of the DT variable, and FIGURE 4 provides the boxplots of the dataset clustered according to DP before and after removing the outliers.

2) INDUSTRY SECTOR

The IS indicates the sector in which the software is maintained and supported. FIGURE 5 shows the histogram of the IS variable, and FIGURE 6 provides the boxplots of the dataset clustered according to IS before and after removing the outliers.

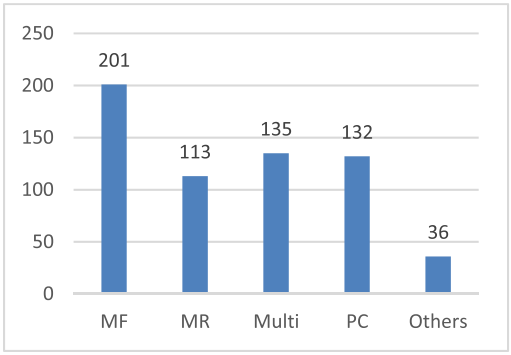

FIGURE 3. Histogram for DT.

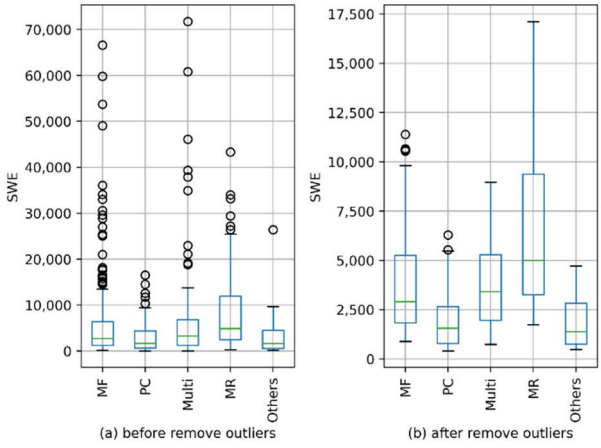

FIGURE 4. Boxplots of the dataset clustered by DP.

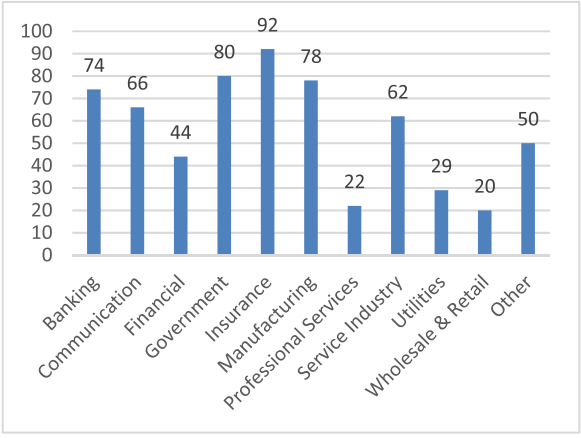

FIGURE 5. Histogram for IS.

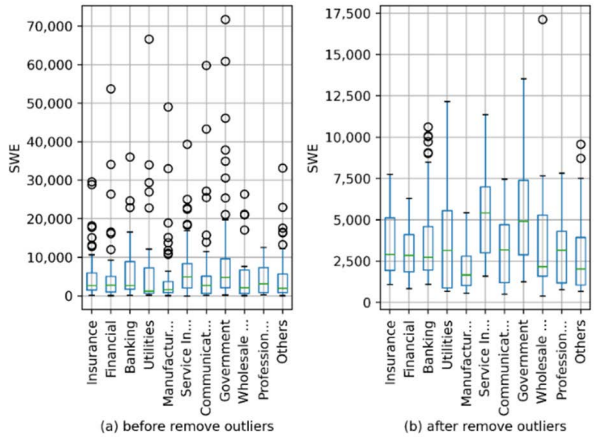

FIGURE 6. Boxplots of the dataset clustered by IS.

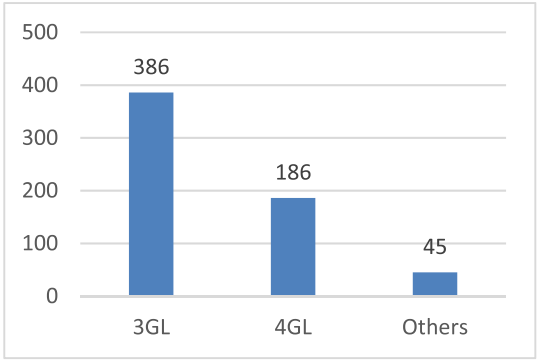

FIGURE 7. Histogram for LT.

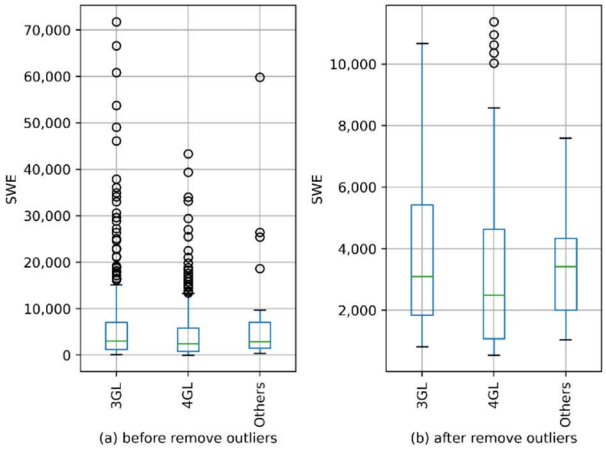

FIGURE 8. Boxplots of the dataset clustered by LT.

3) LANGUAGE TYPE

LT indicates the type of programming language used for the project. In our study, there were three types of language: 3rd generation programming language (3GL), 4GL, and Others (projects with an empty LT field or with less than 20 records). FIGURE 7 shows the histogram of the LT variable, and FIGURE 8 provides the boxplots of the dataset clustered according to LT before and after removing outliers.

4) ORGANIZATION TYPE

OT identifies the type of organization that submitted the project. FIGURE 9 shows the histogram of the OT variable, and FIGURE 10 provides the boxplot of the dataset clustered according to OT before and after removing the outliers.

