IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 27, 2021, accepted December 24, 2021, date of publication December 28, 2021,
date of current version January 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3139183

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

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

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 [11]. Several studies give the guidelines of judgement-based effort estimation [12], [13].
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 [14].
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) [15].
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 [16].
Planning Poker	Non-algorithmic	Estimation, like the Wideband Delphi method, is a consensus among team members [17].

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 [16].
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 [18], [19]. Estimation 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) [16].
Constructive Cost Model (COCOMO)	Algorithmic	This approach uses mathematical equations and calculations to estimate the cost of a project [16]. 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 [20]. 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 [21].

it to suit the type of software project that the organization will perform. However, the SDEE cannot be expected to have absolutely correct results [3], [4]. 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 [5], [6]. Existing research efforts related to SDEE can be classified into three main groups [3], [7], [8]:

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 [9]. 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 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 UCPabc [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 cyclomatic 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

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: Ochodek et al. [51] and Subriadi et al. [39] DS2: 70 industrial projects from three data donators (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: Ochodek et al. [51] and Subriadi et al. [39] DS2: 70 industrial projects from three data donators (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 Ochodek 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 Ochodek 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 donators (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 donators (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 donators (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 donators (D1, D2, and D3) [10]	Developing new estimation models on the original UCP method	Machine learning	2020

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 4. Summary of the accuracy measures used in sdee methods (2016 onward).

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	x	x		
[10]								x	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	x	x								
[67]	x	x											
[68]					x				x				
[69]								x	x		x		
[70]	x												
[71]	x	x											
[72]	x	x									x	x	
[73]	x	x					x	x			x	x	
[74]			x	x	x	x							
[38]							x				x	x	

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). The results show that PRED (0.25) and PRED (0.35) were improved by 16.5% and 25%, respectively.

$$Effort = \frac{8.16}{Productivity} \times Size^{1.17} \tag{1}$$

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).

$$UCP_{AOM} = \alpha_1 \left(UAW \times TCF \times ECF\right) + \alpha_2 (UUCW \times TCF \times ECF) \tag{2}$$

where α_1, α_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.

$$Real_{p20} \sim 1 + UUCW + ECF + UAW \times TCF + UAW^2 + UUCW^2 + TCF^2 + ECF^2 \tag{3}$$

$$Real_{p20} \sim 1 + TCF + ECF + UAW \times UUCW \times UAW^2 + UUCW^2 + TCF^2 + ECF^2 \tag{4}$$

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)), 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 p independent variables $X_1, X_2, \ldots, X_p$ as follows:

$$\begin{cases} y_1 = \alpha_0 + \alpha_1 X_{11} + \alpha_2 X_{12} + \cdots + \alpha_p X_{1p} + \varepsilon_1 \\ y_2 = \alpha_0 + \alpha_1 X_{21} + \alpha_2 X_{22} + \cdots + \alpha_p X_{2p} + \varepsilon_2 \\ \vdots \\ y_n = \alpha_0 + \alpha_1 X_{n1} + \alpha_2 X_{n2} + \cdots + \alpha_p X_{mp} + \varepsilon_n \end{cases} \tag{5}$$

i.e.

$$y_i = \alpha_0 + \alpha_1 X_{i1} + \alpha_2 X_{i2} + \cdots + \alpha_p X_{ip} + \varepsilon_i, \quad i = \overline{1 \ldots m} \tag{6}$$

where y_i is the dependent variable, $X_{i1}, \ldots, X_{ip}$ are the independent variables, α_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 ε_i are the error residuals.

Equation (5) can be rewritten as follows:

$$y = \alpha X + \varepsilon \tag{7}$$

where vector y and vector ε are column vectors of length m, vector α is a column vector of length $p + 1$, and matrix X is an m by $p + 1$ matrix. Using LSR, vector α is calculated as follows:

$$\alpha = (X^T X)^{-1} X^T y \tag{8}$$

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.

$$y_i = \alpha_0 + \alpha_1 X_{i1} + \alpha_2 X_{i2}^2 + \cdots + \alpha_p X_{ip}^n + \varepsilon_i \tag{9}$$

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.

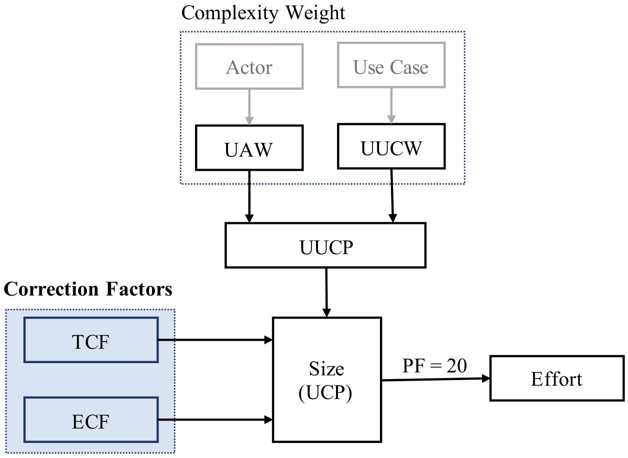

FIGURE 1. The process of the use case points method.

TABLE 5. Actor classifications and their complexity weights.

Actor classification	Description	Weight
Simple	The system through an API	1
Average	The system through a protocol	2
Complex	The system through a GUI	3

TABLE 6. Use case classifications and their complexity weights.

Use case classification	Number of transactions	Weight
Simple	(0, 4)	1
Average	<4, 7>	2
Complex	(7, ∞)	3

The sum of the weighted actors and use cases are created for the Unadjusted Actor Weight (UAW) and Unadjusted Use Case Weight (UUCW). The UAW and UUCW are calculated by using (10) and (11), respectively.

$$UAW = \sum_{i=1}^{3} at_i \times w_i \tag{10}$$

$$UUCW = \sum_{j=1}^{3} uc_j \times w_j \tag{11}$$

where αt_i is the number of actors in actor type i, w_i is the complexity weight of actor i, uc_j is the number of use cases in use case i, and w_j is the complexity weight of use case j.

Correction factors, i.e., TCFs and ECFs are used to describe the experience level of the software development team. The technical and complexity factors are shown in TABLES 7 and 8. The technical factors are calculated using (12), and the environmental factors are calculated using (13) as follows:

$$TCF = 0.6 + 0.01 \sum_{i=1}^{13} T_i \times Wt_i \tag{12}$$

$$ECF = 1.4 - 0.03 \sum_{i=1}^{8} E_i \times We_i \tag{13}$$

TABLE 7. Technical complexity factors.

T_i	Description	Weight (Wt_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 (We_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, Wt_i is the complexity weight of technical factor i, E_i is the value of ECF i, and We_j is the complexity weight of environmental factor i.

The UCP is calculated using (14) as follows:

$$UCP = (UAW + UUCW) \times TCF \times ECF \tag{14}$$

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

$$Effort = UCP \times 20 \tag{15}$$

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:

$$\hat{\beta}(\lambda) = \underset{\beta}{argmin}\left(\frac{\| Y - X\beta \|_2^2}{n} + \lambda \| \beta \|_1\right)$$
$$subject\ to \sum\nolimits_{j=1}^{k} |\beta_j| < \mathrm{t} \tag{16}$$

where:

$$\| Y - X\beta \|_2^2 = \sum\nolimits_{i=0}^{n} (Y_i - (X\beta)_i)^2 \tag{17}$$

$$\| \beta \|_1 = \sum\nolimits_{j=1}^{k} |\beta_j| \tag{18}$$

$\lambda \geq 0$ is the LASSO parameter that controls the strength of the penalty. The LASSO parameter λ 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 β's. The number of nonzero β 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:

$$LaTF = \alpha_0 + \sum\nolimits_{i=1}^{n} \alpha_i \times LaT_i \times WLt_i \tag{19}$$

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. WLt_i is the weight of technical factor i. α_0, α_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:

$$LaEF = \beta_0 + \sum\nolimits_{j=1}^{m} \beta_j \times LaE_j \times WLe_j \tag{20}$$

where LaE_j is an environmental factor that corresponds to the environmental factors. WLe_j is the weight of environmental factor i. α_o, α_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:

