Parametric Software Effort Estimation Based on Optimizing Correction Factors and Multiple Linear Regression

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

Faculty of Applied Informatics, Tomas Bata University in Zlin, 755 01 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/2021/001 and Project RO30216002025.

ABSTRACT Context: Effort estimation is one of the essential phases that must be accurately predicted in the early stage of software project development. Currently, solving problems that affect the estimation accuracy of Use Case Points-based methods is still a challenge to be addressed. Objective: This paper proposes a parametric software effort estimation model based on Optimizing Correction Factors and Multiple Regression Models to minimize the estimation error and the influence of unsystematic noise, which has not been considered in previous studies. The proposed method takes advantage of the Least Squared Regression models and Multiple Linear Regression models on the Use Case Points-based elements. Method: We have conducted experimental research to evaluate the estimation accuracy of the proposed method and compare it with three previous related methods, i.e., 1) the baseline estimation method – Use Case Points, 2) Optimizing Correction Factors, and 3) Algorithmic Optimization Method. Experiments were performed on datasets (Dataset D1, Dataset D2, and Dataset D3). The estimation accuracy of the methods was analysed by applying various unbiased evaluation criteria and statistical tests. Results: The results proved that the proposed method outperformed the other methods in improving estimation accuracy. Statistically, the results proved to be significantly superior to the three compared methods based on all tested datasets. Conclusion: Based on our obtained results, the proposed method has a high estimation capability and is considered a helpful method for project managers during the estimation phase. The correction factors are considered in the estimation process.

INDEX TERMS Algorithmic optimization, multiple linear regression, optimizing correction factors, software development effort estimation, use case points.

I. INTRODUCTION

Software project development has become extremely complicated, and the necessary competence in this industry is high, which requires the skills of highly qualified people. In past decades, to complete a project and deliver it to the customer on time, schedule, and budget, project managers had to estimate the cost of the software product, effort, and project duration or defect density [1]. The 2018 Standish Group CHAOS showed that many software companies could not give the correct practical software cost and completed their projects late schedule and over budget - (48%-65%) or failed to complete them at all - (48%-56%) [2]. The results indicated that most projects’ actual efforts and schedules are over budget compared to the estimates. The project budget plays a role in competitiveness, which means that using an effort estimation method in a software company is mandatory.

Software Development Effort Estimation (SDEE) is a crucial activity in the early stages of software development that plays an important role in the project’s overall success. The SDEE manages project activities before the project begins, specifically designing the project plan and managing the budget. To obtain accurate estimates, a project manager must select an appropriate method and then customize or configure
TABLE 1. Non-algorithmic estimation models.

| Estimation method                        | Type            | Description                                                                                                                                 |
|------------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Expert judgement                         | Non-algorithmic | The estimate is based on an expert’s experience, knowledge, motivation, knowledge about the field, and the exchange between analysts and experts. Several studies give the guidelines of judgement-based effort estimation. |
| Analogy-based                            | Non-algorithmic | This is a form of the Cased Based Reasoning method. The method mainly compares a project’s significant features and attributes because it relies on past information from comparable projects. |
| Bottom-Up and Top-down Approach, Price-to-win | Non-algorithmic | Estimates are based entirely on software project budgets and broken down by project module (top-down) or predicted as the sum of project module estimates (bottom-up). |
| Wideband Delphi                          | Non-algorithmic | This work breakdown structure-based method is a team-based cost estimation method. The effort is evaluated based on team agreement. |
| Planning Poker                           | Non-algorithmic | Estimation, like the Wideband Delphi method, is a consensus among team members. |

TABLE 2. Algorithmic estimation models.

| Estimation method                        | Type      | Description                                                                                                                                 |
|------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Source Lines of Code (SLOC)              | Algorithmic | This approach counts the number of lines in the program’s source code. |
| Function Point Analysis                  | Algorithmic | This method measures the complexity and size of a software system as the functions that the system provides to the end-user. It is based on five function types: Internal Logical File (ILF), External Interface File (EIF), External Input (EI), External Output (EO), and External Inquiry (EQ). |
| Object Point                             | Algorithmic | A weighted total Object-Point count is based on the number and the complexity of objects - (e.g., screens, reports, 3GL components). |
| Constructive Cost Model (COCOMO)         | Algorithmic | This approach uses mathematical equations and calculations to estimate the cost of a project. It provides estimates concerning the effort and schedule for a software project. |
| Use Case Points (UCP)                    | Algorithmic | This approach is based on the elements of the system use cases with technical and environmental aspects. The method is based on a calculation with four elements: Unadjusted Use Case Weight (UUCW), Unadjusted Actor Weight (UAW), Technical Complexity Factor (TCF), and Environmental Complexity Factor (ECF). |
| Software Life Cycle Management (SLIM)    | Algorithmic | This model is based on the Norden/Rayleigh function. SLIM can record analysis data from historical projects, which is then used to calibrate and build the workforce in the existing dataset by answering a series of questions. |

Accurate effort estimation is still an open issue. An effort estimation method is used to minimize the project’s risks or reduce the risk of surprises during the project to the lowest value. It gives project managers good controlling decisions to ensure that the right amount of effort is allocated to the various activities during the project’s development life cycle. As a result, this has led many researchers to investigate software estimation for more accurate SDEE methods. Existing research efforts related to SDEE can be classified into three main groups:

1. Non-algorithmic models, also called non-parametric models, include Expert judgement, Analogue-based, Price-to-win, Top-down, Bottom-up, and Wideband Delphi. These models can develop an estimation by using an expert’s previous experience or historical projects to estimate software development costs. Descriptions of the non-algorithmic models are presented in TABLE 1.

2. Algorithmic models include Source Lines of Code (SLOC), Function Point Analysis, Object Point, Constructive Cost Model (COCOMO), and Use Case Points (UCP). These models use mathematical equations to estimate the software project cost. TABLE 2 below, describes the algorithmic models.

3. Machine learning models that have been exploited in SDEE include Artificial Neural Networks, Fuzzy Logic, Neuro-Fuzzy, Bayesian Network, Regression tree, Support Vector Machines, Genetic Algorithm. Some models are based on nonlinear properties and can learn from historical data and be trained to better estimate effort. Recently, these models have been used...
in combination with or as an alternative to algorithmic models.

In the requirements phase of the software lifecycle, Use Cases can be useful to measure the estimated effort at an early stage of a software project before obtaining the essential information [21]. As a result, the use of the Use Case for SDEE is widespread. In particular, a survey by Neil and Laplante [22] focused on the techniques used for the requirements elicitation, description, and modelling phases and found that over 50% of software projects used use cases in the early phase. The results of another review by Azzeh et al. [23] found that most studies focus on assessing Use Case Point (UCP) as a possible method for early SDEE. Researchers show interest in UCP-based approaches that are used as functional size metrics for effort estimation. UCP is used for object-oriented projects based on a structured scenario and actor analysis of the Use Case Model (UCM) [19].

A. PROBLEM FORMULATION

According to a systematic review of studies [23]–[26], the UCP is a promising effort estimation method during the early stages of software development and has many advantages for the software industry. However, from the project manager’s point of view, there are still two well-known issues in applying UCP methods that could be improved.

First, there is no standard for the specification of use cases. Specifically, use cases are written in natural language, and there is no rigorous procedure to examine the quality or fragmentation of use cases. This leads to the number of steps in a use case that may differ, and the accuracy of the estimation is affected. In addition, the accuracy of the estimate may be affected if there is more than one scenario in a use case. Therefore, to achieve accuracy in estimation, use cases need to be adjusted or calibrated. Almost all previous methods based on UCP for software effort estimation focus on constructing the method, reevaluating the complexity of the use case model, and reevaluating the complexity weights [27]–[37]. For example, researchers focus their attention on extending the UCP model by specifying new complexity levels for use case and actor weights [27], [28] or modifying existing complexity levels into more detailed options for effort prediction [29], [30]. Other studies calibrate complexity weights into different complexity levels [31]–[35]. Other approaches calculate the use case complexity based on transactions and paths [36], [37]. A transaction is defined as a stimulus and response event between an actor and the system. Paths are computed based on a cyclomatic complexity metric from the text representation of the use cases.

Second, the evaluation of Technical Complexity Factors (TCF) and Environmental Complexity Factors (ECF) depends on the experience of experts, which have a certain degree of uncertainty [28], [38], [39]–[45]. It is difficult to assign an appropriate value to an ECF because of a lack of relevant information. The reason is that an ECF is linked to the level of information and experience of a particular software development team. There are similar problems with the value assignment for a TCF. In particular, factor T10 (Concurrent) shows some difficulties. This technical factor could express parallel processing, parallel programming, or if the system works independently or interacts with several other parties. The assignment of values to this factor may not be accurate, as there are no guidelines in the UCP that explain this factor precisely. Huanca and Ore [39] recognized that the main factors affecting the estimation accuracy of the UCP are the ECFs and TCFs. They emphasized that the correction factors need to be reevaluated. Nassif et al. [46] also pointed out the necessity to refine these correction factors.

Combining machine learning to build SDEE models based on the original UCP formula could be a solution to enhance its accuracy. Some approaches [47]–[56] have also explored variant models, particularly using regression models to optimize estimation accuracy based on historical data. These approaches have many improvements that minimize the influence of human error during the analysis of the UCM and simplifying the original principles of the UCP.