The following organization types were used in our study: Banking, Communications, Electricity, Gas, Water, Finan-

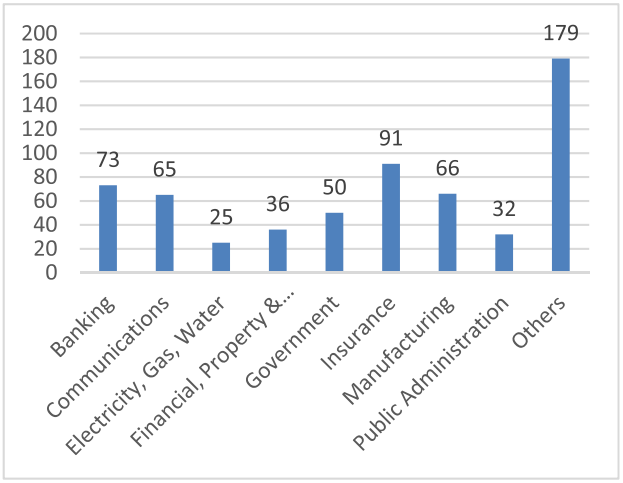

FIGURE 9. Histogram for OT.

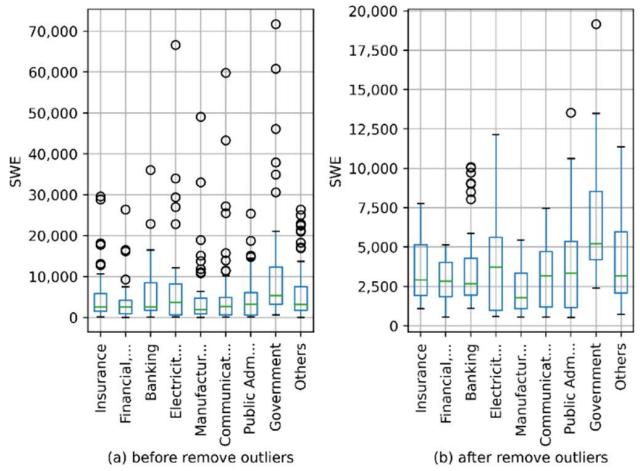

FIGURE 10. Boxplots of the dataset clustered by OT.

TABLE 4. Relative company size.

No.	Abbr.	Relative Size	Functional Size
1	XXS	Extra-extra-small	>= 0 and <10
2	XS	Extra-small	>= 10 and <30
3	S	Small	>= 30 and <100
4	M1	Medium 1	>= 100 and <300
5	M2	Medium 2	>= 300 and <1000
6	L	Large	>= 1000 and < 3000
7	XL	Extra-large	>= 3000 and < 9000
8	XXL	Extra-extra-large	>= 9000 and < 18000
9	XXXL	Extra-extra-extra-large	>= 18,000

cial, Property & Business Services, Government, Insurance, Manufacturing, Public Administration, and Others (blank values for this criteria or less than 20 records).

5) RELATIVE SIZE

In the ISBSG dataset, there are nine relative company sizes, as shown in TABLE 4. There were too few records in the XXS, XS, and XL groups (XXS = 1, XS = 8, XL = 9), and no records in the XXL and XXXL groups. So, we re-defined the S group as companies having functional sizes from 0 to 100, and the L group as companies having functional sizes greater than 1000. According to this re-grouping, the S group contained XXS, XS, and S companies, and the L group included L, XL, XXL, and XXXL companies.

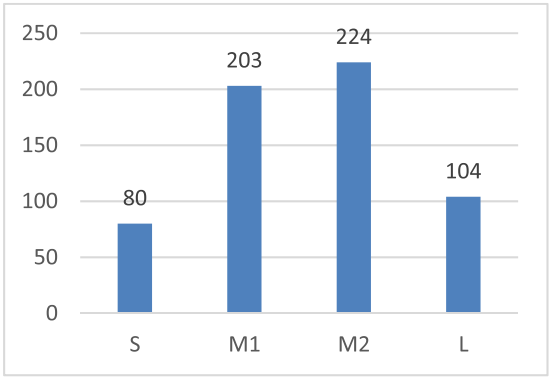

FIGURE 11. Histogram for RS.

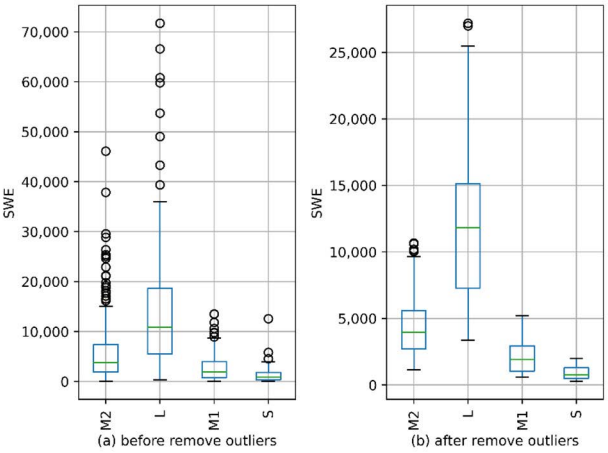

FIGURE 12. Boxplots of the dataset clustered by RS.

FIGURE 11 shows the histogram of the RS variable, and FIGURE 12 presents the boxplot of the dataset clustered according to RS before and after removing the outliers.

6) K-MEANS CLUSTERING

In the experiment using the k-means algorithm for clustering, we proceeded to find the optimal k with the Elbow method [65]. The distortion score reached the elbow value for $k = 6$ (FIGURE 13). FIGURE 14 shows the boxplots plot of the dataset before and after removing outliers. The attributes of the k-means clustering algorithm are EI, EO, EQ, ILF, and EIF.

D. EVALUATION CRITERIA

Choosing the criteria with which to evaluate the accuracy of predictive models is also a matter of concern. According

TABLE 5. The summary evaluation results.