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

where γ_1, γ_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

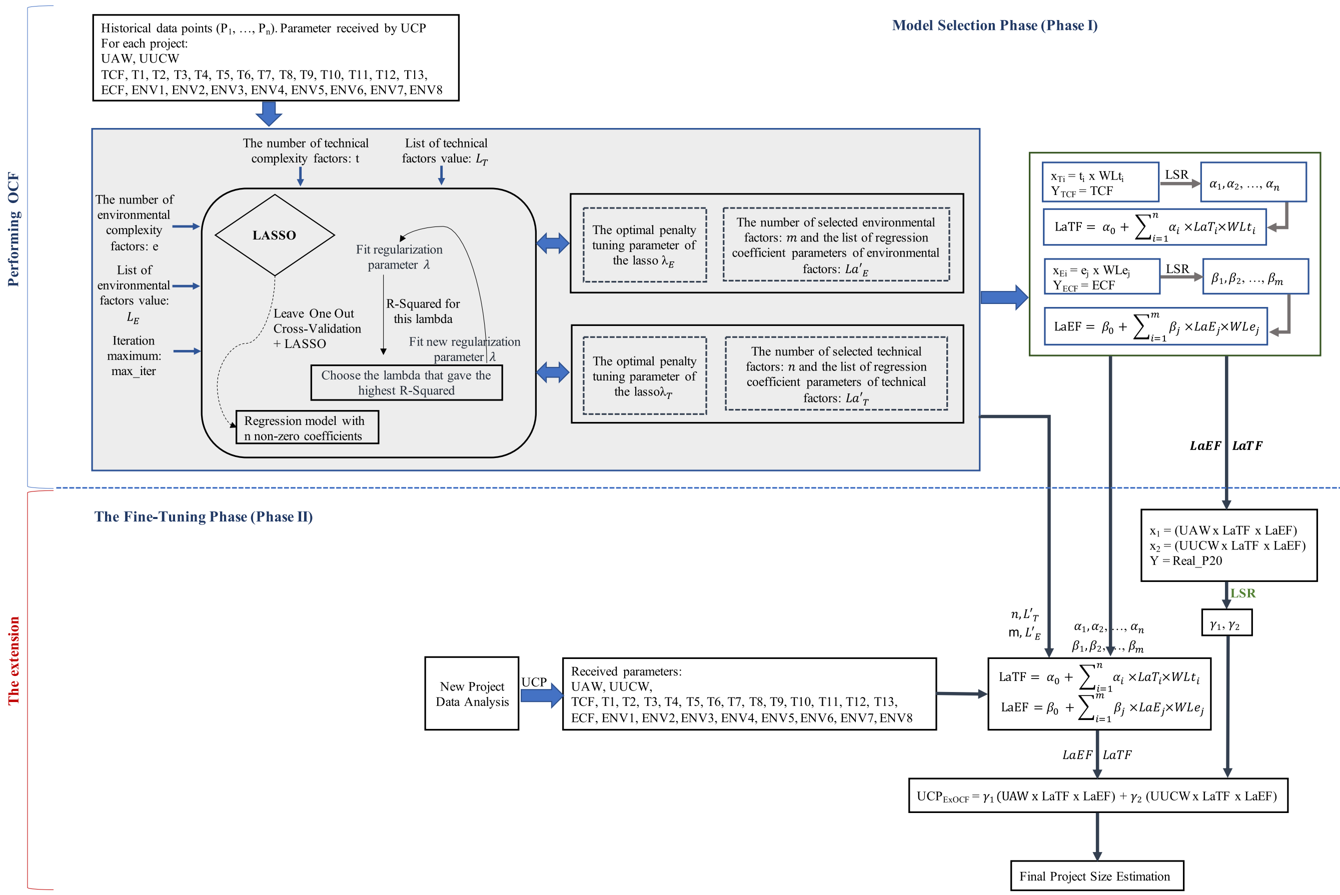

FIGURE 2. **Detailed illustration of the proposed ExOCF method.**

values $(x_{i1}, x_{i2}, y_i), i = \overline{1 \ldots n}$, where y_i is the actual size (Real_P20 values) of the software project from a historical dataset.

$$x_{i1} = (UAW_i \times LaTF_i \times ECF_i) \tag{22}$$
$$x_{i2} = (UUCW_i \times LaEF_i \times ECF_i) \tag{23}$$

The LSR model is then used to obtain the regression coefficients γ_1, γ_2 as follows:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} \times \begin{pmatrix} X_{11} & X_{12} \\ \vdots & \vdots \\ X_{n1} & X_{n2} \end{pmatrix} \tag{24}$$
$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = (X^T X)^{-1} X^T y \tag{25}$$

Because y_i is a real value from a historical dataset, the regression coefficient values of γ_1, γ_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 γ_1, γ_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 donors 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}).

$$x_j = \left(\frac{x_j - x_{min}}{x_{max} - x_{min}}\right) \times (New_{max} - New_{min}) + New_{min} \tag{26}$$

$$x_{max} = max\, x_{j_{1 \le j \le N}}, \quad x_{min} = min x_{j_{1 \le j \le N}} \tag{27}$$

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).

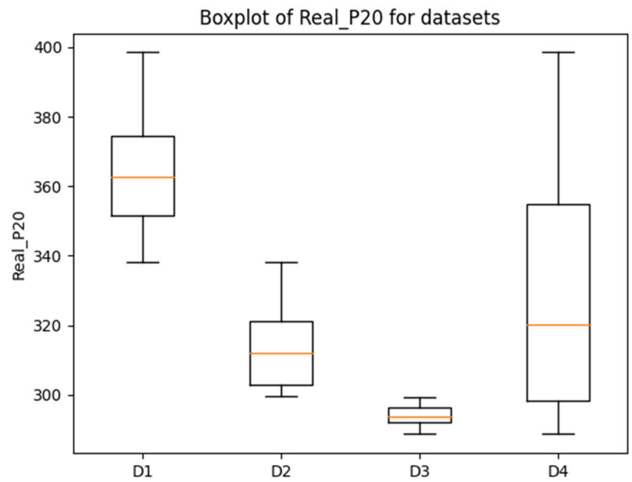

FIGURE 3. Boxplot of Real_P20 in each dataset.

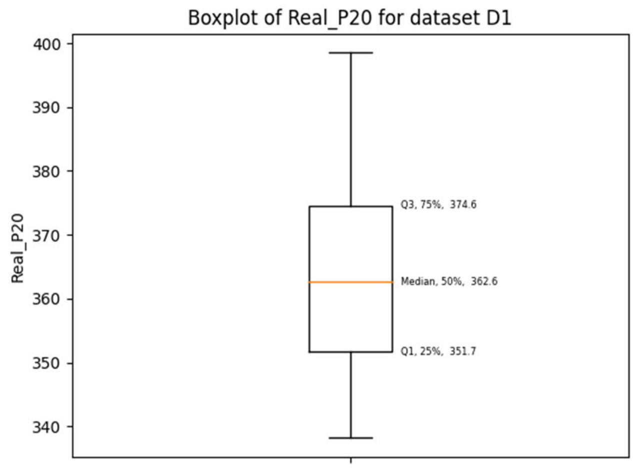

FIGURE 4. Boxplot of Real_P20 for dataset D1.

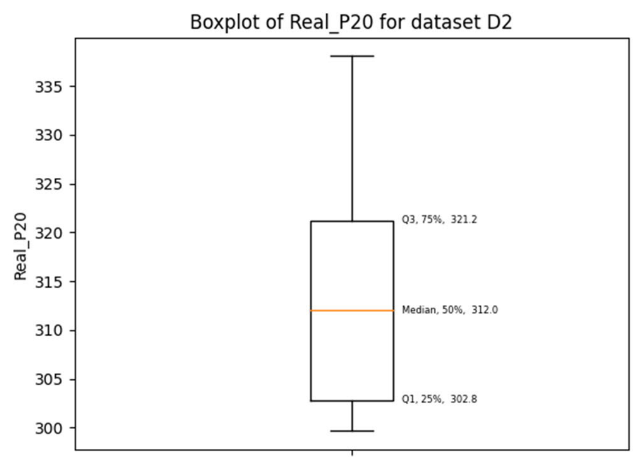

FIGURE 5. Boxplot of Real_P20 for dataset D2.

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum\nolimits_{i=1}^{n} |y_i - \hat{y}_i| \tag{28}$$

TABLE 9. Dataset statistical characteristics.

	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

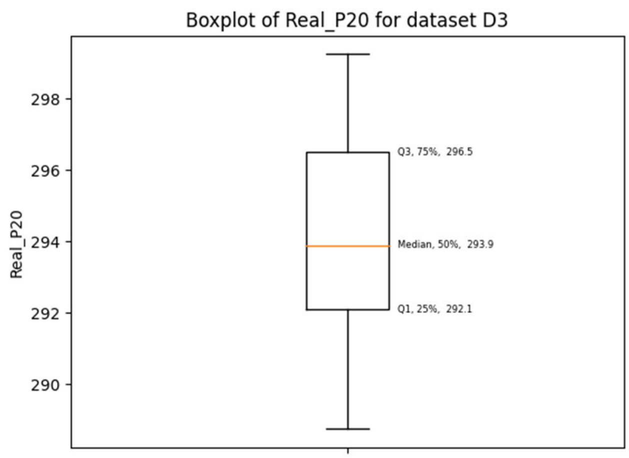

FIGURE 6. Boxplot of Real_P20 for dataset D3.

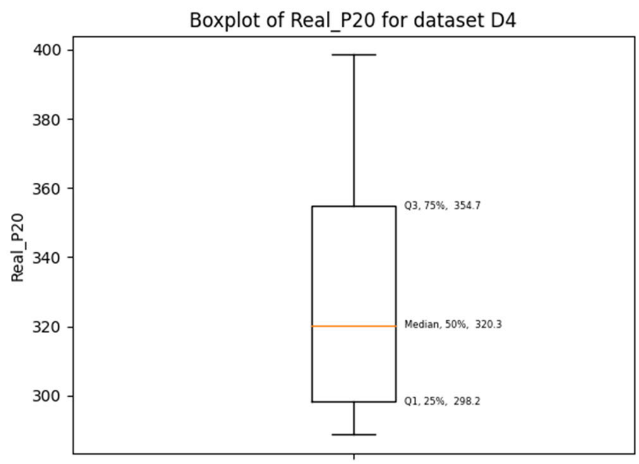

FIGURE 7. Boxplot of Real_P20 for dataset D4.