The main drawback of the above methods is that none of them is comprehensive or provides better accuracy in estimating software effort under all situations. We developed the Optimization Correction Factors (OCF) method [38]. The method has investigated the Least Absolute Shrinkage and Selection Operator (LASSO) method [57], [58] to determine the best technical and environmental complexity factors that significantly affect the estimation accuracy of the UCP method. The OCF method can help project managers reduce risks in evaluating correction factors and produce estimation results close to the actual effort [38]. The method has shown that the Sum of Squared Errors (SSE) is improved by more than 16% compared to the UCP estimation method. The SSE was also examined at the 5% significance level, and the p-value (0.0245) was below the 5% significance level. When analysing the Percentage of Prediction within 25% (PRED (0.25)) of the OCF method, the UCP method has a PRED (0.25) of 0.38, while the OCF method reaches a PRED (0.25) of 0.66. Our method is considered the first step for more intensive research to evaluate the technical and environmental complexity factors in the UCP method. We believe that the accuracy of the OCF method may be different when performed with various other datasets, and therefore, a bottom-up experiment is performed in this paper.

However, the OCF method does not currently provide a highly significant refinement to the estimation. Our goal of modifying the OCF method is aimed at achieving more accurate estimates. The proposed method is inspired by the possibilities of using a standard estimation procedure for solving the considered problems discussed above. Therefore, in this work, we aim to apply the Least Squared Regression (LSR) models or the Multiple Linear Regression (MLR) models to improve the ability of the OCF method to estimate the software size and minimize the prediction error. Our approach uses MLR on historical project data points to build regression
models and minimize errors in the integration process or recursion.

This study proposes a parametric software effort estimation model based on the OCF method and MLR for SDEE – the Extension of Optimizing Correction Factors (ExOCF) method – to minimize the estimation error more efficiently. The research questions answered are as follows:

RQ1: Is it possible to modify the OCF method so that its estimation accuracy improves?

RQ2: Does the proposed method outperform a baseline UCP method and another tested method?

RQ3: Is the difference in the accuracy of the estimate using different methods statistically significant?

To answer the research questions, we conducted an experimental study to evaluate the estimation accuracy of the proposed method and compared it with three methods used in the literature. Each method is run on four different historical datasets (D1, D2, D3, and D4) based on various evaluation criteria (28-34). In this paper, we used statistical pairwise t-test comparisons to validate the accuracy of the proposed method. The following statistical hypothesis was tested:

H0: There is no significant difference in estimation capability between the proposed method and other estimation methods. This means that the estimation accuracy of the proposed method is not significantly different from that of the other methods.

H1: There is a significant difference in estimation capability between the proposed method and other estimation methods. This means that the estimation accuracy of the proposed method is significantly better than that of other methods.

B. CONTRIBUTIONS

The main contributions of this study are as follows:

1) Investigation of the LASSO algorithm’s use in exploring the best environmental and technical complexity factors on different datasets that improve the UCP size metric.

2) Machine learning techniques - LSR or MLR models - are combined with the OCF method to obtain better results in effort estimation. In this method, the software effort is a function of the OCF variables. The MLR formulation was created to estimate software effort values.

3) The results obtained by the proposed method are compared with three different estimation methods used in the literature. The methods are tested using the k-fold cross-validation technique. The training and testing datasets are the same for all methods. The datasets were obtained from the industry datasets of three data donors. To validate the accuracy of these methods, accuracy measures are chosen to avoid bias. The measurement criteria listed in Section 5 show how the evaluation metrics were selected. The experimental results show that the accuracy of the proposed method outperforms the other models.

The remaining sections are divided as follows: Section 2 introduces the related work. Section 3 presents the background of the methods used. The proposed effort estimation methods to achieve the research objectives are presented in Section 4. Section 5 describes the research methodology, including the presentation of the four datasets used in our experiments, the normalization of the data, the procedure of the experiments, and the evaluation criteria/metrics. The results of the experiments are presented in Section 6. Section 7 describes the threats to validity. Section 8 presents the conclusions. In the last section, we present future work.

II. RELATED WORK

Some problems related to the UCP model were presented in the previous section. In particular, many authors focused on adding more complexity levels for use case weight, actor weight, or both, discretizing the existing complexity levels, and calibrating the complexity weights. Kirmani and Wahid [27] added actor and use case weighting in the Re-UCP. They also added one extra rating level to the use case weighting system in UCP Sizing. Nunes et al. [28] identified six actor weightings in the iUCP. Wang et al. [29] integrated fuzzy set theory and Bayesian belief networks into the UCP model to extend the complexity levels of use cases. Periyasamy and Ghode [30] changed the actor complexity levels and reclassified the use case complexity in the e-UCP method. The UCPath [31] approach applies an activity-based costing method to all variables in the UCP method, except the productivity factor is changed to 8.2 person-hours. An adjustment approach to the UCP, called Adapted UCP (AUCP) [32], is applied for incremental development estimations in large-scale projects. Braz and Vergilio [33] proposed two methods: Use Case Size Points (USP), and Fuzzy Use Case Size Points (FUSP), by calibrating the internal level of the use case. A USP introduces new components by considering the structures of a use case, the number and weight of scenarios, actors, preconditions, and postconditions. A FUSP is an extended version of a USP that uses the Fuzzy Set theory to reduce some use case classification problems. Qi et al. [34] improved the estimation accuracy of the UCP by using Bayesian analysis to calibrate the case complexity weights. Rak et al. [35] proposed a model for effort estimation called Use Case Reusability (UCR). The method gives a new classification for use cases based on their reusability. References [36] and [37] proposed an improvement method by computing paths from the cycloomatic complexity of the use case scenario. Although there is a small difference in precision, these approaches show that paths and transactions can be useful in computing the UCP.

In terms of SDEE methods based on machine learning techniques, we categorized them into three groups as follows. The first group uses neural network models such as Cascade Correlation Neural Network (CCNN) model, Multilayer Perceptron (MLP), Fuzzy Logic, or Artificial Neural Network (ANN) to estimate software effort, as shown in [46], [48], [53], [54]. Nassif et al. [46] proposed
a UCP-based effort estimation model using fuzzy logic and neural networks to increase estimation accuracy. Reference [48] introduced a regression model using the Sugeno Fuzzy Inference System (FIS) approach to improve the estimation accuracy. The results show that an MMRE improvement of 11% can be obtained. Reference [53] proposed the CCNN model for use case diagrams. The proposed model was evaluated against the MLR and the UCP model with promising results as an alternative approach for SDEE.

Iraji and Motameni [54] presented the Adaptive Neuro-Fuzzy use Case Size Point (ANFUSP) model to estimate the effort for object-oriented software projects. The model results have less error than the UCP method.

The second group uses soft computing techniques with analogue-based estimation, such as [47], [55], [56]. Nassif et al. [47] proposed a model combining fuzzy logic and neural networks to increase the estimation accuracy of the UCP method. Here, the fuzzy logic used ten degrees

### TABLE 3. Related work on methods using the UCP (2016 onward).

| Cited study | Main author | Dataset | Contribution type | Estimation approach | Publication year |
|-------------|-------------|---------|-------------------|---------------------|------------------|
| [59]        | M. Azzeh et al. | DS1: 65 educational projects DS2: 45 industrial projects | Developing new estimation models on the original UCP method | Algorithmic | 2016 |
| [60]        | R. Silhavy et al. | DS1: 28 industrial projects from two datasets: Ochodék et al. [51] and Subravi et al. [39] DS2: 70 industrial projects from three data donors (D1, D2, and D3) [10] | Evaluating the accuracy of existing methods using historical datasets. | Algorithmic | 2018 |
| [10]        | R. Silhavy et al. | DS1: 28 industrial projects from two datasets: Ochodék et al. [51] and Subravi et al. [39] DS2: 70 industrial projects from three data donors (D1, D2, and D3) | Developing new estimation models on the original UCP method | Algorithmic | 2017 |
| [61]        | M. Azzeh et al. | 234 projects from three industrial datasets Ochodék et al. [52], Nassif et al. [47], Silhavy et al. [62] and from educational projects. | Comparison with other methods based on UCP. | Machine learning | 2018 |
| [62]        | M. Azzeh et al. | DS1: 65 educational projects DS2: 45 industrial projects DS3: merged DS1 and DS2 into one dataset | Developing new estimation models on the original UCP method | Algorithmic | 2017 |
| [63]        | M. Azzeh et al. | DS1: 65 educational projects DS2: 45 industrial projects | Evaluating the accuracy of existing methods using historical datasets. | Algorithmic | 2017 |
| [64]        | Sarwosri et al. | 186 projects from three industrial datasets Ochodék et al. [51], Nassif et al. [46], Silhavy et al. [10] and from educational projects. | Evaluating the accuracy of existing methods using historical datasets. | Algorithmic | 2018 |
| [65]        | M. Azzeh et al. | 2 projects, type of case studies. | Developing new estimation models on the original UCP method | Algorithmic | 2016 |
| [66]        | S.K. Rath et al. | DS1: 65 educational projects DS2: 45 industrial projects | Developing new estimation models on the original UCP method | Algorithmic | 2016 |
| [67]        | M. Badri et al. | 149 projects that obtained from A.B. Nassif et al. [46] | Developing new estimation models on the original UCP method | Algorithmic | 2017 |
| [68]        | Z. Prokopova et al. | 5 open source Java projects | Evaluating the accuracy of existing methods using historical datasets. | Machine learning | 2017 |
| [69]        | S. Bagheri et al. | 70 industrial projects from three data donors (D1, D2, and D3) [10] | Evaluating the accuracy of existing methods using historical datasets. | Machine learning | 2018 |
| [70]        | K. Qi et al. | 1 projects (Case studies) | Developing new estimation models on the original UCP method | Algorithmic | 2018 |
| [71]        | H.T. Hoc et al. | 22 educational projects | Developing new estimation models on the original UCP method | Machine learning | 2020 |
| [72]        | R. Silhavy et al. | 28 industrial projects | Developing new estimation models on the original UCP method | Machine learning | 2021 |
| [73]        | R. Silhavy et al. | 70 industrial projects from three data donors (D1, D2, and D3) [10] | Developing new estimation models on the original UCP method | Machine learning | 2017 |
| [74]        | A.B. Nassif et al. | 70 industrial projects from three data donors (D1, D2, and D3) [10] | Developing new estimation models on the original UCP method | Machine learning | 2019 |
| [38]        | H.L.T.K. Nhung et al. | 70 industrial projects from three data donors (D1, D2, and D3) [10] | Developing new estimation models on the original UCP method | Machine learning | 2020 |
for the complexity of the use cases, and the neural network was used to represent the input vectors of the UCP model. Bardsiri et al. [55] proposed a hybrid model based on Analogy Based Estimation (ABE) and Particle Swarm Optimization (PSO) algorithm. The model creates an attribute system that is weighted differently depending on the cluster. The results of the proposed model showed significantly improved accuracy of the estimates. Chiu and Huang [56] studied the effect of a genetic algorithm for adjusting the reused effort based on the distance between pairs of projects.