	MAE			MAPE			RMSE			MBRE			MIBRE		
	FPA	EEAC	HM	FPA	EEAC	HM	FPA	EEAC	HM	FPA	EEAC	HM	FPA	EEAC	HM
WC	588.97	534.88	545.26	14.025	13.020	13.332	1,317.42	1,170.19	1,191.32	0.142	0.135	0.137	0.101	0.100	0.101
DP	362.02	274.30	331.78	10.430	8.131	9.603	583.92	410.44	500.19	0.105	0.085	0.099	0.085	0.071	0.082
IS	288.37	204.63	315.49	8.489	6.768	10.733	478.53	327.40	481.72	0.086	0.072	0.125	0.073	0.062	0.097
LT	366.46	300.36	354.79	9.763	8.514	9.504	579.91	442.27	514.21	0.099	0.088	0.101	0.083	0.076	0.086
OT	329.63	245.40	312.24	8.860	7.222	9.741	532.47	365.78	451.47	0.091	0.077	0.109	0.077	0.067	0.089
RS	350.75	290.95	322.54	9.168	7.970	8.683	536.74	415.98	465.54	0.094	0.084	0.090	0.078	0.072	0.077
k-means	372.669	293.028	390.759	9.525	7.840	10	603.531	458.199	600.007	0.096	0.081	0.103	0.081	0.069	0.087

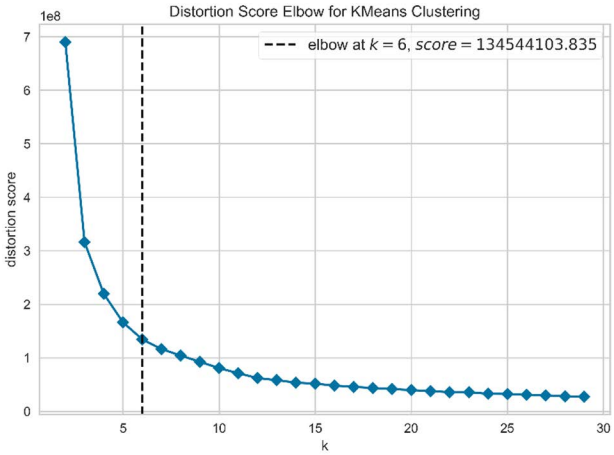

FIGURE 13. Elbow method for optimizing k.

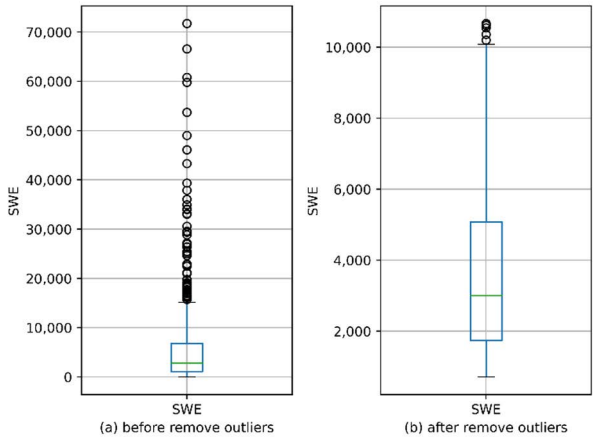

FIGURE 14. Boxplots of the dataset for k = 6 be and after removing outliers.

to [7], [69], and [70], the Mean Magnitude of Relative Error (MMRE) and Mean Magnitude of Relative Error Relative (MMER) criteria are the most common metrics used. However, [71], [72], and [73] have demonstrated that these methods are biased. Azzeh *et al.* [74] recommend using unbiased methods instead. In addition, de Myttenaere *et al.* [75] state that the Mean Absolute Percentage Error (MAPE) is practically and theoretically appropriate for evaluating regression models and in its intuitive interpretation of the relative error. Therefore, this study used the following evaluation criteria: the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Balance Relative Error (MBRE), Mean Inverted Balance Relative Error (MIBRE), and MAPE. These criteria are defined below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{7}$$

$$MBRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|(y_i - \hat{y}_i)|}{min(y_i - \hat{y}_i)} \tag{8}$$

$$MIBRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|(y_i - \hat{y}_i)|}{max(y_i - \hat{y}_i)} \tag{9}$$

$$MAPE = \frac{1}{N}\sum_{i=0}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100, \tag{10}$$

where y_i is the actual value, $\hat{y}_i$ is the estimated value, and n is the number of projects.

V. RESULTS AND DISCUSSION

The experiments were based on data clustering using categorical variables and the k-means algorithm, then performing effort estimation using one of the three methods (FPA, EEAC, or HM). The categorical variables used here were DP, IS, LT, OT, and RS. In addition, baseline experiments on the entire non-clustered dataset were performed to evaluate the clustered performances. All three methods were trained on each cluster and the entire dataset without clustering. The results were then evaluated using the five evaluation criteria MAE, MAPE, RMSE, MBRE, and MIBRE. TABLE 5 provides the experimental results.

In general, TABLE 5 shows the evaluation results for the clusters using MAE, RMSE, MBRE, MIBRE, and MAPE. The first row contains the evaluation results for the entire dataset without clustering. The evaluation results of the five categorical variables are displayed in the next five rows. The last row contains the evaluation results for the k-means clustering. For each evaluation criterion, there are three

columns representing the FPA method, the proposed EEAC method, and the HM method.

There are three main comparisons to be made in this context: 1) comparing the values in the non-clustering row with the remaining rows, 2) comparing the values in the EEAC column with the values in the FPA column for each evaluation criterion, and 3) comparing the values in the EEAC column and the values in the HM column for each evaluation criterion.

First, observe is that the estimation errors with clustering are always better than those for without clustering, confirming once more that applying the original FPA method across all areas is inaccurate. FIGURE 15 to FIGURE 19 show the evaluation results according to the evaluation criteria (Eq. 6-10).

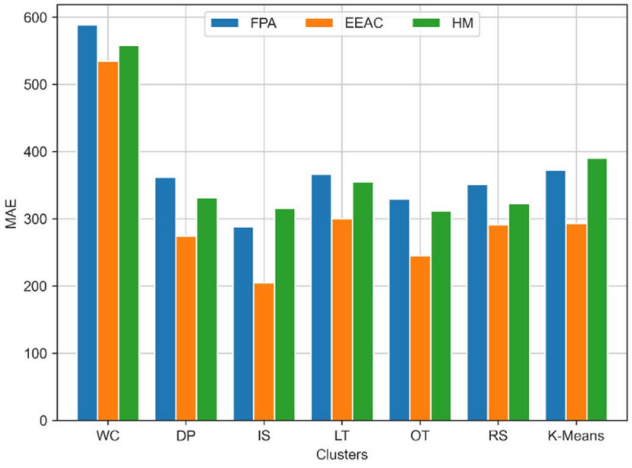

FIGURE 15. The MAE evaluation results for FPA, EEAC, and HM.