- Mean Balance Relative Error (MBRE)

$$MBRE = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{|(y_i - \hat{y}_i)|}{min(y_i - \hat{y}_i)} \tag{29}$$

- Mean Inverted Balance Relative Error (MIBRE)

$$MIBRE = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{|(y_i - \hat{y}_i)|}{max(y_i - \hat{y}_i)} \tag{30}$$

- Median of Magnitude of Relative Error (MdMRE)

$$MdMRE = median_i(\frac{|y_i - \hat{y}_i|}{y_i}) \tag{31}$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{32}$$

where n is the number of observations, y_i is the real known value, $\hat{y}_i$ is the predicted value, and ε 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)

$$SSE = \sum\nolimits_{i=1}^{n} \varepsilon_i^2 \tag{33}$$

- Percentage of Prediction within x% (PRED (x))

$$PRED\,(x) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \begin{cases} 1 & if \;\; \frac{|y_i - \hat{y}_i|}{y_i} \le x \\ 0 & otherwise \end{cases} \tag{34}$$

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,

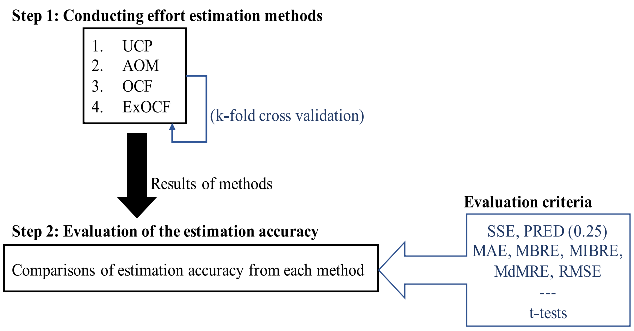

FIGURE 8. Experimental setup.

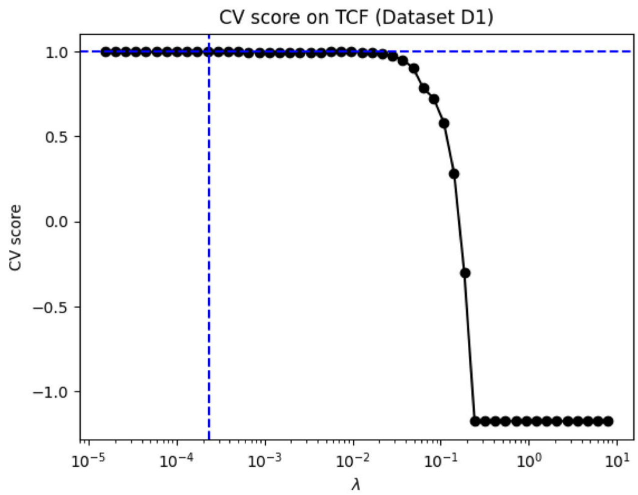

FIGURE 9. CV score on TCF (D1 dataset).

TABLE 10. The estimated tcf coefficients in the lasso regression.

	D1	D2	D3	D4
λ	0.000231	0.000268	0.000227	0.000236
intercept	0.690619	0.693400	0.720820	0.695850
T1	0.009451	0.009725	0.009547	0.009505
T2	-	-	-	-
T3	0.010897	0.010902	0.010311	0.010456
T4	0.009330	0.008877	0.009888	0.009556
T5	0.010430	0.011130	0.015199	0.010622
T6	0.009576	0.010157	-	0.009202
T7	0.008536	-	0.007298	0.008989
T8	-	-	-	-
T9	0.010551	0.014018	0.013144	0.010334
T10	0.010526	0.010893	0.009730	0.010902
T11	0.007387	0.006516	-	0.005998
T12	-	-	-	-
T13	-	-	-	-

MdMRE, and RMSE, as presented in (28-34). A pairwise t-test (at a 5% significance level) was also used to validate the accuracy of the methods.

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.

TABLE 11. The estimated ecf coefficients in the lasso regression.

	D1	D2	D3	D4
λ	0.000177	0.000192	0.000247	0.000327
intercept	1.373478	1.376197	1.404496	1.387716
ENV1	-	-	-	-
ENV2	-	-	-	-
ENV3	-0.032072	-0.042706	-0.032954	-0.033555
ENV4	-0.042291	-0.037886	-0.025558	-0.033001
ENV5	-0.029170	-0.028453	-0.029931	-0.029393
ENV6	-0.028133	-0.027549	-0.030139	-0.029072
ENV7	-0.027981	-0.026382	-0.029221	-0.028660
ENV8	-0.028193	-0.028713	-0.031169	-0.029333

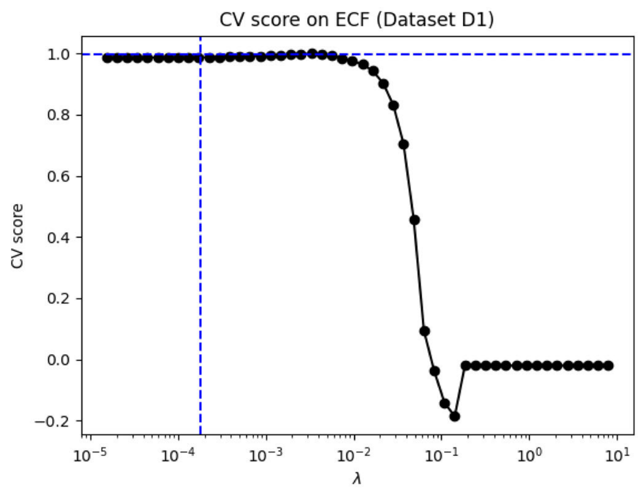

FIGURE 10. CV score on ECF (D1 dataset).

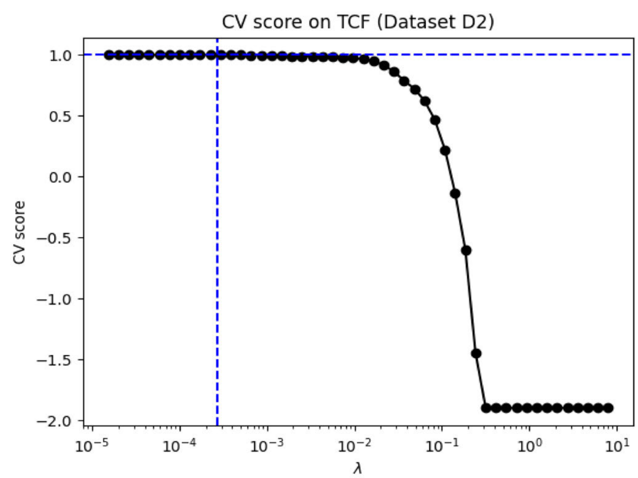

FIGURE 11. CV score on TCF (D2 dataset).

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

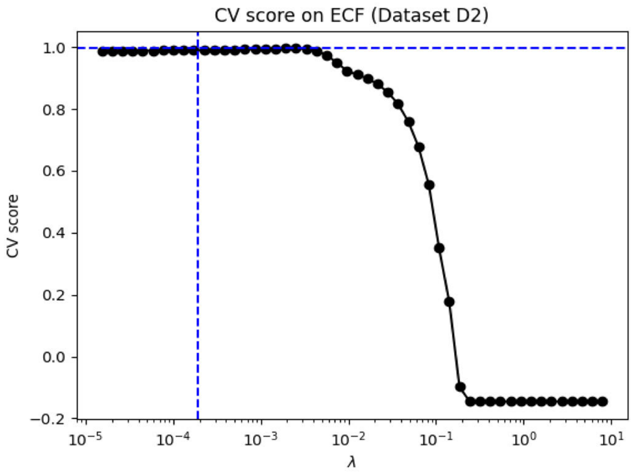

FIGURE 12. CV score on ECF (D2 dataset).

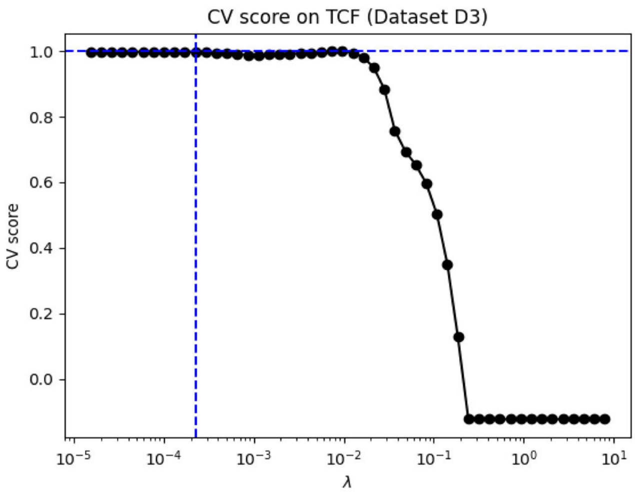

FIGURE 13. CV score on TCF (D3 dataset).

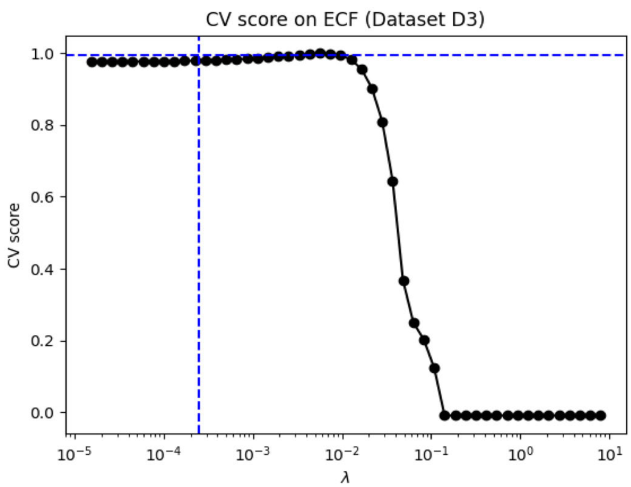

FIGURE 14. CV score on ECF (D3 dataset).

results. In contrast, high values for the PRED (0.25) show good results.