The last group applies regression models such as linear, nonlinear, and stepwise models [49]–[52]. Regression models can provide higher accuracy for effort estimation by examining the validity of UCP variables. Specifically, Nassif et al. [49] proposed a regression model based on the use case point size. The model considers the nonlinear relationship between software size in the UCP (Size) and the effort in person-hours (Effort), as well as the impact of the environmental complexity factors of a project on the productivity factor. The equation of the model is presented in (1).

$$\text{Effort} = 8.16 \times \frac{\text{Productivity}}{\text{Size}}^{1.17}$$ (1)

where the productivity value is between 0.4 and 1.3.

Jorgensen [50] reported all variables included in the models to illustrate the accuracy and bias variation of the SDEE methods using regression analysis. Ochodek et al. [51] simplified the UCP method by discarding the UAW, measuring the UCP based on steps, or calculating the total number of steps in use cases.

Silhavy et al. [52] developed the Algorithmic Optimisation Method (AOM) to increase the accuracy of the correction coefficients of the effort estimation process. The proposed method uses multiple least squares regression with all UCP elements. The equation of the AOM method is presented in (2).

$$\text{UCP}_{\text{AOM}} = \alpha_1 (UAW \times TCF \times ECF) + \alpha_2 (UUCW \times TCF \times ECF)$$ (2)

where $\alpha_1$, $\alpha_2$ are coefficient parameters from the regression model applied to historical projects.

The authors then conducted several experiments to investigate the significance of the UCP variables on two different datasets [9]. Residual analysis and stepwise multiple linear regression models were used to examine the influence of model complexity through correlation analysis. They proved that all UCP parameters were associated with the dependent variable to varying degrees and had significant estimation accuracy.

The regression equation is shown in (3-4), which contains an intercept, linear terms, and squared terms.

$$\text{Real}_{p20} \sim 1 + UUCW + ECF + UAW \times TCF + UAW^2$$
$$+ UUCW^2 + TCF^2 + ECF^2$$ (3)

$$\text{Real}_{p20} \sim 1 + TCF + ECF + UAW \times UUCW \times UAW^2$$
$$+ UUCW^2 + TCF^2 + ECF^2$$ (4)

The next part discusses the latest development (2016 onward) in effort estimation accuracy achieved using the UCP. TABLE 3 lists studies on estimation methods related to our work. The table also shows that the datasets used for three industrial projects include Ochodek et al. [51], Nassif et al. [46], and Silhavy et al. [9], and educational projects. Moreover, most studies focus on developing new estimation models for the original UCP method or evaluating the accuracy of existing methods using historical datasets.

The accuracy measures used in these studies (2016 onward) are summarized in TABLE 4. Frequent accuracy metrics were applied in these studies, and experiments were conducted, such as the Mean Magnitude of Relative Error (MMRE), Percentage of Prediction within x% (PRED (x)),

| Cited study | MMRE | PRED | MBRE | MIBRE | MAE | SA | MAPE | MSE | RMSE | NRMSE | SSE | $R^2$ | RSS |
|-------------|------|------|------|-------|-----|----|------|-----|------|-------|-----|------|-----|
| [59]        | x    | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [60]        |      | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [61]        | x    | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [62]        | x    | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [63]        | x    | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [64]        | x    | x    | x    | x     |     |    |      |     |      |       |     |      |     |
| [65]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [66]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [67]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [68]        |      |      |      |       |     |    |      |     |      |       |     |      |     |
| [69]        |      |      |      |       |     |    |      |     |      |       |     |      |     |
| [70]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [71]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [72]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [73]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [74]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |
| [75]        | x    |      |      |       |     |    |      |     |      |       |     |      |     |

TABLE 4. Summary of the accuracy measures used in sdee methods (2016 onward).
Sum of Squared Error (SSE), Standardized Accuracy (SA), Mean of Absolute Error (MAE), Median Magnitude of Relative Error (MdMRE), Mean Balanced Relative Error (MBRE) and Mean Inverse Balanced Relative Error (MIBRE), and Root Mean Square Error (RMSE).

III. BACKGROUND
A. MULTIPLE REGRESSION MODELS
Multiple regression models relate to estimating regression effort applications where there is more than one independent variable [3], [24], [50]. The purpose is to obtain the best-fit line that minimizes the regression model’s sum of squared residuals [75]. The form of the regression model is presented as a linear equation between a dependent variable and a set of independent variables is illustrated as a polynomial of degree n.

\[
y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_p x_p + \varepsilon
\]

Equation (5) can be rewritten as follows:

\[
y_i = \alpha_0 + \alpha_1 x_{i1} + \alpha_2 x_{i2} + \cdots + \alpha_p x_{ip} + \varepsilon_i, \quad i = 1, \ldots, m
\]

where \( y_i \) is the dependent variable, \( x_{i1}, \ldots, x_{ip} \) are the independent variables, \( \alpha_0 \) is the intercept parameter, and \( \alpha_1, \ldots, \alpha_p \) are the regression coefficients. These variables are unknown constants that must be estimated from the dataset, and \( \varepsilon_i \) are the error residuals.

Equation (5) can be rewritten as follows:

\[
y = \alpha X + \varepsilon
\]

where vector \( y \) and vector \( \varepsilon \) are column vectors of length \( m \), vector \( \alpha \) is a column vector of length \( p + 1 \), and matrix \( X \) is a \( m \times p + 1 \) matrix. Using LSR, vector \( \alpha \) is calculated as follows:

\[
\alpha = (X^T X)^{-1} X^T y
\]

Polynomial regression is a multiple regression in which the relationship between the dependent variable and \( p \) independent variables is illustrated as a polynomial of degree \( n \).

\[
y_i = \alpha_0 + \alpha_1 x_{i1} + \alpha_2 x_{i2}^2 + \cdots + \alpha_p x_{ip}^n + \varepsilon_i
\]

Based on the polynomial equation, a model can obtain a minimum error or minimum cost function. The model gives the best approximation of the relationship between the dependent and independent variables [55].

B. USE CASE POINTS
The original UCP method [19] is based on assigning weights to clustered actors and use cases (complexity weights). The elements of the UCP are shown in FIGURE 1.

The actor and use case employ three cluster classes (simple, average, and complex), as shown in TABLES 5 and 6.
TABLE 7. Technical complexity factors.

| $T_i$ | Description                          | Weight ($W_{T_i}$) |
|-------|--------------------------------------|--------------------|
| $T_1$ | Distributed System                   | 2                  |
| $T_2$ | Response Adjectives                  | 2                  |
| $T_3$ | End-Use Efficiency                   | 1                  |
| $T_4$ | Complex Processing                    | 1                  |
| $T_5$ | Reusable Code                        | 1                  |
| $T_6$ | Easy to install                      | 0.5                |
| $T_7$ | Easy to Use                          | 0.5                |
| $T_8$ | Portability                          | 2                  |
| $T_9$ | Easy to Change                       | 1                  |
| $T_{10}$ | Concurrency                         | 1                  |
| $T_{11}$ | Security Features                    | 1                  |
| $T_{12}$ | Access for Third Parties             | 1                  |
| $T_{13}$ | Special Training Facilities          | 1                  |

TABLE 8. Environmental complexity factors.

| $E_i$ | Description                          | Weight ($W_{E_i}$) |
|-------|--------------------------------------|--------------------|
| $E_1$ | Family with RUP                      | 1.5                |
| $E_2$ | Application Experience               | 0.5                |
| $E_3$ | Object-oriented Experience           | 1                  |
| $E_4$ | Lead Analyst Capability              | 0.5                |
| $E_5$ | Motivation                           | 1                  |
| $E_6$ | Stable Requirements                  | 2                  |
| $E_7$ | Part-time Workers                    | -1                 |
| $E_8$ | Difficult Programming Language       | 2                  |

where $T_i$ is the value of TCF $i$, $W_{T_i}$ is the complexity weight of technical factor $i$, $E_i$ is the value of ECF $i$, and $W_{E_j}$ is the complexity weight of environmental factor $i$.