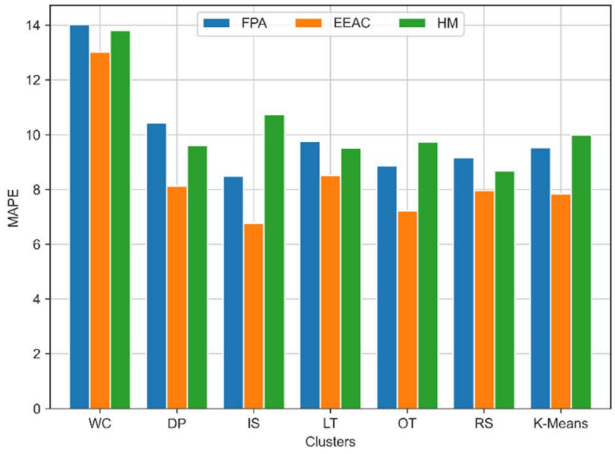

FIGURE 16. The MAPE results for FPA, EEAC, and HM.

For all the evaluation criteria, the error for the proposed method is consistently lower than the errors for the FPA and HM methods for without clustering and each clustering method. Hence, the proposed method is better than the FPA and HM methods according to all criteria.

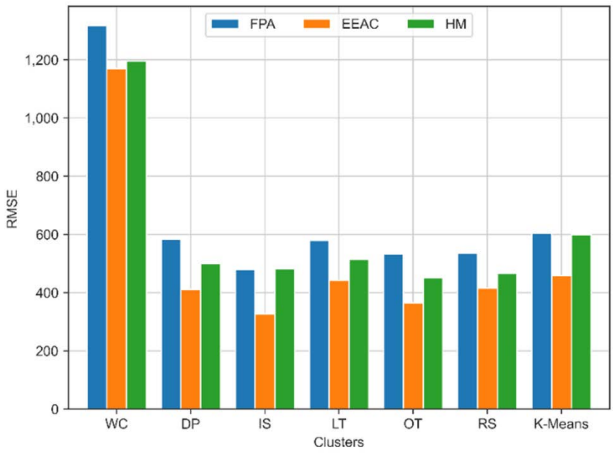

FIGURE 17. The RMSE evaluation results for FPA, EEAC, and HM.

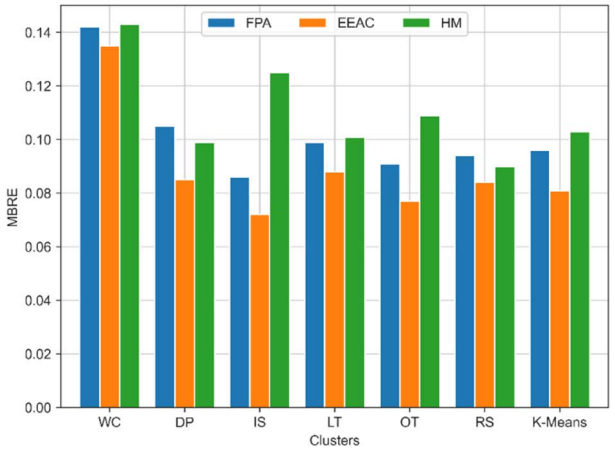

FIGURE 18. The MBRE evaluation results for FPA, EEAC, and HM.

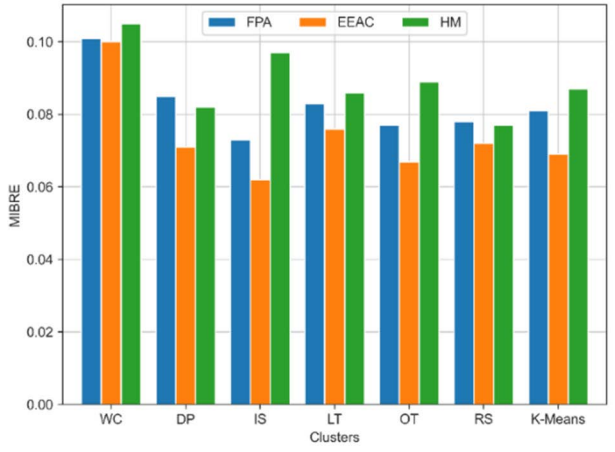

FIGURE 19. The MIBRE evaluation results for FPA, EEAC, and HM.

From TABLE 5, there are three exciting points to be made: 1) the error values always decrease from the FPA column to the EEAC column, 2) the error values for the categorical variable rows and k-means row are always smaller than the

TABLE 6. The statistical T-test based on evaluation results for the FPA method on the whole dataset without clustering with each cluster.

Pairs of methods		DP vs. WC	IS vs. WC	LT vs. WC	OT vs. WC	RS vs. WC	k-means vs. WC
MAE results	Avg. MAE	362.022 vs. 588.971	288.367 vs. 588.971	366.46 vs. 588.971	329.631 vs. 588.971	350.747 vs. 588.971	372.669 vs 588.971
	Avg. p-value	0.00000	0.00000	0.00047	0.00002	0.00002	0.00057
	Statistical conclusion	>>	>>	>>	>>	>>	>>
MAPE results	Avg. MAPE	10.43 vs. 14.025	8.489 vs. 14.025	9.763 vs. 14.025	8.86 vs. 14.025	9.168 vs. 14.025	9.525 vs 14.025
	Avg. p-value	0.00089	0.00035	0.00086	0.00007	0.00149	0.00254
	Statistical conclusion	>>	>>	>>	>>	>>	>>
RMSE results	Avg. RMSE	583.917 vs. 1317.422	478.53 vs. 1317.422	579.911 vs. 1317.422	532.472 vs. 1317.422	536.738 vs. 1317.422	603.531 vs 1317.422
	Avg. p-value	0.00000	0.00000	0.00001	0.00001	0.00000	0.00007
	Statistical conclusion	>>	>>	>>	>>	>>	>>
MBRE results	Avg. MBRE	0.105 vs. 0.142	0.086 vs. 0.142	0.099 vs. 0.142	0.091 vs. 0.142	0.093 vs. 0.142	0.096 vs 0.142
	Avg. p-value	0.00085	0.00039	0.00078	0.00006	0.00164	0.00259
	Statistical conclusion	>>	>>	>>	>>	>>	>>
MIBRE results	Avg. MIBRE	0.00289	0.073 vs. 0.101	0.083 vs. 0.101	0.077 vs. 0.101	0.078 vs. 0.101	0.081 vs 0.101
	Avg. p-value	0.00289	0.00033	0.00158	0.00015	0.00214	0.01135
	Statistical conclusion	>>	>>	>>	>>	>>	>>

error values listed in the row for without clustering; 3) the error values for the categorical variable IS are always smaller than the error values for the other clustering methods in the FPA and EEAC columns.