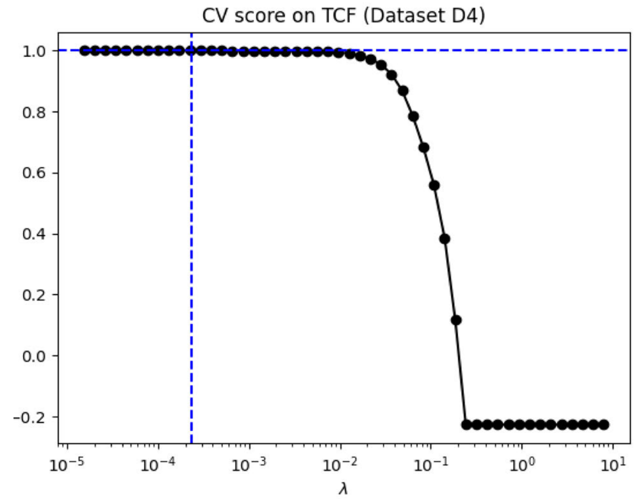

FIGURE 15. CV score on TCF (D4 dataset).

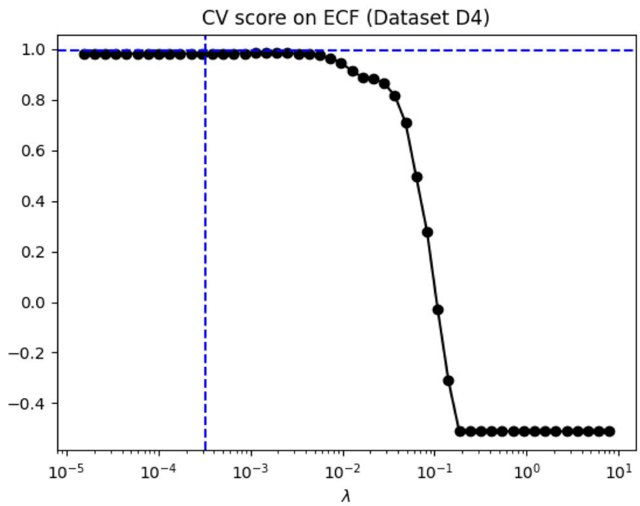

FIGURE 16. CV score on ECF (D4 dataset).

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 ≤ 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.

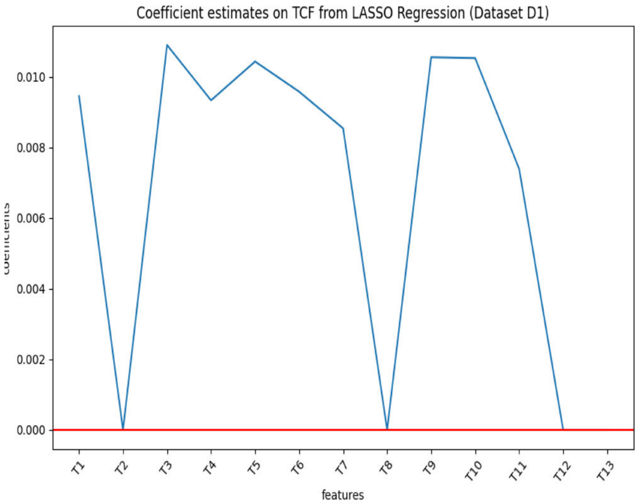

FIGURE 17. **Coefficient estimates on TCF from LASSO regression (D1 dataset).**

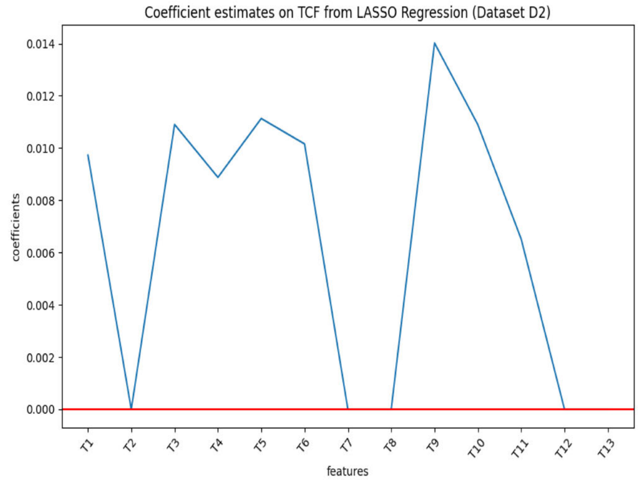

FIGURE 19. **Coefficient estimates on TCF from LASSO regression (D2 dataset).**

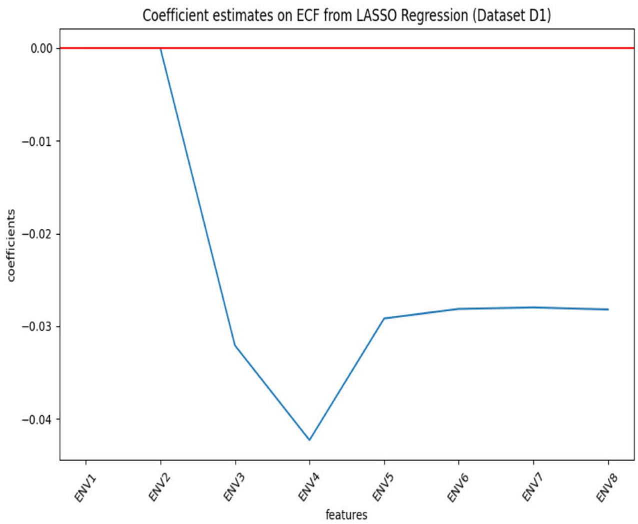

FIGURE 18. **Coefficient estimates on ECF from LASSO regression (D1 dataset).**

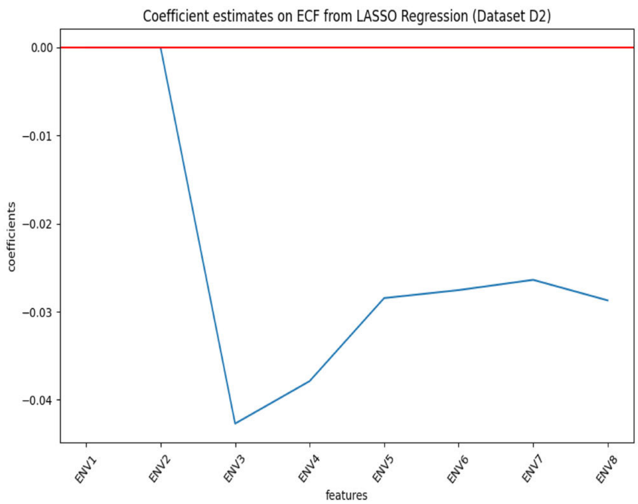

FIGURE 20. **Coefficient estimates on ECF from LASSO regression (D2 dataset).**

FIGURES 9-16 show a sequence of different R-squared values in proportion to different values of λ. The selected λ 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 λ values.

The details of the technical and environmental factors selected in each dataset with the determined λ, 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, T7, 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 $\lambda_{ECF} = 0.000327$ 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,

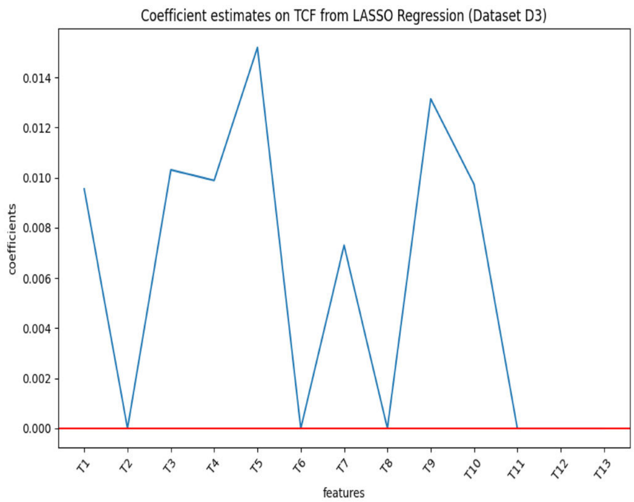

FIGURE 21. Coefficient estimates on TCF from LASSO regression (D3 dataset).

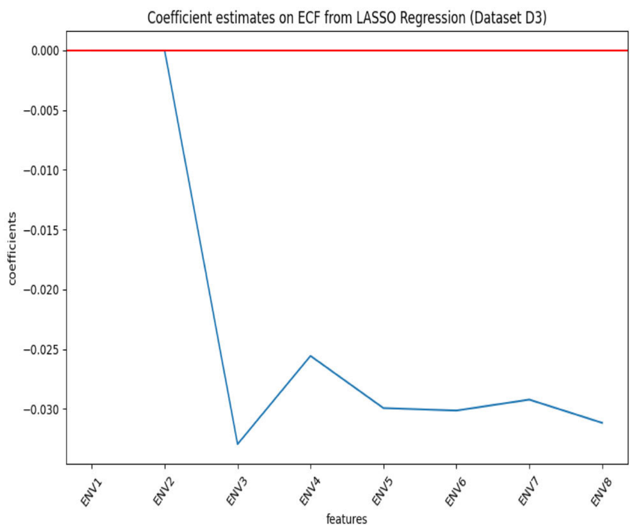

FIGURE 22. Coefficient estimates on ECF from LASSO regression (D3 dataset).