The UCP is calculated using (14) as follows:

$$UCP = (UAW + UUCW) \times TCF \times ECF$$  \hspace{1cm} (14)

For SDEE, Karner suggested a factor of 20 man-hours per UCP to measure work effort. This is presented in (15).

$$Effort = UCP \times 20$$  \hspace{1cm} (15)

IV. THE PROPOSED METHOD

The ExOCF method can be divided into two phases. The first phase (Model Selection Phase) focuses on determining which of the technical and environmental complexity factors significantly affect the accuracy of the UCP based on the feature selection model. Then, two new regression formulas are created to calculate the selected factors through MLR models. The second phase (Fine-Tuning Phase) is conducted to optimize the OCF element obtained from phase 1. A detailed illustration of the ExOCF method is shown in FIGURE 2.

A. MODEL SELECTION PHASE

The Least Absolute Shrinkage and Selection Operator (LASSO) regression model [57], [58] is used to determine the factors selected in the regression analysis.

The LASSO estimate denoted $\hat{\beta}(\lambda)$ is determined as follows:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left( \frac{\| Y - X \beta \|_2^2}{n} + \lambda \| \beta \|_1 \right)$$

subject to $\sum_{j=1}^{k} |\beta_j| < t$ \hspace{1cm} (16)

where:

$$\| Y - X \beta \|_2^2 = \sum_{i=0}^{n} (Y_i - (X\beta)^i)^2$$ \hspace{1cm} (17)

$$\| \beta \|_1 = \sum_{j=1}^{k} |\beta_j|$$ \hspace{1cm} (18)

$\lambda \geq 0$ is the LASSO parameter that controls the strength of the penalty. The LASSO parameter $\lambda$ is determined by the Leave One Out Cross-Validation (LOO-CV) method [76], [77]. This parameter’s choice is adjusted based on the lowest possible prediction errors and a lack of bias towards the correction factors of the samples in the training set. The LASSO parameter relates directly to the number of selected correction factors via the number of nonzero $\beta$’s. The number of nonzero $\beta$ values can be changed by modifying the model parameter shown as t in (16).

The LASSO-based $n$ selected technical factors are named LaTF. A LASSO-technical factor (LaTF) can be described as follows:

$$LaTF = \alpha_0 + \sum_{i=1}^{n} \alpha_i \times LaT_i \times WL_i$$  \hspace{1cm} (19)

where $LaT_i$ is a technical factor that takes values from the interval [0, 5]. A value of “0” means that the technical complexity factor is irrelevant, while a value of “5” is essential. $WL_i$ is the weight of technical factor $i$. $\alpha_0, \alpha_i$ are regression coefficient parameters that are obtained from the MLR model.

The LASSO-based $m$ selected environmental factors are named LaEF. A LASSO-environmental factor (LaEF) can be determined as follows:

$$LaEF = \beta_0 + \sum_{j=1}^{m} \beta_j \times LaE_j \times WLe_j$$  \hspace{1cm} (20)

where $LaE_j$ is an environmental factor that corresponds to the environmental factors. $WLe_j$ is the weight of environmental factor $i$. $\alpha_0, \alpha_i$ are regression coefficient parameters that are obtained from the MLR model.

B. THE FINE-TUNING PHASE

In this phase, the effort estimation model is built using MLR as follows:

$$UCP_{ExOCF} = \gamma_1 (UAW \times LaTF \times LaEF) + \gamma_2 (UUCW \times LaTF \times LaEF)$$  \hspace{1cm} (21)

where $\gamma_1, \gamma_2$ are obtained according to two steps. First, the historical data points $(P_1, \ldots, P_n)$ are collected. The UAW, UUAW, LaTF, and LaEF elements for each project are identified. The result of this step is the collection of
values \((x_{i1}, x_{i2}, y_i), i = 1 \ldots n\), where \(y_i\) is the actual size (Real_P20 values) of the software project from a historical dataset.

\[
x_{i1} = (UAW_i \times LaTF_i \times ECF_i)
\]
\[
x_{i2} = (UUCW_i \times LaEF_i \times ECF_i)
\]

The LSR model is then used to obtain the regression coefficients \(\gamma_1\), \(\gamma_2\) as follows:

\[
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_n
\end{bmatrix} =
\begin{bmatrix}
  \gamma_1 \\
  \vdots \\
  \gamma_2
\end{bmatrix} \times
\begin{bmatrix}
  X_{11} & X_{12} \\
  \vdots & \vdots \\
  X_{n1} & X_{n2}
\end{bmatrix}
\]
\[
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_n
\end{bmatrix} = (X^T X)^{-1} X^T y
\]

Because \(y_i\) is a real value from a historical dataset, the regression coefficient values of \(\gamma_1\), \(\gamma_2\) can vary from each dataset. This means that when a historical dataset changes, this phase needs to be performed again to obtain new regression coefficient values. The second step of this phase will calculate the UAW, UUCW, LaTF, and LaEF of the current project, and (21) is applied with values \(\gamma_1\), \(\gamma_2\) from step 1 to estimate the UCP.

V. RESEARCH METHODOLOGY
In this section, we describe the empirical analysis of our research methodology. The section begins with a description of the datasets for the experiment, including the statistical characteristics in the four datasets and data normalization. The next part is the process of setting up the experiment to evaluate the software effort estimation methods.

A. DATASET DESCRIPTION
The proposed method was evaluated using a dataset that the authors collected and used [9]. The dataset is based on three data donations (D1, D2, and D3). The projects from each data donor differ in size (measured by the UCP). All data donors work in different government, health, and business sectors. The projects were developed in Java and C# programming languages. After analysing the dataset, we noticed that the Real_P20 of some projects varied extensively. FIGURE 3 presents a boxplot of Real_P20 in each dataset. Real_P20 is real effort in man-hours, divided by productivity (PF - man-hours per 1 UCP).

We observed a substantial difference in Real_P20 between the data donors. The distribution of Real_P20 is observed according to the data donors. In particular, data donor D1 had the largest projects, while data donor D3 had the smallest...
projects. The significant difference in Real_P20 makes the dataset heterogeneous. Therefore, applying the same model to all projects was not effective. We grouped projects according to data donors, making the datasets more homogeneous. Datasets (D1, D2, and D3) were provided by data vendors. Projects in each dataset may be understood as being local data for each of the companies.

In addition, we also evaluate the effect of mixing projects with different data providers, and a fourth dataset (D4) was also added, which combined all three datasets.

Statistical characteristics of the Real_P20 of the four datasets are described in TABLE 9, FIGURES 4-7. Median person-hours represent the workforce value of the project development period, which was applied from the project’s start date to acceptance date. The median Real_P20 shows the same value divided by PF = 20. It assumes that 20 person-hours corresponds to 1 UCP [19]. This transformation was made because data donors did not provide estimations using the UCP. The minimum Real_P20 and maximum Real_P20 describe the smallest and largest project sizes, respectively. The Real_P20 range describes the difference between the minimum Real_P20 and maximum Real_P20. The last column (n) indicates the number of projects in the dataset.

B. DATA NORMALIZATION

All variables in the four datasets were standardized using Min-Max normalization [78], [79] to ensure that they had the same influence degree. Variables usually have various ranges, which may have a negative impact on the learning step. Using (26-27), the variables are scaled and standardized from \((x_{min}, x_{max})\) to \((New_{min}, New_{max})\).

\[
x_j = \left( \frac{x_j - x_{min}}{x_{max} - x_{min}} \right) \times (New_{max} - New_{min}) + New_{min}
\]

\[
x_{max} = \max x_j_{1 \leq j \leq N}, \quad x_{min} = \min x_j_{1 \leq j \leq N}
\]  

C. EVALUATION CRITERIA

In SDEE, different criteria are needed to evaluate the estimation accuracy of methods. The SDEE’s accuracy in terms of the MMRE Men Magnitude of Error Relative to the estimate (MMER) [15], [48], [80] are the most commonly used metrics. However, these metrics may become biased [81], [82]. According to the systematic review of Azzeh et al. [23], the authors encouraged us to discard biased measures such as MMRE and MMER.

Therefore, to evaluate the proposed estimation method, we use alternative criteria that produce an unbiased and symmetric distribution, as follows: (28) Mean Absolute Error (MAE), (29) Mean Balance Relative Error (MBRE), (30) Mean Inverted Balance Relative Error (MIBRE), (31) Median of Magnitude of Relative Error (MdMRE) and (32) Root Mean Square Error (RMSE).

- Mean Absolute Error (MAE)

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|
\]
### TABLE 9. Dataset statistical characteristics.

| Dataset  | Median man-hours | Median Real P20 | Range Real P20 | Standard deviation | Minimum Real P20 | Maximum Real P20 | n  |
|----------|------------------|----------------|---------------|-------------------|-----------------|-----------------|----|
| Dataset D1 | 7252.000         | 362.600        | 60.300        | 18.820            | 338.200         | 398.500         | 27 |
| Dataset D2 | 6240.000         | 312.000        | 38.400        | 12.156            | 299.650         | 338.050         | 23 |
| Dataset D3 | 5878.000         | 293.900        | 10.500        | 3.287             | 288.750         | 299.250         | 20 |
| Dataset D4 | 6406.000         | 320.300        | 109.750       | 33.212            | 288.750         | 398.500         | 70 |

### Figures

**FIGURE 6.** Boxplot of Real P20 for dataset D3.

**FIGURE 7.** Boxplot of Real P20 for dataset D4.

- Mean Balance Relative Error (MBRE)
  
  \[ MBRE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{\min(y_i - \hat{y}_i)} \]  

- Mean Inverted Balance Relative Error (MIBRE)
  
  \[ MIBRE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{\max(y_i - \hat{y}_i)} \]  

- Median of Magnitude of Relative Error (MdMRE)
  
  \[ MdMRE = \text{median}_{i} \frac{|y_i - \hat{y}_i|}{y_i} \]  

- Root Mean Square Error (RMSE)
  
  \[ RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \]  

where \( n \) is the number of observations, \( y_i \) is the real known value, \( \hat{y}_i \) is the predicted value, and \( \varepsilon \) is the prediction error value.