Based on these statements and the previous results, we can now answer the research questions.

RQ1: How does data clustering affect the FPA method's estimation accuracy? Are there significant improvements in accuracy?

TABLE 5 shows the evaluation results for the FPA method. We can see that in each FPA column for MAE, MAPE, RMSE, MBRE, and MIBRE, the errors for the categorical variables and k-means clustering are all smaller than the error for the row without clustering. Therefore, we can assert that using the FPA method with clustering is more accurate than using it with no clustering.

To determine whether there was a significant improvement in accuracy due to clustering or not, we performed a pairwise t-tests. TABLE 6 lists the average p-values and tMAE, MAPE, RMSE, MBRE, and MIBRE values over five different runs and the final statistical conclusions. The notation $\gg$ reflects the statistical superiority of the clustering over no clustering at the 95% confidence level. Therefore, H1 is accepted i.e., there is a significant improvement in the FPA method when it used in conjunction with clustering.

TABLE 7 shows the percentage improvements in the FPA evaluation results with clustering compared to without clustering. The mean percentage improvements across the evaluation criteria vary from 21.29% for the MIBRE to 58.06%, for the RMSE.

RQ2: Does the EEAC model outperform the FPA and HM models on the dataset with and without clustering?

TABLE 8 shows the results of the pairwise t-tests for the clusters. There are two comparisons to be made for no clustering and with each clustering method: EEAC versus FPA, and EEAC versus HM. The results confirm that the EEAC method with and without clustering is statistically superior to the other methods at the 95% confidence level. Therefore, we accept hypothesis H2, i.e., the EEAC model outperforms the FPA and HM models with and without clustering.

The improvement percentages in the errors for EEAC versus FPA across the clusters can be found in TABLE 9.

RQ3. Which clustering method leads to the best accuracy among the studied methods? shows that the estimation errors for IS for both the FPA and EEAC methods are always the smallest among the clustering methods (bold values in the table). Thus, the estimation accuracies of the FPA and EEAC methods are highest for this clustering method. On the other hand, the results obtained for the k-means algorithm and the OT criterion are also very positive. For this research question, we can assert that using the categorical variable IS for clustering leads to the best accuracy among the studied clustering methods.

VI. VALIDITY

Internal validity is an incorrect/inaccurate evaluation approach to analyzing the proposed method. However, statistical sample validation needed to be considered. The k-fold cross-validation method was used to mitigate the threat to this validity, ensuring that the suggested method was appropriately appraised. Another internal hazard that could have affected the validity of the generated findings was ML parameter selection. We employed the BRR technique's default parameter configuration in this work.

The external validity of the results produced in this study is concerned with generalizability. The proposed method's prediction ability was tested using the ISBSG

TABLE 7. The statistical t-test based on the evaluation results.

Pairs of methods		DP		IS		LT		OT		RS		K-Means	
		EEAC vs. FPA	EEAC vs. HM	EEAC vs. FPA	EEAC vs. HM	EEAC vs. FPA	EEAC vs. HM	EEAC vs. FPA	EEAC vs. HM	EEAC vs. FPA	EEAC vs. HM	EEAC vs. FPA	EEAC vs. HM
MAE results	Avg. MAE	274.303 vs. 362.022	274.303 vs. 331.779	204.631 vs. 288.367	204.631 vs. 315.485	300.357 vs. 366.46	300.357 vs. 354.792	245.401 vs. 329.631	245.401 vs. 312.244	290.95 vs. 350.747	290.95 vs. 322.544	293.028 vs. 372.669	293.028 vs. 390.759
	Avg. p-value	0.00000	0.00000	0.00000	0.00000	0.00006	0.00001	0.00000	0.00000	0.00001	0.00268	0.00000	0.00002
	Conclusion	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
MAPE results	Avg. MAPE	8.131 vs. 10.43	8.131 vs. 9.603	6.768 vs. 8.489	6.768 vs. 10.733	8.514 vs. 9.763	8.514 vs. 9.504	7.222 vs. 8.86	7.222 vs. 9.741	7.97 vs. 9.168	7.97 vs. 8.683	7.84 vs. 9.525	7.84 vs. 10.0
	Avg. p-value	0.00000	0.00002	0.00000	0.00000	0.00001	0.00004	0.00000	0.00013	0.00000	0.00005	0.00000	0.00013
	Conclusion	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
RMSE results	Avg. RMSE	410.441 vs. 583.917	410.441 vs. 500.193	327.403 vs. 478.53	327.403 vs. 481.722	442.266 vs. 579.911	442.266 vs. 514.21	365.776 vs. 532.472	365.776 vs. 451.471	415.983 vs. 536.738	415.983 vs. 465.544	458.199 vs. 603.531	458.199 vs. 600.007
	Avg. p-value	0.00000	0.00000	0.00000	0.00000	0.00002	0.00001	0.00000	0.00001	0.00001	0.00082	0.00000	0.00001
	Conclusion	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
MBRE results	Avg. MBRE	0.085 vs. 0.105	0.085 vs. 0.099	0.072 vs. 0.086	0.072 vs. 0.125	0.088 vs. 0.099	0.088 vs. 0.101	0.077 vs. 0.091	0.077 vs. 0.109	0.083 vs. 0.093	0.083 vs. 0.09	0.081 vs. 0.096	0.081 vs. 0.103
	Avg. p-value	0.00000	0.00033	0.00000	0.00002	0.00005	0.00163	0.00000	0.00237	0.00000	0.00015	0.00000	0.00020
	Conclusion	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>
MIBRE results	Avg. MIBRE	0.071 vs. 0.085	0.071 vs. 0.082	0.062 vs. 0.073	0.062 vs. 0.097	0.076 vs. 0.083	0.076 vs. 0.086	0.067 vs. 0.077	0.067 vs. 0.089	0.072 vs. 0.078	0.072 vs. 0.077	0.069 vs. 0.081	0.069 vs. 0.086
	Avg. p-value	0.00001	0.00017	0.00000	0.00000	0.00035	0.00058	0.00000	0.00024	0.00000	0.00056	0.00000	0.00007
	Conclusion	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>	>>

TABLE 8. Improvement percentage of the clustering in comparison to without-clustering using the FPA method.