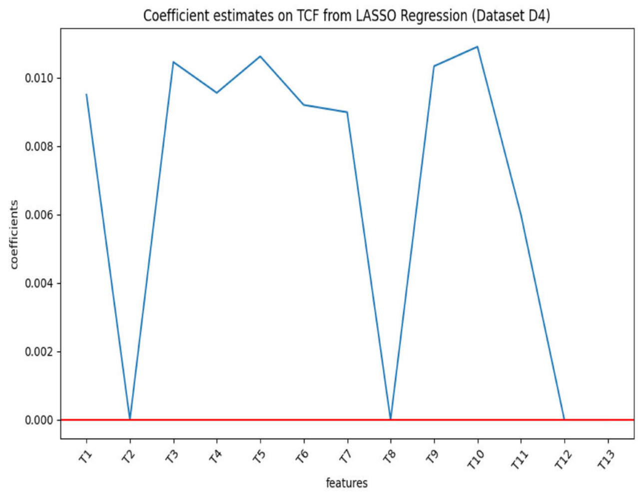

FIGURE 23. Coefficient estimates on TCF from LASSO regression (D4 dataset).

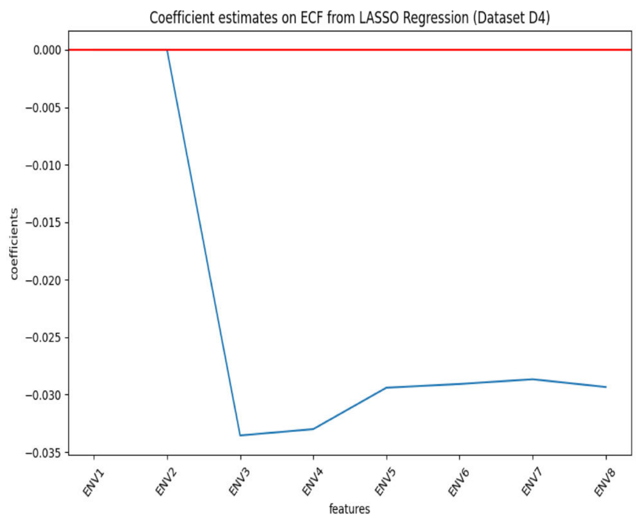

FIGURE 24. Coefficient estimates on ECF from LASSO regression (D4 dataset).

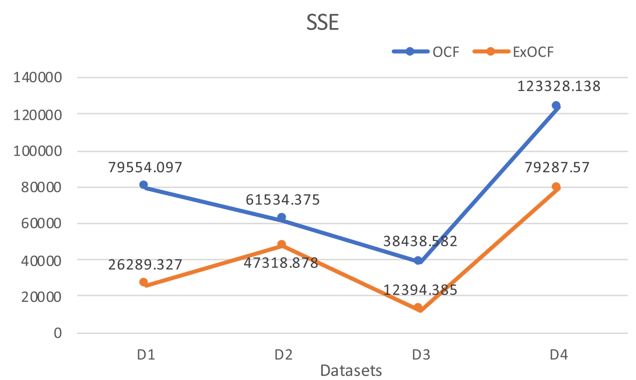

FIGURE 25. The SSE results of the OCF and ExOCF methods.

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

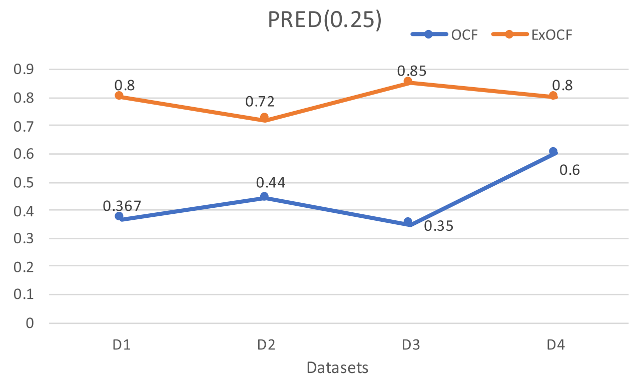

FIGURE 26. **The PRED (0.25) results of the OCF and ExOCF methods.**

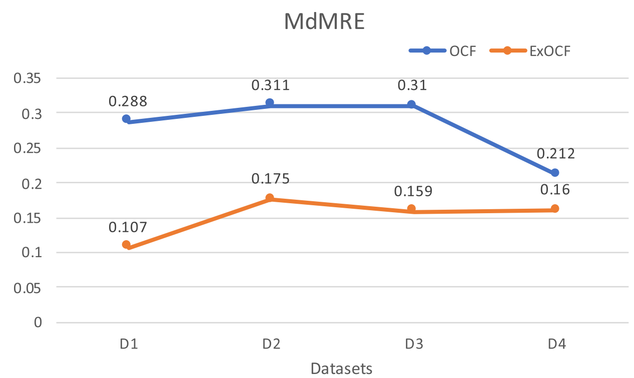

FIGURE 27. **The MdMRE results of the OCF and ExOCF methods.**

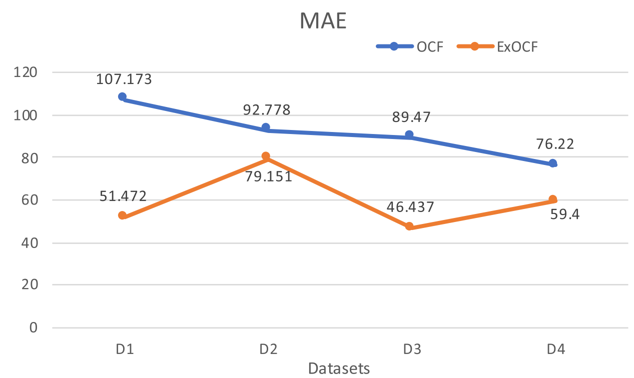

FIGURE 28. **The MAE results of the OCF and ExOCF methods.**

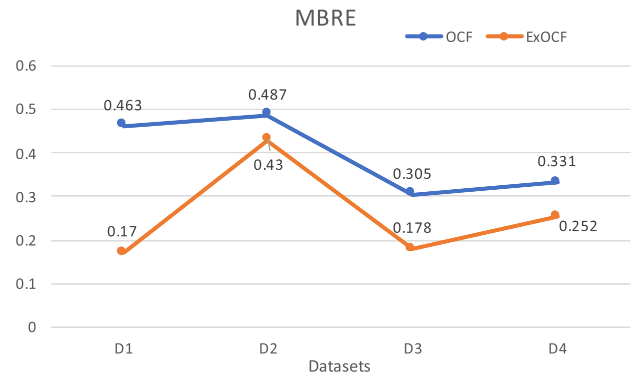

FIGURE 29. **The MBRE results of the OCF and ExOCF methods.**

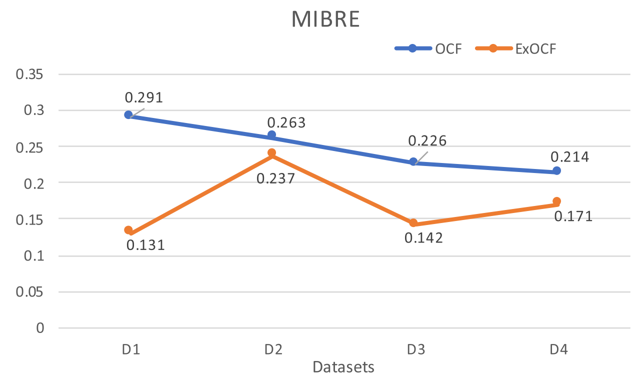

FIGURE 30. **The MIBRE results of the OCF and ExOCF methods.**

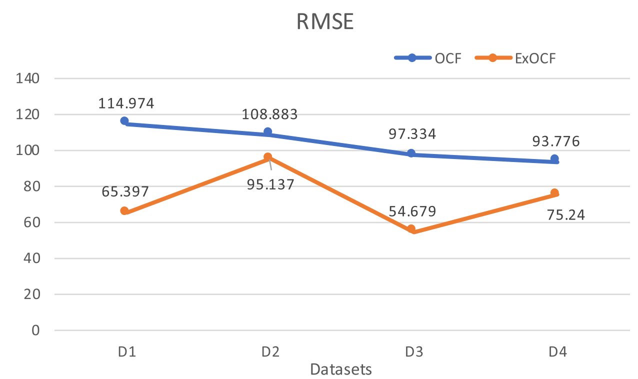

FIGURE 31. **The RMSE results of the OCF and ExOCF methods.**

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

TABLE 12. The results on the D1 dataset for each run.

D1 dataset	UCP	OCF	AOM	ExOCF
		Run #1		
SSE	89873.809	73732.493	30812.91	23577.641
PRED (0.25)	0.50	0.33	0.67	0.83
MdMRE	0.277	0.278	0.154	0.065
MAE	110.067	102.037	56.788	43.789
MBRE	0.540	0.428	0.205	0.152
MIBRE	0.295	0.272	0.152	0.113
RMSE	122.389	110.855	71.662	62.687
		Run #2		
SSE	97956.072	82891.999	39818.108	30362.475
PRED (0.25)	0.33	0.33	0.50	0.67
MdMRE	0.288	0.293	0.228	0.146
MAE	115.909	108.695	73.691	59.764
MBRE	0.572	0.466	0.252	0.194
MIBRE	0.311	0.290	0.190	0.150
RMSE	127.773	117.539	81.464	71.137
		Run #3		
SSE	95228.322	79219.801	45101.683	36052.935
PRED (0.25)	0.17	0.17	0.50	0.67
MdMRE	0.281	0.285	0.238	0.148
MAE	118.413	111.398	78.426	66.665
MBRE	0.578	0.463	0.262	0.200
MIBRE	0.320	0.298	0.198	0.160
RMSE	125.982	114.906	86.700	77.517
		Run #4		
SSE	83117.621	67977.319	21111.679	13414.242
PRED (0.25)	0.50	0.50	0.83	1
MdMRE	0.269	0.273	0.131	0.095
MAE	101.134	93.868	53.599	38.405
MBRE	0.525	0.418	0.167	0.111
MIBRE	0.282	0.262	0.138	0.095
RMSE	117.698	106.440	59.318	47.283
		Run #5		
SSE	111781.122	93948.873	38346.112	28039.34
PRED (0.25)	0.50	0.50	0.67	0.83
MdMRE	0.314	0.309	0.136	0.083
MAE	127.842	119.867	63.397	48.735
MBRE	0.658	0.541	0.257	0.192
MIBRE	0.356	0.333	0.177	0.134
RMSE	136.492	125.133	79.944	68.361
		Average of the five runs		
SSE	95591.389	79554.097	35038.098	26289.327
PRED (0.25)	0.40	0.37	0.63	0.80
MdMRE	0.286	0.288	0.177	0.107
MAE	114.673	107.173	65.18	51.472
MBRE	0.575	0.463	0.229	0.17
MIBRE	0.313	0.291	0.171	0.131
RMSE	126.067	114.974	75.818	65.397

TABLE 13. The results on the D2 dataset for each run.