On the other hand, we also used two measures to evaluate the accuracy of the estimation models, such as (33) Sum of Squares Errors (SSE) and (34) Percentage of Prediction within x% (PRED(x)). In particular, SSE is an important metric to estimate the variation in modelling error [75]. It is used because of its ability to describe errors for selected datasets. Second, PRED(x) is less biased towards underestimation and generally determines the same best method as the Standardized Accuracy (SA). According to the empirical evaluation of Idri et al. [83], an SDEE method that has high estimation accuracy (when PRED(x) values are high) is also reasonable (when SA values are high).

- Sum of Squares Errors (SSE)
  
  \[ SSE = \sum_{i=1}^{n} \varepsilon_i^2 \]  

- Percentage of Prediction within x% (PRED(x))
  
  \[ PRED(x) = \frac{1}{n} \sum_{i=1}^{n} \left\{ 1 \text{ if } \frac{|y_i - \hat{y}_i|}{y_i} \leq x; 0 \text{ otherwise} \right\} \]  

### D. EXPERIMENTAL SETUP

In this section, we present a series of experimental setups to evaluate the effectiveness of software effort estimation methods (see FIGURE 8). In step 1, the methods in this research direction are installed for experiments as follows:

- ExOCF (proposed in Section 4)

The results are compared with estimation methods as follows:

- OCF [38]
- UCP [19]
- AOM [52]

To evaluate the estimation accuracy, we experimented with five different runs (5-fold cross-validation). The comparisons of the effort estimation accuracy of each method are then based on the average results of these five runs.

In step 2, the results were then evaluated using some evaluation criteria, SSE, PRED (0.25), MAE, MBRE, MIBRE,
VI. RESULTS AND DISCUSSIONS

This section presents the empirical results obtained from the analysis of the correction factors that significantly affect the accuracy of the UCP-based SDEE methods and presents the answer to our research questions.

The purpose of the results is to minimize the SSE, MdMRE, MAE, MBRE, MIBRE, and RMSE and maximize the PRED (0.25). Specifically, low values for the SSE, MdMRE, MAE, MBRE, MIBRE, and RMSE show good accuracy of the UCP-based SDEE methods.
results. In contrast, high values for the PRED (0.25) show good results.

We also performed paired samples t-test comparisons [84]–[87] to investigate whether the ExOCF method is significantly different from the other methods to confirm the evaluation conclusions. The notations $\gg$, $\ll$ and $\approx$ are used to express the empirical conclusion based on their p-value, which indicate the statistical superiority, inferiority, and similarity of the ExOCF method compared to each of the other methods, respectively. When the p-value $\leq 0.05$, we can conclude that the difference in estimation accuracy between the ExOCF method and each other method is significant. In this work, we use the SSE, PRED (0.25), MdMRE, MAE, MBRE, MIBRE, and RMSE results as the sample test set for each method.

A. CORRECTION FACTORS ANALYSIS

Feature selection using LASSO is conducted to determine the best technical and environmental factors for each dataset.
FIGURE 17. Coefficient estimates on TCF from LASSO regression (D1 dataset).

FIGURE 18. Coefficient estimates on ECF from LASSO regression (D1 dataset).

FIGURE 19. Coefficient estimates on TCF from LASSO regression (D2 dataset).

FIGURE 20. Coefficient estimates on ECF from LASSO regression (D2 dataset).

FIGURES 9-16 show a sequence of different R-squared values in proportion to different values of $\lambda$. The selected $\lambda$ value is determined using the LOO-CV technique at which the R-squared reaches its highest value. The highest R-squared represents the goodness of fit of the LaTF and LaEF regression models.

FIGURES 17-24 show the selected technical and environmental factors corresponding to the determined $\lambda$ values.

The details of the technical and environmental factors selected in each dataset with the determined $\lambda$, as well as their coefficient estimates, are shown in TABLES 10 and 11. Specifically, there are nine remaining technical correction factors in the D1 dataset at $\lambda_{TCF} = 0.000231$, T1, T3, T4, T5, T6, T9, T10, and T11, and at $\lambda_{ECF} = 0.000177$ there are six remaining environmental factors, ENV3 to ENV8. In the D2 dataset, the eight selected technical factors at $\lambda_{TCF} = 0.000268$ are T1, T3, T4, T5, T6, T9, T10, and T11, and the selected environmental factors at $\lambda_{ECF} = 0.000192$ are ENV3 to ENV8. In the D3 dataset, the seven selected technical factors at $\lambda_{TCF} = 0.000227$ are T1, T3, T4, T5, T7, T9, and T10, and the selected environmental factors at $\lambda_{ECF} = 0.000247$ are ENV3 to ENV8. In the D4 dataset, the nine selected technical factors at $\lambda_{TCF} = 0.000236$ are T1, T3, T4, T5, T6, T7, T9, T10, and T11, and the environmental factors at $\lambda_{ECF} = 0.00327$ are ENV3-ENV8.

B. RQ1
Is it possible to modify the OCF method so that its estimation accuracy improves?

The accuracies of empirical validation for the two methods are given in TABLES 12-15 over the four datasets. As the results show, we can comfortably confirm that the proposed ExOCF method produces the best SSE, MdMRE, MAE,
MBRE, MIBRE, RMSE, and PRED (0.25) values, which indicates that it is possible to modify the OCF method to improve its estimation accuracy.

Specifically, the average SSE results of the ExOCF method decreased by 3.02 times, 1.3 times, 3.1 times, and 1.55 times compared with those of the OCF method on datasets D1, D2, D3, and D4, respectively (see FIGURE 25). Similarly, compared to the OCF method, the ExOCF method increases the PRED (0.25) average values by 2.18 times, 1.64 times, 2.4 times, and 1.33 times on datasets D1, D2, D3, and D4, respectively (see FIGURE 26). The average MdMRE results of the ExOCF method are 2.08 times, 1.17 times, 1.92 times, and 1.28 times lower than those of the OCF method (see FIGURE 27). The average MAE results of the ExOCF method are reduced by 2.08 times, 1.17 times, 1.92, and 1.28 times those of OCF (see FIGURE 28). The average MBRE results of the ExOCF method are 2.72 times, 1.13 times, 1.71 times, and 1.31 times lower than those of the OCF method (see FIGURE 29).

Then, the average MIBRE results of the ExOCF method are 2.22 times, 1.1 times, 1.59 times, and 1.25 times lower
than those of the OCF method (see FIGURE 30). Finally, the average RMSE results of the ExOCF method are 1.76 times, 1.14 times, 1.78 times, and 1.25 times lower than those of the OCF method (see FIGURE 31).

Above all, we believe the use of the MLR model on the OCF variables has shown its effectiveness.

C. RQ2
Does the proposed method outperform a baseline UCP method and another tested method?

We measured the accuracy improvements achieved by the proposed ExOCF method over the baseline UCP method and another tested method, the AOM method. As shown in TABLES 12-15, the detailed results in each run prove that the proposed method performs better on the four datasets.

First, we consider the SSE and PRED (0.25) results of the experimental methods (see FIGURE 32 and FIGURE 33). Compared to the two methods (UCP and AOM), the proposed ExOCF method on the D1 dataset decreases the average SSE results by 3.64 times and 1.33 times and increases the average PRED (0.25) scores by 50% and 20.87%, respectively. Similarly, on the D2 dataset, the ExOCF method decreases the average SSE results by 2.92 times and increases the average PRED (0.25) scores by 33.33% compared to the other methods. On the D3 dataset, the average SSE of the ExOCF method results decreases by 6.43 times and 2.1 times. On the
D4 dataset, the SSE of the ExOCF method results decreases by 1.95 times and 1.36 times.

Next, as shown in FIGURES 34-38, we can comfortably observe that the proposed method outperforms all other methods with superior accuracy in the MAE, MdMRE, MBRE, MIBRE, and RMSE. In particular, on the D1 dataset, the proposed ExOCF method outperforms the UCP and AOM methods by 2.67 times and 1.65 times, respectively, for the MdMRE, by 3.38 times and 1.35 times, respectively, for the MBRE, by 2.39 times and 1.31 times, respectively, for the MIBRE, by 2.23 times and 1.27 times, respectively, for the MAE, and by 1.93 times and 1.16 times, respectively, for the RMSE. Similarly, the proposed method outperforms the UCP and AOM methods by 1.44 times and 1.3 times, respectively, for the MdMRE, by 1.2 times and 1.24 times, respectively, for the MBRE, by 1.19 times and 1.24 times, respectively, for the MIBRE, by 1.56 times and 1.61 times, respectively, for the MAE, and by 1.69 times and 1.71 times, respectively, for the RMSE, on the D2 dataset.