	MAE (%)	MAPE (%)	RMSE (%)	MBRE (%)	MIBRE (%)
DP	38.53	25.63	55.68	26.06	15.84
IS	51.04	39.47	63.68	39.44	27.72
LT	37.78	30.39	55.98	30.28	17.82
OT	44.03	36.83	59.58	35.92	23.76
RS	40.45	34.63	59.26	33.8	22.77
k-means	36.73	32.09	54.19	32.39	19.8
Mean	41.43	33.17	58.06	32.98	21.29

TABLE 9. Percentage improvement of EEAC versus FPA.

	MAE (%)	MAPE (%)	RMSE (%)	MBRE (%)	MIBRE (%)
WC	9.18	7.17	11.18	4.93	0.99
DP	24.23	22.04	29.71	19.05	16.47
IS	29.04	20.27	31.58	16.28	15.07
LT	18.04	12.79	23.73	11.11	8.43
OT	25.55	18.49	31.31	15.38	12.99
RS	17.05	13.07	22.5	10.64	7.69
k-means	21.37	17.69	24.08	15.63	14.81
Mean	22.55	17.39	27.15	14.68	12.58

repository's August 2020 R1 dataset. The dataset covers a variety of software projects from various organizations throughout the world, each with its own set of features, fields, and size.

This study used the evaluation criteria MAE, MAPE, RMSE, MBRE, and MIBRE to assess the experiment's accuracy. According to published studies, i.e., [76], and [77], the above evaluation criteria are classified as unbiased evaluation criteria. Therefore, we can conclude that this study's experimental results are highly generalizable.

VII. CONCLUSION AND FUTURE WORK

In this paper, we investigated the effectiveness of clustering criteria for the FPA method. Specifically, we selected categorical variables (DT, DP, LT, IT, OT, and RS) and a

clustering algorithm (k-means) as our clustering methods. These methods all lead to better effort estimation than no clustering.

The evaluation results for these methods were compared, and it was determined that the categorical variable IS was the best clustering method used in this study. In addition, it should be emphasized that using a selective ML algorithm will provide better results than using the standard FPA method.

With the FPA method, the average improvement rate for the evaluation criteria for clustering over non-clustering reached as high as 58.06% (RMSE). The other criteria had average improvement rates of 51.04% (MAE), 39.47% (MAPE), 39.44% (MBRE), and 27.72% (MIBRE). Furthermore, the best clustering method (IS) had a percentage improvement high of 63.68% compared to the other clustering methods.

With the EEAC method, the average improvement rate for clustering compared to non-clustering reached a high of 65.53% for the RMSE. The other evaluation criteria had average improvement rates of 61.74% (MAE), 48.02% (MAPE), 46.67% (MBRE), and 38% (MIBRE). According to this criterion, the best clustering method (IS) had a percentage improvement high of 72.02% over the other clustering methods.

In a comparison between the FPA and EEAC methods, the average percentage improvement across all clusters peaked at 27.15% for the RMSE evaluation criteria; the other criteria had average improvements of 22.55% (MAE), 17.39% (MAPE), 14.68% (MBRE), and 12.58% (MIBRE). For the best clustering method (IS), the percentage improvement was lowest for the MIBRE (15.07%) and highest for the RMSE (31.58%).

For the ISBSG dataset, many different clustering methods can be used. In this study, we only chose six methods for evaluation, which was not an exhaustive list. Therefore, in the future, we will continue to study additional clustering methods, especially the Application Type criterion e mentioned in the related works section. This criterion should be considered due to its complexity. In addition, we will continue searching for the best ML clustering algorithm.