D2 dataset	UCP	OCF	AOM	ExOCF
		Run #1		
SSE	180847.52	81712.875	178916.935	68413.804
PRED (0.25)	0.4	0.4	0.4	0.6
MdMRE	0.299	0.389	0.342	0.187
MAE	147.302	108.565	148.822	100.496
MBRE	0.706	0.663	0.683	0.604
MIBRE	0.334	0.320	0.334	0.307
RMSE	190.183	127.838	189.165	116.973
		Run #2		
SSE	31539.052	44382.268	48595.858	28216.834
PRED (0.25)	0.6	0.4	0.4	0.8
MdMRE	0.249	0.285	0.326	0.148
MAE	75.942	85.712	94.974	57.623
MBRE	0.309	0.372	0.439	0.275
MIBRE	0.228	0.243	0.292	0.175
RMSE	79.422	94.215	98.586	75.122
		Run #3		
SSE	144626.376	42913.083	143532.966	33889.05
PRED (0.25)	0.6	0.4	0.6	0.8
MdMRE	0.177	0.255	0.237	0.175
MAE	118.319	70.304	123.63	72.976
MBRE	0.402	0.311	0.425	0.322
MIBRE	0.242	0.194	0.258	0.218
RMSE	170.074	92.642	169.430	82.327
		Run #4		
SSE	189039.784	99138.916	172400.055	75108.318
PRED (0.25)	0.2	0.4	0.4	0.6
MdMRE	0.309	0.434	0.283	0.245
MAE	158.1	129.189	146.865	107.213
MBRE	0.763	0.752	0.708	0.662
MIBRE	0.368	0.353	0.344	0.309
RMSE	194.443	140.811	185.688	122.563
		Run #5		
SSE	146826.325	39524.734	149114.91	30966.383
PRED (0.25)	0.6	0.6	0.6	0.8
MdMRE	0.226	0.190	0.240	0.123
MAE	121.537	70.122	122.062	57.449
MBRE	0.411	0.337	0.403	0.288
MIBRE	0.247	0.204	0.241	0.173
RMSE	171.363	88.910	172.693	78.697
		Average of the five runs		
SSE	138575.811	61534.375	138512.145	47318.878
PRED (0.25)	0.48	0.44	0.48	0.72
MdMRE	0.252	0.311	0.285	0.175
MAE	124.24	92.778	127.271	79.151
MBRE	0.518	0.487	0.531	0.43
MIBRE	0.284	0.263	0.294	0.237
RMSE	161.097	108.883	163.112	95.137

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

TABLE 14. The results on the D3 dataset for each run.

D3 dataset	UCP	OCF	AOM	ExOCF
		Run #1		
SSE	66322.232	49961.858	12340.594	9092.464
PRED (0.25)	0	0.25	0.75	1
MdMRE	0.387	0.378	0.150	0.136
MAE	120.936	105.746	49.27	39.407
MBRE	0.411	0.358	0.176	0.153
MIBRE	0.283	0.258	0.145	0.124
RMSE	128.766	111.761	55.544	47.677
		Run #2		
SSE	124881.181	36364.975	42750.234	17426.662
PRED (0.25)	0	0.25	0.5	0.75
MdMRE	0.630	0.318	0.301	0.217
MAE	175.809	88.146	94.897	59.755
MBRE	0.602	0.301	0.325	0.224
MIBRE	0.375	0.224	0.237	0.177
RMSE	176.693	95.348	103.381	66.005
		Run #3		
SSE	99401.111	45149.046	35680.144	18590.572
PRED (0.25)	0	0.25	0.5	0.75
MdMRE	0.568	0.337	0.258	0.228
MAE	151.778	100.272	76.712	62.036
MBRE	0.518	0.341	0.262	0.228
MIBRE	0.334	0.248	0.190	0.180
RMSE	157.640	106.242	94.446	68.174
		Run #4		
SSE	59437.838	36555.809	22016.661	9746.603
PRED (0.25)	0.25	0.5	0.5	1
MdMRE	0.372	0.268	0.224	0.142
MAE	103.782	81.279	69.422	41.462
MBRE	0.354	0.278	0.245	0.162
MIBRE	0.241	0.204	0.193	0.131
RMSE	121.899	95.598	74.190	49.362
		Run #5		
SSE	48544.378	24161.22	17657.292	7115.625
PRED (0.25)	0.25	0.5	0.5	0.75
MdMRE	0.384	0.251	0.162	0.071
MAE	96.515	71.909	52.569	29.527
MBRE	0.329	0.248	0.179	0.123
MIBRE	0.232	0.194	0.141	0.097
RMSE	110.164	77.719	66.440	42.177
		Average of the five runs		
SSE	79717.348	38438.582	26088.985	12394.385
PRED (0.25)	0.1	0.35	0.55	0.85
MdMRE	0.468	0.31	0.219	0.159
MAE	129.764	89.47	68.574	46.437
MBRE	0.443	0.305	0.237	0.178
MIBRE	0.293	0.226	0.181	0.142
RMSE	139.032	97.334	78.8	54.679

TABLE 15. The SSE, MMRE, and PRED (0.25) results on the D4 dataset for each run.

D4 dataset	UCP	OCF	AOM	ExOCF
		Run #1		
SSE	163434.992	112155.553	114869.079	84466.505
PRED (0.25)	0.57	0.64	0.64	0.71
MdMRE	0.235	0.201	0.174	0.175
MAE	88.118	74.777	70.264	66.955
MBRE	0.406	0.310	0.3545	0.265
MIBRE	0.242	0.210	0.208	0.188
RMSE	108.045	89.504	90.581	77.674
		Run #2		
SSE	169380.711	141828.43	79939.216	75437.844
PRED (0.25)	0.5	0.57	0.64	0.79
MdMRE	0.263	0.205	0.201	0.141
MAE	88.477	77.884	62.463	58.902
MBRE	0.349	0.321	0.262	0.237
MIBRE	0.231	0.213	0.186	0.172
RMSE	109.993	100.650	75.564	73.405
		Run #3		
SSE	127945.384	126319.665	83476.391	80728.511
PRED (0.25)	0.57	0.57	0.57	0.86
MdMRE	0.217	0.210	0.217	0.183
MAE	81.964	76.524	65.636	61.61
MBRE	0.298	0.277	0.227	0.200
MIBRE	0.216	0.200	0.174	0.158
RMSE	95.598	94.989	77.218	75.936
		Run #4		
SSE	147307.497	115868.755	133700.708	76426.761
PRED (0.25)	0.57	0.71	0.50	0.79
MdMRE	0.234	0.191	0.240	0.149
MAE	85.1	76.203	84.383	55.856
MBRE	0.413	0.376	0.411	0.276
MIBRE	0.244	0.229	0.247	0.174
RMSE	102.577	90.974	97.724	73.885
		Run #5		
SSE	163804.311	120468.285	128906.951	79378.23
PRED (0.25)	0.43	0.50	0.50	0.86
MdMRE	0.285	0.254	0.250	0.149
MAE	100.049	75.714	80.831	53.679
MBRE	0.410	0.370	0.361	0.279
MIBRE	0.274	0.217	0.234	0.161
RMSE	108.168	92.762	95.956	75.299
		Average of the five runs		
SSE	154374.579	123328.138	108178.469	79287.57
PRED (0.25)	0.53	0.60	0.57	0.80
MdMRE	0.247	0.212	0.217	0.16
MAE	88.742	76.22	72.715	59.4
MBRE	0.376	0.331	0.323	0.252
MIBRE	0.242	0.214	0.21	0.171
RMSE	104.876	93.776	87.409	75.24

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 show the average p-value results and the SSE, PRED (0.25), MdMRE, MAE, MBRE, MIBRE, RMSE over five different runs and the final statistical conclusions. 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.