Moreover, the proposed method outperforms the UCP and AOM methods by 2.94 times and 1.38 times, respectively, for the MdMRE, by 2.49 times and 1.33 times, respectively, for the MBRE, by 2.06 times and 1.27 times, respectively, for the MIBRE, by 2.79 times and
1.48 times, respectively, for the MAE, and by 2.54 times and 1.44 times, respectively, for the RMSE on the D3 dataset.

On the D4 dataset, the proposed method outperforms the UCP and AOM methods by 1.49 times and 1.22 times, respectively, for the MAE, by 1.54 times and 1.36 times, respectively, for the MdMRE, by 1.49 times and 1.28 times, respectively, for the MBRE, by 1.41 times and 1.22 times, respectively, for the MdMRE, and by 1.39 times and 1.16 times, respectively, for the RMSE.

Above all, we can confidently confirm that the proposed method works better than the UCP and AOM methods.

### D. RQ3

Is the difference in the accuracy of the estimate using different methods statistically significant?

To answer RQ3, we examined the statistical properties of the estimates resulting from methods based on paired t-test comparisons, as shown in TABLES 16-19. The results confirm that the ExOCF method is statistically significant at the 95% confidence level compared to previous methods. Therefore, we are inclined to accept the alternative hypothesis (H1), which is also consistent with the results presented above.
VII. THREATS TO VALIDITY

The threats to the validity of this study, particularly to internal, external, and construct validity, can be summarized as follows:

A. INTERNAL VALIDITY

There is no superior approach to determine the regularization parameter $\lambda$ to extract a selected variable set, as shown in (16), before applying LASSO regression. In practice, the tuning parameter $\lambda$, which controls the strength of the penalty, has an important effect. In particular, if $\lambda$ is sufficiently large, the coefficients must be exactly zero, leading to the dimensionality being reduced. The larger the parameter $\lambda$ is, the greater the number of coefficients reduced to zero. Thus, we determined the $\lambda$ value based on the LOO-CV technique, where the R-squared reaches its highest value. This technique is used because of its deterministic property and suitability for small datasets. The dataset summarizes data from three donors for a long time period. Independent variables were partly submitted by the data vendors. The complete process of using case point calculation—mainly in the factor weights—may influence data quality and comparability between data donors. In past publications, datasets used were preprocessed, which may also have an impact on reliability.
FIGURE 38. The RMSE results of the UCP, AOM and ExOCF methods.

TABLE 16. The statistical t-test based on evaluation criteria (28-34) for the proposed method with each of the methods compared on the d1 dataset.

| Pairs of methods | ExOCF vs. UCP | ExOCF vs. OCF | ExOCF vs. AOM |
|------------------|---------------|---------------|---------------|
| **SSE results**  |               |               |               |
| Avg. SSE         | 26289.327     | 26289.327     | 26289.327     |
| vs. 9595.389     | 7955.097      | 35038.098     |
| Avg. p-value     | 0.00003       | 0.00006       | 0.00005       |
| Statistical conclusion | >> | >> | >> |
| **PRED (0.25) results** | | | |
| Avg. PRED (0.25) | 0.8           | 0.8           | 0.8           |
| vs. 0.4          | 0.37          | 0.63          |
| Avg. p-value     | 0.00003       | 0.0002        | 0.0000       |
| Statistical conclusion | >> | >> | >> |
| **MAE results**  |               |               |               |
| Avg. MAE         | 51.472        | 51.472        | 51.472        |
| vs. 114.673      | 107.173       | 65.18         |
| Avg. p-value     | 0.00008       | 0.0001        | 0.0001       |
| Statistical conclusion | >> | >> | >> |
| **MdMRE results**|               |               |               |
| Avg. MdMRE       | 0.107         | 0.107         | 0.107         |
| vs. 0.286        | 0.288         | 0.177         |
| Avg. p-value     | 0.00003       | 0.0002        | 0.001        |
| Statistical conclusion | >> | >> | >> |
| **MBRE results** |               |               |               |
| Avg. MBRE        | 0.170         | 0.170         | 0.170         |
| vs. 0.575        | 0.463         | 0.229         |
| Avg. p-value     | 0.00000       | 0.0002        | 0.0000       |
| Statistical conclusion | >> | >> | >> |
| **MBRE results** |               |               |               |
| Avg. MBRE        | 0.131         | 0.131         | 0.131         |
| vs. 0.313        | 0.291         | 0.171         |
| Avg. p-value     | 0.00004       | 0.00006       | 0.0000       |
| Statistical conclusion | >> | >> | >> |
| **RMSE results** |               |               |               |
| Avg. RMSE        | 65.397        | 65.397        | 65.397        |
| vs. 126.076      | 114.974       | 75.818        |
| Avg. p-value     | 0.00005       | 0.0001        | 0.0003       |
| Statistical conclusion | >> | >> | >> |

TABLE 17. The statistical t-test based on evaluation criteria (28-34) for the proposed method with each of the methods compared on the d2 dataset.

| Pairs of methods | ExOCF vs. UCP | ExOCF vs. OCF | ExOCF vs. AOM |
|------------------|---------------|---------------|---------------|
| **SSSE results** |               |               |               |
| Avg. SSE         | 47318.878     | 47318.878     | 47318.878     |
| vs. 138575.81    | 61534.375     | 138512.14     |
| Avg. p-value     | 0.007         | 0.003         | 0.003         |
| Statistical conclusion | >> | >> | >> |
| **PRED (0.25) results** | | | |
| Avg. PRED (0.25) | 0.72          | 0.72          | 0.72          |
| vs. 0.48         | 0.44          | 0.48          |
| Avg. p-value     | 0.001         | 0.002         | 0.001         |
| Statistical conclusion | >> | >> | >> |
| **MAE results**  |               |               |               |
| Avg. MAE         | 79.151        | 79.151        | 79.151        |
| vs. 124.24       | 92.7784       | 127.271       |
| Avg. p-value     | 0.001         | 0.03          | 0.0002        |
| Statistical conclusion | >> | >> | >> |
| **MdMRE results**|               |               |               |
| Avg. MdMRE       | 0.175         | 0.175         | 0.175         |
| vs. 0.252        | 0.311         | 0.285         |
| Avg. p-value     | 0.009         | 0.003         | 0.007         |
| Statistical conclusion | >> | >> | >> |
| **MBRE results** |               |               |               |
| Avg. MBRE        | 0.43          | 0.43          | 0.43          |
| vs. 0.518        | 0.487         | 0.531         |
| Avg. p-value     | 0.002         | 0.02          | 0.003         |
| Statistical conclusion | >> | >> | >> |
| **MBRE results** |               |               |               |
| Avg. MBRE        | 0.237         | 0.237         | 0.237         |
| vs. 0.284        | 0.263         | 0.294         |
| Avg. p-value     | 0.004         | 0.08          | 0.01          |
| Statistical conclusion | >> | >> | >> |
| **RMSE results** |               |               |               |
| Avg. RMSE        | 95.137        | 95.137        | 95.137        |
| vs. 161.097      | 108.833       | 163.112       |
| Avg. p-value     | 0.007         | 0.001         | 0.002         |
| Statistical conclusion | >> | >> | >> |

**B. CONSTRUCT VALIDITY**

Construction validity concerns generalizing the results. In the case of this study, the goal of experiments was to minimize an estimation error. The process is based on a common process for tuning an estimation model. Implementation of 5-fold cross-validation and dealing with four datasets allows us to generalize the results. To avoid monomethod bias, measurements using several evaluation criteria were used. Unbiased evaluation criteria and statistical pairwise t-tests were used to confirm the validity of the results, such as the SSE, PRED (0.25), MAE, MdMRE, MBRE, MIBRE and RMSE, which have no asymmetric error distribution. Thus, we can conclude that the experimental results of this study are highly generalizable.

**C. EXTERNAL VALIDITY, NAMELY, THE EXPERIMENTAL DATA**

Our experiments are based on a collection of publicly available datasets, so the conclusions should be convincing.
datasets are a small part of all datasets in the real world. Therefore, the conclusions about these datasets may not be appropriate for other datasets.

VIII. CONCLUSION
In this paper, our goal is that by modifying of our OCF method, more accurate estimates can be realized. The proposed ExOCF method is inspired by the possibilities of using a standard estimation procedure to solve the problem of the influence of human errors during the analysis of the UCM and simplifying the original principles of the UCP that the OCF method is having. Specifically, we used MLR models on historical project data points to build regression models and minimize errors in the integration process or recursion. The proposed method improves the OCF method’s ability to estimate a software size and minimizes the prediction error.

This paper analysed important research questions related to the proposed method, as mentioned in Section 1. Regarding RQ1, according to the accuracy of the empirical validation for both the OCF and ExOCF methods regarding the SSE, PRED (0.25), MdMRE, MAE, MBRE, and RMSE, we can confirm that the ExOCF method is superior to the OCF method over four datasets. Applying the MLR model to the OCF variables using the ExOCF method has improved its estimation accuracy. For RQ2, we can declare that the proposed method outperforms the UCP and AOM methods. For RQ3, to confirm the validity of the empirical results, we analysed the statistical properties based on paired t-test comparisons. It can be concluded that the proposed method is statistically significantly superior to the other methods.