REFERENCES

[1] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.
[2] F. J. Heemstra, "Software cost estimation," *Inf. Softw. Technol.*, vol. 34, no. 10, pp. 627–639, Oct. 1992.
[3] T. Vera, S. F. Ochoa, and D. Perovich, "Survey of software development effort estimation taxonomies," Comput. Sci. Dept., Univ. Chile, Santiago, Chile, Tech. Rep., 2017.
[4] B. Khan, W. Khan, M. Arshad, and N. Jan, "Software cost estimation: Algorithmic and non-algorithmic approaches," *Int. J. Data Sci. Adv. Anal.*, vol. 2, no. 2, pp. 1–5, 2020.
[5] 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.
[6] P. Faria and E. Miranda, "Expert judgment in software estimation during the bid phase of a project—An exploratory survey," in *Proc. Joint Conf. 22nd Int. Workshop Softw. Meas. 7th Int. Conf. Softw. Process Product Meas.*, Oct. 2012, pp. 126–131.
[7] B. Boehm, *Software Engineering Economics*. New Jersey, NY, USA: Prentice-Hall, 1981.
[8] F. W. Quentin and J. M. Koppelman, "Earned value project management," Project Manage. Inst., USA, Tech. Rep., 2010.
[9] Y. Okayama and L. D. Chirillo, "Product work breakdown structure," National Shipbuilding Research Program, Todd Pacific Shipyards Corporation, Los Angeles, CA, USA, Tech. Rep., 1982.
[10] J. Grenning, "Planning poker or how to avoid analysis paralysis while release planning," Renaissance Softw. Consulting, Hawthorn Woods, IL, USA, Tech. Rep., 2002, pp. 1–3.
[11] M. Cohn, *Agile Estimating and Planning*. New Jersey, NJ, USA: Prentice-Hall, 2005.
[12] L. H. Putnam, "A general empirical solution to the macro software sizing and estimating problem," *IEEE Trans. Softw. Eng.*, vol. SE-4, no. 4, pp. 345–361, Jul. 1978.
[13] A. S. Jamil, "Used SLIM model to estimate software cost," *Al-Mansour J.*, vol. 2007, no. 10, pp. 49–63, 2007.
[14] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece, *Software Cost Estimation With COCOMO II*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.
[15] G. Karner, "Metrics for objectory," Diploma, Univ. Linkoping, Sweden, Tech. Rep. LiTH-IDA-Ex-9344, Dec. 1993, vol. 21.
[16] *Software and Systems Engineering—Software Measurement—IFPUG Functional Size Measurement Method*, Standard ISO/IEC 20926:2009, 2009.
[17] "IFPUG function point counting rules," in *The IT Measurement Compendium: Estimating and Benchmarking Success With Functional Size Measurement*. Berlin, Germany: Springer, 2008, pp. 453–482.
[18] *Software Engineering—COSMIC: A Functional Size Measurement Method*, Standard ISO/IEC 19761:2011, 2011.
[19] *Information Technology—Systems and Software Engineering—FiSMA 1.1 Functional Size Measurement Method*, Standard ISO/IEC 29881:2010, 2010.
[20] *Software Engineering—MK II Function Point Analysis—Counting Practices Manual*, Standard ISO/IEC 20968:2002, 2002.
[21] *Software Engineering—NESMA Functional Size Measurement Method Version 2.1—Definitions and Counting Guidelines for the Application of Function Point Analysis*, Standard ISO/IEC 24570:2005, 2005.
[22] P. Sharma and J. Singh, "Systematic literature review on software effort estimation using machine learning approaches," in *Proc. Int. Conf. Next Gener. Comput. Inf. Syst. (ICNGCIS)*, Dec. 2017, pp. 43–47.
[23] V. V. Hai, H. L. T. K. Nhung, and H. T. Hoc, "A review of software effort estimation by using functional points analysis," in *Advances in Intelligent Systems and Computing*, vol. 1047. Cham, Switzerland: Springer, 2019.
[24] A. J. Albrecht, "Measuring application development productivity," in *Proc. IBM Appl. Develop. Symp.*, 1979, p. 83.
[25] V. V. Hai, H. L. T. K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "A new approach to calibrating functional complexity weight in software development effort estimation," *Computers*, vol. 11, no. 2, p. 15, Jan. 2022.
[26] M. O. Elish, "Assessment of voting ensemble for estimating software development effort," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2013, pp. 316–321.
[27] K. An and J. Meng, "Voting-averaged combination method for regressor ensemble," in *Proc. Int. Conf. Intell. Comput.*, 2010, pp. 540–546.
[28] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[29] P. Rai, S. Kumar, and D. K. Verma, "Prediction of software effort in the early stage of software development: A hybrid model," *IEEE Can. J. Electr. Comput. Eng.*, vol. 44, no. 3, pp. 376–383, 2021.
[30] P. Silhavy, R. Silhavy, and Z. Prokopova, "Spectral clustering effect in software development effort estimation," *Symmetry*, vol. 13, no. 11, p. 2119, Nov. 2021.
[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, "Stepwise regression clustering method in function points estimation," in *Advances in Intelligent Systems and Computing*, vol. 859. Cham, Switzerland: Springer, 2019.
[33] *ISBSG Repository*, Int. Softw. Benchmarking Standards Group, South Melbourne, VI, Australia, Aug. 2020. [Online]. Available: https://www.isbsg.org/
[34] D. R. Anderson, D. J. Sweeney, and T. A. William, *Statistics for Business and Economics*. Mason, OH, USA: Thomson South-Western, 2009.

[35] A. Ross and V. L. Willson, "Paired samples t-test," in *Basic and Advanced Statistical Tests*. Rotterdam, The Netherlands: Sense, 2017, pp. 17–19.

[36] 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.

[37] 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.* Cham, Switzerland: Springer, 2017, pp. 479–490.

[38] O. F. Sarac and N. Duru, "A novel method for software effort estimation: Estimating with boundaries," in *Proc. IEEE INISTA*, Jun. 2013, pp. 1–5.

[39] 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.

[40] S. K. Sehra, J. Kaur, Y. S. Brar, and N. Kaur, "Analysis of data mining techniques for software effort estimation," in *Proc. 11th Int. Conf. Inf. Technol., New Generat.*, Apr. 2014, pp. 633–638.

[41] W. D. Sunindyo and C. Rudiyanto, "Improvement of COCOMO II model to increase the accuracy of effort estimation," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Jul. 2019, pp. 140–145.

[42] M. Fernández-Diego and J.-M. Torralba-Martínez, "Discretization methods for NBC in effort estimation: An empirical comparison based on ISBSG projects," in *Proc. ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, 2012, pp. 103–106.

[43] K. Iwata, T. Nakashima, Y. Anan, and N. Ishii, "Applying machine learning classification to determining outliers in effort for embedded software development projects," in *Proc. 6th Int. Conf. Comput. Sci./Intell. Appl. Informat. (CSII)*, May 2019, pp. 78–83.

[44] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[45] J. Huang, Y.-F. Li, J. W. Keung, Y. T. Yu, and W. K. Chan, "An empirical analysis of three-stage data-preprocessing for analogy-based software effort estimation on the ISBSG data," in *Proc. IEEE Int. Conf. Softw. Quality, Rel. Secur. (QRS)*, Jul. 2017, pp. 442–449.

[46] K. Meridji, K. T. Al-Sarayreh, M. Abu-Arqoub, and W. M. Hadi, "Exploration of development projects of renewable energy applications in the ISBSG dataset: Empirical study," in *Proc. 2nd Int. Conf. Appl. Inf. Technol. Developing Renew. Energy Processes Syst. (IT-DREPS)*, Dec. 2017, pp. 1–6.

[47] Z. Prokopova, P. Silhavy, and R. Silhavy, "Influence analysis of selected factors in the function point work effort estimation," in *Advances in Intelligent Systems and Computing*. Szczecin, Poland: Springer-Verlag, 2019, pp. 112–124.

[48] K. Kaewbanjong and S. Intakosum, "Statistical analysis with prediction models of user satisfaction in software project factors," in *Proc. 17th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Jun. 2020, pp. 637–643.

[49] S. P. Pillai, S. D. Madhukumar, and T. Radharamanan, "Consolidating evidence based studies in software cost/effort estimation—A tertiary study," in *Proc. IEEE Region 10 Conf.*, Nov. 2017, pp. 833–838.

[50] C. Lopez-Martin, "Feedforward neural networks for predicting the duration of maintained software projects," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 528–533.