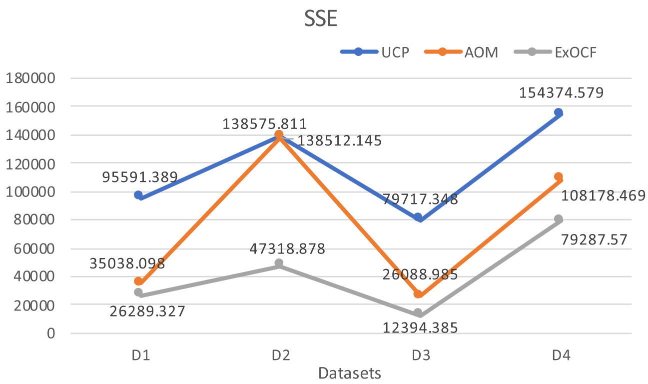

FIGURE 32. **The SSE results of the UCP, AOM and ExOCF methods.**

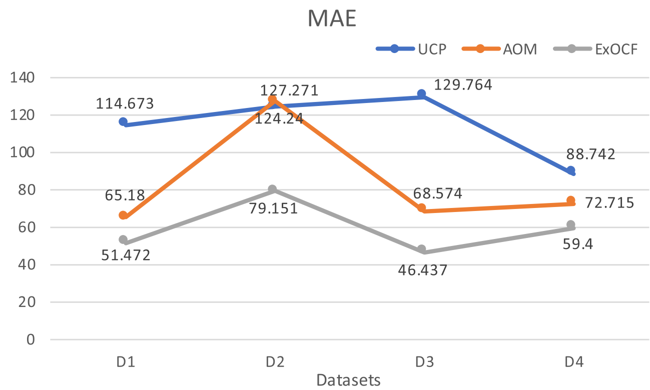

FIGURE 35. **The MAE results of the UCP, AOM and ExOCF methods.**

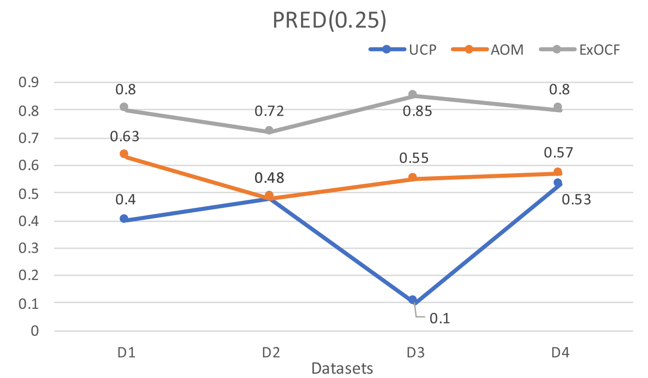

FIGURE 33. **The PRED (0.25) results of the UCP, AOM and ExOCF methods.**

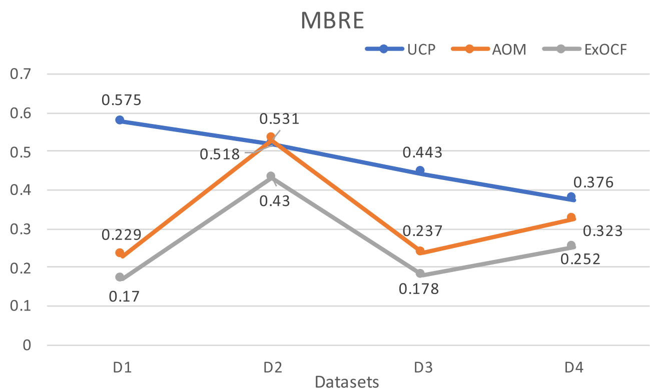

FIGURE 36. **The MBRE results of the UCP, AOM and ExOCF methods.**

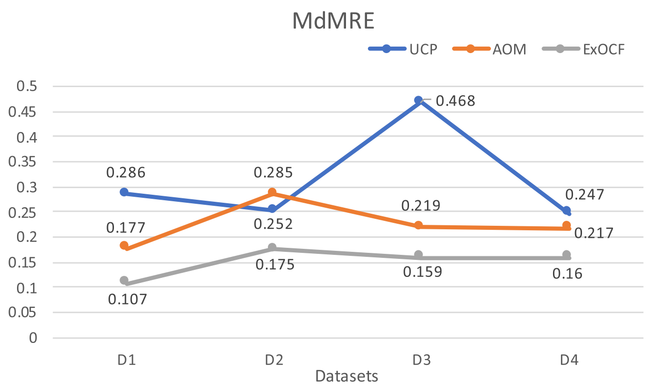

FIGURE 34. **The MdMRE results of the UCP, AOM and ExOCF methods.**

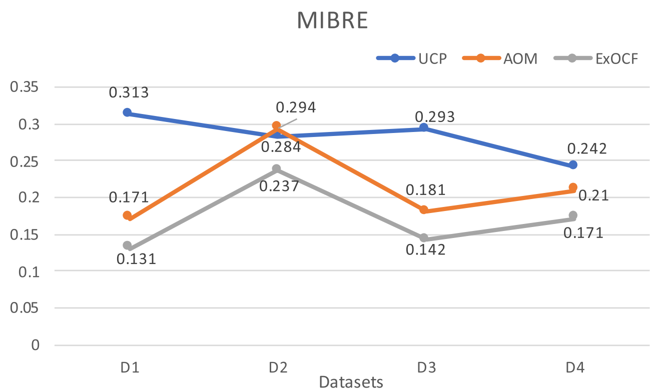

FIGURE 37. **The MIBRE results of the UCP, AOM and ExOCF methods.**

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 λ to extract a selected variable set, as shown in (16), before applying LASSO regression. In practice, the tuning parameter λ, which controls the strength of the penalty, has an important effect. In particular, if λ is sufficiently large, the coefficients must be exactly zero, leading to the dimensionality being reduced. The larger the parameter λ is, the greater the number of coefficients reduced to zero. Thus, we determined the λ 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 – is not known. This may influence data quality and comparability between data donors. In past publications, datasets used were preprocessed, which may also have an impact on reliability.

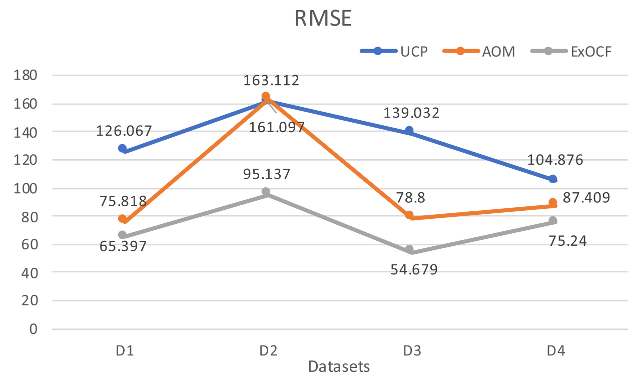

FIGURE 38. The RMSE results of the UCP, AOM and ExOCF methods.

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 vs. 95591.389	26289.327 vs. 79554.097	26289.327 vs. 35038.098
	Avg. p-value	0.00003	0.00006	0.00005
	Statistical conclusion	>>	>>	>>
PRED (0.25) results	Avg. PRED (0.25)	0.8 vs. 0.4	0.8 vs. 0.37	0.8 vs. 0.63
	Avg. p-value	0.0003	0.0002	0.00000
	Statistical conclusion	>>	>>	>>
MAE results	Avg. MAE	51.472 vs. 114.673	51.472 vs. 107.173	51.472 vs. 65.18
	Avg. p-value	0.00008	0.0001	0.00001
	Statistical conclusion	>>	>>	>>
MdMRE results	Avg. MdMRE	0.107 vs. 0.286	0.107 vs. 0.288	0.107 vs. 0.177
	Avg. p-value	0.0003	0.0002	0.001
	Statistical conclusion	>>	>>	>>
MBRE results	Avg. MBRE	0.170 vs. 0.575	0.170 vs. 0.463	0.170 vs. 0.229
	Avg. p-value	0.00000	0.00002	0.00000
	Statistical conclusion	>>	>>	>>
MIBRE results	Avg. MIBRE	0.131 vs. 0.313	0.131 vs. 0.291	0.131 vs. 0.171
	Avg. p-value	0.00004	0.00006	0.00000
	Statistical conclusion	>>	>>	>>
RMSE results	Avg. RMSE	65.397 vs. 126.067	65.397 vs. 114.974	65.397 vs. 75.818
	Avg. p-value	0.00005	0.0001	0.00003
	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

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

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. These

TABLE 18. The statistical t-test based on evaluation criteria (28-34) for the proposed method with each of the methods compared on the d3 dataset.

Pairs of methods		ExOCF vs. UCP	ExOCF vs. OCF	ExOCF vs. AOM
SSE results	Avg. SSE	12394.385 vs. 79717.348	12394.385 vs. 38438.582	12394.385 vs. 26088.985
	Avg. p-value	0.002	0.001	0.01
	Statistical conclusion	>>	>>	>>
PRED (0.25) results	Avg. PRED (0.25)	0.85 vs. 0.1	0.85 vs. 0.35	0.85 vs. 0.55
	Avg. p-value	0.0003	0.001	0.001
	Statistical conclusion	>>	>>	>>
MAE results	Avg. MAE	46.437 vs. 129.764	46.437 vs. 89.47	46.437 vs. 68.574
	Avg. p-value	0.0004	0.001	0.004
	Statistical conclusion	>>	>>	>>
MdMRE results	Avg. MdMRE	0.159 vs. 0.468	0.159 vs. 0.31	0.159 vs. 0.219
	Avg. p-value	0.0003	0.002	0.009
	Statistical conclusion	>>	>>	>>
MBRE results	Avg. MBRE	0.178 vs. 0.443	0.178 vs. 0.305	0.178 vs. 0.237
	Avg. p-value	0.0006	0.009	0.007
	Statistical conclusion	>>	>>	>>
MIBRE results	Avg. MIBRE	0.142 vs. 0.293	0.142 vs. 0.226	0.142 vs. 0.181
	Avg. p-value	0.0002	0.002	0.009
	Statistical conclusion	>>	>>	>>
RMSE results	Avg. RMSE	54.679 vs. 139.032	54.679 vs. 97.334	54.679 vs. 78.8
	Avg. p-value	0.0001	0.001	0.003
	Statistical conclusion	>>	>>	>>

TABLE 19. The statistical t-test based on evaluation criteria (28-34) for the proposed method with each of the methods compared on the d4 dataset.