In conclusion, we believe that the results can also be understood as beneficial for industrial application, as they

| Pairs of methods | ExOCF vs. UCP | ExOCF vs. OCF | ExOCF vs. AOM |
|------------------|--------------|--------------|--------------|
| SSE results      |              |              |              |
| Avg. SSE         | 12394.385    | 12394.385    | 12394.385    |
| Avg. p-value     | 0.002        | 0.001        | 0.01         |
| Statistical conclusion | >>           | >>           | >>           |
| PRED (0.25) results |            |              |              |
| Avg. PRED        | 0.85         | 0.85         | 0.85         |
| Avg. p-value     | 0.0003       | 0.001        | 0.001        |
| Statistical conclusion | >>           | >>           | >>           |
| MAE results      |              |              |              |
| Avg. MAE         | 46,437       | 46,437       | 46,437       |
| Avg. p-value     | 0.0004       | 0.001        | 0.002        |
| Statistical conclusion | >>           | >>           | >>           |
| MdMRE results    |              |              |              |
| Avg. MdMRE       | 0.159        | 0.159        | 0.159        |
| Avg. p-value     | 0.0003       | 0.002        | 0.007        |
| Statistical conclusion | >>           | >>           | >>           |
| MBRE results     |              |              |              |
| Avg. MBRE        | 0.178        | 0.178        | 0.178        |
| Avg. p-value     | 0.0006       | 0.009        | 0.007        |
| Statistical conclusion | >>           | >>           | >>           |
| RMSE results     |              |              |              |
| Avg. RMSE        | 54.679       | 54.679       | 54.679       |
| Avg. p-value     | 0.0001       | 0.001        | 0.003        |
| Statistical conclusion | >>           | >>           | >>           |

This paper analysed important research questions related to the proposed method, as mentioned in Section 1. Regarding RQ1, according to the accuracy of the empirical validation for both the OCF and ExOCF methods regarding the SSE, PRED (0.25), MdMRE, MAE, MBRE, and RMSE, we can confirm that the ExOCF method is superior to the OCF method over four datasets. Applying the MLR model to the OCF variables using the ExOCF method has improved its estimation accuracy. For RQ2, we can declare that the proposed method outperforms the UCP and AOM methods. For RQ3, to confirm the validity of the empirical results, we analysed the statistical properties based on paired t-test comparisons. It can be concluded that the proposed method is statistically significantly superior to the other methods.

In conclusion, we believe that the results can also be understood as beneficial for industrial application, as they

| Pairs of methods | ExOCF vs. UCP | ExOCF vs. OCF | ExOCF vs. AOM |
|------------------|--------------|--------------|--------------|
| SSE results      |              |              |              |
| Avg. SSE         | 79287.57     | 79287.57     | 79287.57     |
| Avg. p-value     | 0.0003       | 0.001        | 0.03         |
| Statistical conclusion | >>           | >>           | >>           |
| PRED (0.25) results |            |              |              |
| Avg. PRED        | 0.8          | 0.8          | 0.8          |
| Avg. p-value     | 0.002        | 0.01         | 0.006        |
| Statistical conclusion | >>           | >>           | >>           |
| MAE results      |              |              |              |
| Avg. MAE         | 59.4         | 59.4         | 59.4         |
| Avg. p-value     | 0.001        | 0.001        | 0.04         |
| Statistical conclusion | >>           | >>           | >>           |
| MdMRE results    |              |              |              |
| Avg. MdMRE       | 0.16         | 0.16         | 0.16         |
| Avg. p-value     | 0.005        | 0.01         | 0.01         |
| Statistical conclusion | >>           | >>           | >>           |
| MBRE results     |              |              |              |
| Avg. MBRE        | 0.252        | 0.252        | 0.252        |
| Avg. p-value     | 0.0005       | 0.0005       | 0.01         |
| Statistical conclusion | >>           | >>           | >>           |
| RMSE results     |              |              |              |
| Avg. RMSE        | 75.24        | 75.24        | 75.24        |
| Avg. p-value     | 0.0002       | 0.0008       | 0.02         |
| Statistical conclusion | >>           | >>           | >>           |
demonstrate that the proposed method leads to more accurate estimates of software size and effort.

IX. FUTURE WORK
In this paper, we proposed parametric software effort estimation based on Optimizing Correction Factors and Multiple Linear Regression for use in the early stages of software development. The ExOCF method uses the weighting of technical and environmental complexity factors as defined in the original UCP. These factors reflect how much productivity could lead to an increase in the estimation error for SDEE. MLR models: the heterogeneity of the historical data. This will be carried out in the future.

Another concern relates to an important aspect of deriving MLR models: the heterogeneity of the historical data. This could lead to an increase in the estimation error for SDEE. There are many solutions performed in the preprocessing step, such as outlier elimination, which is considered a solution performed in MLR-based effort estimation. However, the estimation accuracy is not significantly better because the difference in the distribution of historical data points cannot be resolved [56], [60], [88]. The use of clustering approaches is considered a solution to improve the estimation accuracy of the ExOCF method in our future work.

REFERENCES
[1] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, 1st ed. Hoboken, NJ, USA: Wiley, 1998.
[2] The Standish Group: CHAOS Chronicles, Standish Group Int., Boston, MA, USA, 2018.
[3] B. Boehm, C. Abts, and S. Chulani, “Software development cost estimation approaches—A survey,” Ann. Softw. Eng., vol. 10, no. 1, pp. 177–205, Nov. 2000, doi: 10.1023/A:1018917173582.
[4] A. Trendowicz and R. Jeffery, “Software project effort estimation,” in Foundations and Best Practice Guidelines of Success. Cham, Switzerland: Springer, 2014.
[5] M. Jørgensen and M. Shepperd, “A systematic review of software development cost estimation studies,” IEEE Trans. Softw. Eng., vol. 33, no. 1, pp. 33–53, Jan. 2006, doi: 10.1109/TSE.2007.256943.
[6] H. Le Thi Kim Nhung, H. T. Hoc, and V. V. Hai, “A review of use case based development effort estimation methods in the system development context,” in Intelligent Systems Applications in Software Engineering, Cham, Switzerland: Springer, 2019.
[7] T. Vera, S. F. Ochoa, and D. Perovich, “Survey of software development effort estimation taxonomies,” Dept. Comput. Sci., Univ. Chile, Santiago, Chile, Tech. Rep. TK_DCC-20180320-002. 2017.
[8] 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.
[9] 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, doi: 10.1016/j.jss.2016.11.029.
[10] C. Rush and R. Roy, “Expert judgement in cost estimation: Modelling the reasoning process,” Concurrent Eng., vol. 9, no. 4, pp. 271–284, Dec. 2001.
[11] K. Moløkken and M. Jørgensen, “Expert estimation of web-development projects: Are software professionals in technical roles more optimistic than those in non-technical roles?” Empirical Softw. Eng., vol. 10, no. 1, pp. 7–30, Jan. 2005, doi: 10.1007/BF03084321.46871.5c.
[12] M. Jørgensen and T. Halkjelsvik, “The effects of request formats on judgment-based effort estimation,” J. Syst. Softw., vol. 83, no. 1, pp. 29–36, Jan. 2010.
[13] M. Shepperd and C. Schofield, “Estimating software project effort using analogies,” IEEE Trans. Softw. Eng., vol. 23, no. 11, pp. 736–743, Nov. 1997, doi: 10.1109/32.637387.
[14] M. Jørgensen, “Top-down and bottom-up expert estimation of software development effort,” Inf. Softw. Technol., vol. 46, no. 1, pp. 3–16, Jan. 2004.
[15] B. Boehm, Software Engineering Economics. Upper Saddle River, NJ, USA: Prentice-Hall, 1981.
[16] M. Cohn, Agile Estimating and Planning. Upper Saddle River, NJ, USA: Prentice-Hall, 2005.
[17] International Function Point Users Group (IFPUG) Function Point Counting Practices Manual (FPPCM), Int. Function Point Users Group, Princeton, NJ, USA, 2005.
[18] G. C. Low and D. R. Jeffery, “Function points in the estimation and evaluation of the software process,” IEEE Trans. Softw. Eng., vol. 16, no. 1, pp. 64–71, Jan. 1990, doi: 10.1109/32.443646.
[19] G. Karner, “Resource estimation for objector projects,” Objective Syst. SF 17, pp. 1–13, 2000.
[20] 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, doi: 10.1109/TSE.1978.231521.
[21] A. Minkiewicz, Use Case Sizing. Mount Laurel, NJ, USA: PRICE Systems, 2015.
[22] C. J. Neil and P. A. Laplante, “Requirements engineering: The state of the practice,” IEEE Softw., vol. 20, no. 6, pp. 40–45, Nov./Dec. 2003.
[23] M. A. R. Azeez, A. B. Nassif, and F. B. Attili, “Predicting software effort from use case points: A systematic review,” Sci. Comput. Program., vol. 204, Apr. 2021, Art. no. 102596.
[24] Y. Singh, P. K. Bhatia, A. Kaur, and O. P. Sangwan, “A review of studies on effort estimation techniques of software development,” in Proc. 2nd Nat. Conf. Math. Techn., Emerg. Paradigms Electron. IT Ind., 2008, pp. 188–196.
[25] Y. Mahmood, N. Kama, and A. Azmi, “A systematic review of studies on use case points and expert-based estimation of software development effort,” J. Softw., Evol. Process, vol. 32, no. 7, pp. 1–20, Jul. 2020.
[26] A. P. Subriadi and P. A. Ningrum, “Critical review of the effort rate value in use case point method for estimating software development effort,” J. Theor. Appl. Inf. Technol., vol. 59, no. 3, pp. 735–744, Jan. 2014.
[27] M. Manzoor and A. Wahid, “Revised use case point (Re-UCP) model for software effort estimation,” in Proc. Int. Conf. Comput. Intell. Softw. Eng., vol. 6, no. 3, pp. 65–71, 2015, doi: 10.1545/IASCA.2015.006310.
[28] N. Nunes, L. Constantine, and R. Kazman, “IUCP: Estimating interactive-software project size with enhanced-use case points,” IEEE Softw., vol. 28, no. 4, pp. 64–73, Jul. 2011.
[29] F. Wang, X. Yang, X. Zhu, and L. Chen, “Extended use case points method for software cost estimation,” in Proc. Int. Conf. Comput. Intell. Softw. Eng., Dec. 2009, pp. 1–5, doi: 10.1109/CISE.2009.5364706.
[30] K. Periyasamy and A. Ghode, “Cost estimation using extended use case point (e-UCP) model,” in Proc. Int. Conf. Comput. Intell. Softw. Eng., Dec. 2009, pp. 1–5, doi: 10.1109/CISE.2009.5364515.
[31] S. Sholig, R. S. Dewi, and A. P. Subriadi, “A comparative study of software development size estimation method: UCPC vs function points,” in Proc. 4th Int. Softw. Syst. Eng. Conf., Jul. 2017, pp. 470–477. [Online]. Available: https://www.researchgate.net/profile/Sholig_Sholig
[32] P. Mohagheghi, B. Anda, and R. Conradi, “Effort estimation of use cases for incremental large-scale software development,” in Proc. 27th Int. Conf. Softw. Eng., May 2005, pp. 303–311, doi: 10.1109/ICSE.2005.1553573.
[33] M. Braz and S. Vergilio, “Software effort estimation based on use cases,” in Proc. 30th Annu. Int. Comput. Softw. Appl. Conf. (COMPASS), 2006, pp. 221–228, doi: 10.1109/COMPASS.2006.77.
[34] K. Qi, A. Hira, E. Venson, and B. W. Boehm, “Calibrating use case points using Bayesian analysis,” in Proc. 12th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas., Oct. 2018, pp. 1–10, doi: 10.1145/3239235.3239236.
[35] K. Rak, Ž. Car, and I. Lovrek, “Effort estimation model for software development projects based on use case reuse,” J. Softw., Evol. Process, vol. 31, no. 2, p. e2119, Feb. 2019, doi: 10.1002/smr.2119.
[36] G. Robiolo, C. Badano, and R. Orsoco, “Transactions and paths: Two use case based metrics which improve the early effort estimation,” in Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas., Oct. 2009, pp. 422–425, doi: 10.1109/IEEESEM.2009.5316021.
[37] L. Lavazza and G. Robiolo, “The role of the measure of functional complexity in effort estimation,” in Proc. 6th Int. Conf. Predictive Models Softw. Eng. (PROMISE), 2010, pp. 1–10, doi: 10.1109/1868328.1868338.