[51] Y. Li, L. Shi, J. Hu, Q. Wang, and J. Zhai, "An empirical study to revisit productivity across different programming languages," in *Proc. 24th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2017, pp. 526–533.

[52] M. Fernández-Diego and F. González-Ladrón-de-Guevara, "Application of mutual information-based sequential feature selection to ISBSG mixed data," *Softw. Quality J.*, vol. 26, no. 4, pp. 1299–1325, Dec. 2018.

[53] K. Usharani, V. V. Ananth, and D. Velmurugan, "A survey on software effort estimation," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 505–509.

[54] J. Liu, Q. Du, and J. Xu, "A learning-based adjustment model with genetic algorithm of function point estimation," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City, IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 51–58.

[55] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms," *J. Syst. Softw.*, vol. 137, pp. 184–196, Mar. 2018.

[56] J. I. S. Martinez, F. V. Souto, and M. R. Monje, "Analysis of automated estimation models using machine learning," in *Proc. 8th Int. Conf. Softw. Eng. Res. Innov. (CONISOFT)*, Nov. 2020, pp. 110–116.

[57] L. Song, L. L. Minku, and X. Yao, "Software effort interval prediction via Bayesian inference and synthetic bootstrap resampling," *ACM Trans. Softw. Eng. Methodol.*, vol. 28, no. 1, pp. 1–46, Feb. 2019.

[58] *Function Point Counting Practices Manual*, International Function Point Users Group, Westerville, OH, USA, 2010.

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

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

[61] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Nov. 1967, vol. 1, no. 233, pp. 281–297.

[62] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognit. Lett.*, vol. 25, no. 11, pp. 1293–1302, 2004.

[63] P. Bholowalia and A. Kumar, "EBK-means: A clustering technique based on elbow method and K-means in WSN," *Int. J. Comput. Appl.*, vol. 105, no. 9, pp. 17–24, 2014.

[64] Y. Q. Xie and R. M. Fang, "A K-means clustering algorithm for automatically obtaining K value," in *Proc. 3rd Int. Conf. Electr. Control Autom. Eng. (ECAE)*, 2018, pp. 135–139.

[65] *Yellowbrick*. Accessed: Apr. 2022. [Online]. Available: https://www.scikit-yb.org

[66] J. M. Lichtenberg and Ö. Şimşek, "Simple regression models," in *Proc. Mach. Learn. Res.*, vol. 58, 2016, pp. 13–25.

[67] M. Hammad and A. Alqaddoumi, "Features-level software effort estimation using machine learning algorithms," in *Proc. Int. Conf. Innov. Intell. Informat., Comput., Technol. (3ICT)*, Nov. 2018, pp. 1–3.

[68] *ISBSG Repository*, Int. Softw. Benchmarking Standards Group, Melbourne, VI, Australia, Aug. 2020.

[69] A. B. Nassif, L. F. Capretz, and D. Ho, "Estimating software effort based on use case point model using Sugeno fuzzy inference system," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 393–398.

[70] L. C. Briand, K. E. Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell, "An assessment and comparison of common software cost estimation modeling techniques," in *Proc. 21st Int. Conf. Softw. Eng. (ICSE)*, May 1999, pp. 313–323.

[71] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 820–827, Aug. 2012.

[72] M. Azzeh, A. B. Nassif, S. Banitaan, and F. Almasalha, "Pareto efficient multi-objective optimization for local tuning of analogy-based estimation," *Neural Comput. Appl.*, vol. 27, no. 8, pp. 2241–2265, Nov. 2016.

[73] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric software effort estimation based on optimizing correction factors and multiple linear regression," *IEEE Access*, vol. 10, pp. 2963–2986, 2022.

[74] M. Azzeh, A. B. Nassif, and I. B. Attili, "Predicting software effort from use case points: A systematic review," *Sci. Comput. Program.*, vol. 204, Apr. 2021, Art. no. 102596.

[75] 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.

[76] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature," *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.

[77] K. Todros and J. Tabrikian, "On order relations between lower bounds on the MSE of unbiased estimators," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 1663–1667.

VO VAN HAI was born in Quang Nam, Vietnam, in 1977. He received the B.S. degree in computer science from Dalat University, Dalat, Vietnam, in 1999, and the M.S. degree in computer science from the Faculty of Information Technology, Hue University, Hue City, Vietnam, in 2011. He is currently pursuing the Ph.D. degree in engineering informatics with the Department of Computer and Communication Systems, Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic. From 2001 to 2018, he was a Lecturer with the Faculty of Information Technology, Industrial University of Ho Chi Minh City, Vietnam. His research interests include software engineering, software effort estimation, and software development.

HO LE THI KIM NHUNG was born in Ho Chi Minh City, Vietnam, in 1986. She received the B.S. and M.S. degrees in information systems from the University of Science (HCMUS), Vietnam, in 2010. She is currently pursuing the Ph.D. degree in software engineering with Tomas Bata University in Zlin, Czech Republic. From 2010 to 2018, she was a Lecturer with the Department of Information Systems, Faculty of Information Systems, HCMUS. Her research interests include database management systems, software engineering, and software effort estimation methods based on use case points.

ZDENKA PROKOPOVA was born in Rimavska Sobota, Slovakia, in 1965. She received the master's degree in automatic control theory and the Ph.D. degree in technical cybernetics from Slovak Technical University, in 1988 and 1993, respectively. She worked as an Assistant at Slovak Technical University, from 1988 to 1993. From 1993 to 1995, she worked as a Programmer of database systems with the Data-Lock business firm. From 1995 to 2000, she worked as a Lecturer at the Brno University of Technology. Since 2001, she has been at the Faculty of Applied Informatics, Tomas Bata University in Zlin. She currently holds the position of 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.

RADEK SILHAVY was born in Vsetin, 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 Zlin, in 2004, 2006, and 2009, respectively. He is currently an Associate Professor and a Researcher with the Computer and Communication Systems Department, Tomas Bata University in Zlin. His research interests include effort estimation in software engineering and empirical methods in software and system engineering.

PETR SILHAVY was born in Vsetíìn, 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 Zlin, in 2004, 2006, and 2009, respectively. From 1999 to 2018, he was the CTO of a company that specializes in database systems development. He currently holds the position of an Associate Professor at Tomas Bata University in Zlin. His research interests include software engineering, empirical software engineering, system engineering, data mining, and database systems.