Pairs of methods		ExOCF vs. UCP	ExOCF vs. OCF	ExOCF vs. AOM
SSE results	Avg. SSE	79287.57 vs. 154374.57	79287.57 vs. 123328.13	79287.57 vs. 108178.46
	Avg. p-value	0.0003	0.001	0.03
	Statistical conclusion	>>	>>	>>
PRED (0.25) results	Avg. PRED (0.25)	0.8 vs. 0.53	0.8 vs. 0.6	0.8 vs. 0.57
	Avg. p-value	0.002	0.01	0.006
	Statistical conclusion	>>	>>	>>
MAE results	Avg. MAE	59.4 vs. 88.742	59.4 vs. 76.22	59.4 vs. 72.715
	Avg. p-value	0.001	0.001	0.04
	Statistical conclusion	>>	>>	>>
MdMRE results	Avg. MdMRE	0.16 vs. 0.247	0.16 vs. 0.247	0.16 vs. 0.247
	Avg. p-value	0.005	0.01	0.01
	Statistical conclusion	>>	>>	>>
MBRE results	Avg. MBRE	0.252 vs. 0.376	0.252 vs. 0.331	0.252 vs. 0.323
	Avg. p-value	0.00005	0.0005	0.01
	Statistical conclusion	>>	>>	>>
MIBRE results	Avg. MIBRE	0.171 vs. 0.242	0.171 vs. 0.242	0.171 vs. 0.242
	Avg. p-value	0.001	0.001	0.02
	Statistical conclusion	>>	>>	>>
RMSE results	Avg. RMSE	75.24 vs. 104.876	75.24 vs. 93.776	75.24 vs. 87.409
	Avg. p-value	0.0002	0.0008	0.02
	Statistical conclusion	>>	>>	>>

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, MIBRE, 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 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 is approximately affected. One of our future works is to calibrate the weighting values of the correction factors to address the latest trend in the software engineering industry and improve the accuracy of the ExOCF method. Therefore, an approach to calibrate the weights of the correction factors using an artificial neural network [40] in the ExOCF model 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:1018991717352.

[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. TR_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 estimating: 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: 10.1023/B:EMSE.0000048321.46871.2e.

[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 Couting Practices Manual (FPCPM)*, 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: 10.1109/32.44364.

[19] G. Karner, "Resource estimation for objector projects," *Objective Syst. SF AB*, vol. 17, pp. 1–9, Sep. 1993.

[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. 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.

[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," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 3, pp. 65–71, 2015, doi: 10.14569/IJACSA.2015.060310.

[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. Sholiq, R. S. Dewi, and A. P. Subriadi, "A comparative study of software development size estimation method: UCPabc vs function points," in *Proc. 4th Inf. Syst. Int. Conf.*, vol. 124, Jan. 2017, pp. 470–477. [Online]. Available: https://www.researchgate.net/profile/Sholiq-Sholiq

[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. (COMPSAC)*, 2006, pp. 221–228, doi: 10.1109/COMPSAC.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. Orosco, "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/ESEM.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: 10.1145/1868328.1868338.

[38] H. L. T. K. Nhung, 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. 6, pp. 425–437, Jan. 1997.

[41] R. P. Pantoni, E. A. Mossin, and D. Brandao, "Task effort fuzzy estimation for software development," *INFOCOMP J. Comput. Sci.*, vol. 7, pp. 84–89, Jun. 2008.

[42] M. R. Braz and S. R. Vergilio, "Using fuzzy theory for effort estimation of object-oriented software," in *Proc. 16th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2004, pp. 196–201, doi: 10.1109/ICTAI.2004.119.

[43] R. Alves, P. Valente, and N. J. Nunes, "Improving software effort estimation with human-centric models: A comparison of UCP and iUCP accuracy," in *Proc. 5th ACM SIGCHI Symp. Eng. Interact. Comput. Syst. (EICS)*, 2013, pp. 287–296, doi: 10.1145/2494603.2480300.

[44] P. Jovan, P. Sofija, B. Marko, and B. Dragan, "Enhancing use case point estimation method using fuzzy algorithms," in *Proc. 23rd Telecommun. Forum Telfor (TELFOR)*, Nov. 2015, pp. 886–889, doi: 10.1109/TELFOR.2015.7377607.

[45] M. Saroha and S. Sahu, "Software effort estimation using enhanced use case point model," in *Proc. Int. Conf. Comput., Commun. Autom.*, May 2015, pp. 779–784, doi: 10.1109/CCAA.2015.7148515.

[46] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *J. Syst. Softw.*, vol. 86, no. 1, pp. 144–160, Jan. 2013.

[47] A. B. Nassif, L. F. Capretz, and D. Ho, "Enhancing use case points estimation method using soft computing techniques," *J. Global Res. Comput. Sci.*, vol. 1, no. 4, p. 12, Dec. 2010.

[48] 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.

[49] A. B. Nassif, D. Ho, and L. F. Capretz, "Regression model for software effort estimation based on the use case point method," in *Proc. Int. Conf. Comput. Softw. Modeling (IPCSIT)*, vol. 14, 2011, pp. 106–110.

[50] M. Jørgensen, "Regression models of software development effort estimation accuracy and bias," *Empirical Softw. Eng.*, vol. 9, no. 4, pp. 297–314, Dec. 2004.

[51] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on use case points," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 200–213, Mar. 2011.

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

[53] A. B. Nassif, L. F. Capretz, and D. Ho, "Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model," in *Proc. 13th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Aug. 2012, pp. 589–594.

[54] M. S. Iraji and H. Motameni, "Object oriented software effort estimate with adaptive neuro fuzzy use case size point (ANFUSP)," *Int. J. Intell. Syst. Appl.*, vol. 4, no. 6, pp. 14–24, Jun. 2012.

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

[56] N.-H. Chiu and S.-J. Huang, "The adjusted analogy-based software effort estimation based on similarity distances," *J. Syst. Softw.*, vol. 80, no. 4, pp. 628–640, Apr. 2007, doi: 10.1016/j.jss.2006.06.006.

[57] V. Fonti, "Feature selection using LASSO," Res. Paper Bus. Anal., Vrije Univ. Amsterdam, Amsterdam, The Netherlands, Tech. Rep., 2017.

[58] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ICACA)*, Oct. 2016, pp. 18–20, doi: 10.1109/ICACA.2016.7887916.

[59] M. Azzeh and A. B. Nassif, "A hybrid model for estimating software project effort from use case points," *Appl. Soft Comput.*, vol. 49, pp. 981–989, Dec. 2016, doi: 10.1016/j.asoc.2016.05.008.

[60] 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, doi: 10.1016/j.infsof.2017.12.009.

[61] 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.

[62] 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.

[63] M. Azzeh and A. B. Nassif, "Project productivity evaluation in early software effort estimation," *J. Softw., Evol. Process*, vol. 30, no. 12, p. e2110, Dec. 2018.

[64] 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.

[65] M. Azzeh, 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.

[66] 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.

[67] 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.

[68] 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.

[69] S. Bagheri and A. Shameli-Sendi, "Software project estimation using improved use case point," in *Proc. IEEE 16th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, Jun. 2018, pp. 143–150, doi: 10.1109/SERA.2018.8477225.

[70] 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.

[71] H. T. Hoc, V. Van Hai, and H. L. T. K. Nhung, "Adamoptimizer 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.

[72] 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.

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

[74] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–17, Feb. 2019.

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

[76] 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/stnd.2006.24.3.351.

[77] 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

[78] 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.

[79] S. G. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.

[80] 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. Int. Conf. Softw. Eng.*, May 1999, pp. 313–323.

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

[82] 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.

[83] A. Idri, I. Abnane, and A. Abran, "Evaluating Pred(p) and standardized accuracy criteria in software development effort estimation," *J. Softw., Evol. Process*, vol. 30, no. 4, p. e1925, Apr. 2018.

[84] D. R. Anderson, D. J. Sweeney, and T. A. William, *Statistics for business and Economics*. Mason, OH, USA: Thomson South-Western, 2009.

[85] A. Ross and V. L. Willson, "Paired sample T-test," in *Basic and Advanced Statistical Tests*. Rotterdam, The Netherlands: Sense Publishers, 2017, doi: 10.1007/978-94-6351-086-8_4.

[86] L. N. H. Nam and H. B. Quoc, "The hybrid filter feature selection methods for improving high-dimensional text categorization," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 25, no. 2, pp. 235–265, Apr. 2017.

[87] L. N. H. Nam and H. B. Quoc, "The ranking methods in the filter feature selection process for text categorization system," in *Proc. PACIS*, vol. 159, 2016.

[88] W. Y. Hsu, C. Y. Lin, and W. F. Kuo, "Unsupervised fuzzy c-means clustering for motor imagery EEG recognition," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 8, pp. 4965–4976, Aug. 2011.

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 Zlín, Czech Republic.

From 2010 to 2018, she was a Lecturer with the Department of Information Systems, Faculty of Information Systems, University of Science (HCMUS). Her research interests include database management systems, software engineering, and software effort estimation methods based on use case points.

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 engineering informatics with the Department of Computer and Communication Systems, Faculty of Applied Informatics, Tomas Bata University in Zlín, 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 Vsetiìn, 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 Vsetiìn, 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.