[38] H. L. T. K. Chung, H. T. Hoc, and V. V. Hai, “An evaluation of technical and environmental complexity factors for improving use case points estimation,” in Software Engineering Perspectives in Intelligent Systems (Advances in Intelligent Systems and Computing), Cham, Switzerland: Springer, 2020.

[39] L. M. Huanca and S. B. Ore, “Factors affecting the accuracy of use case points,” in Trends and Applications in Software Engineering, Cham, Switzerland: Springer, 2016, pp. 133–142.

[40] A. R. Gray and S. G. MacDonell, “A comparison of techniques for developing predictive models of software metrics,” Inf. Softw. Technol., vol. 39, no. 9, pp. 425–437, Jan. 1997.

[41] R. P. Pantoni, A. E. Mossin, and D. Brandao, “Task effort fuzzy estimation using a soft computing technique for predicting software effort based use case points,” IET Softw., vol. 12, no. 1, pp. 19–29, Feb. 2018.

[42] M. Azzez, 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.

[43] M. Azzez, A. B. Nassif, “Project productivity evaluation in early software effort estimation,” J. Softw., Evol. Process, vol. 30, no. 12, p. e2210, Dec. 2018.

[44] Sarwosri, M. J. A. Haiyan, M. Husein, and A. P. Ferza, “The development of method of the enhancement of technical factor (TF) and environmental factor (EF) to the use case point (UCP) to calculate the estimation of software’s effort,” in Proc. Int. Conf. Inf. Commun. Technol. Syst. (ICTS), Oct. 2016, pp. 203–207, doi: 10.1109/ICTS.2016.7910299.

[45] M. Azzez, A. B. Nassif, S. Banitaan, and C. Lopez-Martin, “Ensemble of learning project productivity in software effort based on use case points,” in Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2018, pp. 1427–1431.

[46] S. M. Satapathy, B. P. Acharya, and S. K. Rath, “Early stage software effort estimation using random forest technique based on use case points,” IET Softw., vol. 10, no. 1, pp. 10–17, Feb. 2016.

[47] M. Badri, L. Badri, W. Flageol, and F. Toure, “Source code size prediction using use case metrics: An empirical comparison with use case points,” Innov. Syst. Softw. Eng., vol. 13, nos. 2–3, pp. 143–159, Sep. 2017, doi: 10.1007/s11334-016-0285-7.

[48] Z. Prokopova, R. Silhavy, and P. Silhavy, “The effects of clustering to software size estimation for the use case points methods,” in Software Engineering Trends and Techniques in Intelligent Systems, vol. 575, Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-57141-6_51.

[49] S. Bagheri and A. Shameli-Sendi, “Software project estimation using improved use case point,” in Proc. IEEE 16th Int. Conf. Softw. Engg. Res., Manage. Appl. (SERAM), Jun. 2018, pp. 143–150, doi: 10.1109/SERA.2018.8477275.

[50] K. Qi and B. W. Boehm, “Detailed use case points (DUCPs): A size metric automatically countable from sequence and class diagrams,” in Proc. IEEE/ACM 10th Int. Workshop Modelling Softw. Eng. (MISE), May/Jun. 2018, pp. 17–24.

[51] H. T. Hoc, V. Van Hai, and H. L. T. K. Chung, “Adaoptimizer for the optimisation of use case points estimation,” in Software Engineering Perspectives in Intelligent Systems (Advances in Intelligent Systems and Computing), vol. 1294, Cham, Switzerland: Springer, 2020, pp. 747–756.

[52] R. Silhavy, P. Silhavy, and Z. Prokopova, “Using actors and use cases for software size estimation,” Electronics, vol. 10, no. 5, p. 592, Mar. 2021, doi: 10.3390/ELECTRONICS10050592.

[53] R. Silhavy, P. Silhavy, and Z. Prokopova, “Improving algorithmic optimisation method by spectral clustering,” in Proc. Comput. Sci. Line Conf., 2017, pp. 1–10.

[54] A. B. Nassif, M. Azzez, A. Idri, and A. Abran, “Software development effort estimation using regression fuzzy models,” Comput. Intell. Neurosci., vol. 2019, pp. 1–17, Feb. 2019.

[55] M. Golberg and H. Cho, Introduction to Regression Analysis, Las Vegas, NV, USA: Univ. of Nevada, 2010.

[56] A. W. Van der Vaart, S. Dudoit, and M. J. van der Laan, “Oracle inequalities for multi-fold cross validation,” Statist. Decis., vol. 24, no. 3, pp. 351–371, Dec. 2006, doi: 10.1524/stdm.2006.24.3.351.

[57] M. J. Van der Lann and S. Dudoit, “Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator,” Finite sample Oracle inequalities and examples,” U. C. Berkeley Division Biostatistics Work. Paper Ser., Berkeley, CA, USA, Tech. Rep. 130, 2003. [Online]. Available: http://biostats.bepress.com/ucbbiostat/paper130

[58] G. Guo and D. Neagu, “Similarity-based classifier combination for decision making,” in Proc. IEEE Int. Conf. Syst., Man Cybern., vol. 1, Oct. 2005, pp. 176–181, doi: 10.1109/ICSMC.2005.1571141.

[59] S. G. K. Patro and K. K. Sahu, “Normalization: A preprocessing stage,” 2015, arXiv:1503.06462.
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, Vietnam, in 2011. He is currently pursuing the Ph.D. degree in computer science and 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.

RADEK SILHAVY was born in Vsetín, 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. His major research interests include effort estimation in software engineering and empirical methods in software and system engineering.

ZDENKA PROKOPOVA was born in Rimavska Sobota, Slovak Republic, in 1965. She received the master’s degree in automatic control theory and the Technical Cybernetics Doctoral degree from Slovak Technical University, in 1988 and 1993, respectively. 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 in 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 holds the position of an associate professor at 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.

PETR SILHAVY was born in Vsetín, 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 appointed as a CTO in a company specialized on database systems development. He currently holds the position of an associate professor at Tomas Bata University in Zlin. His major research interests include software engineering, empirical software engineering, system engineering, data mining, and database systems.

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 and 2014, respectively. She is currently pursuing the Ph.D. degree in software engineering with Tomas Bata University in Zlin, Czech Republic. She holds the position of an associate professor at the Department of Computer and Communication Systems, Faculty of Information Systems, University of Science (HCMUS). Her research interests include database management systems, software and system engineering, and software effort estimation methods based on use case points.

RADEK SILHAVY was born in Vsetín, 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. His major research interests include effort estimation in software engineering and empirical methods in software and system engineering.