Optimisation of Software Effort Estimation by Improving Functional Points Analysis

Vo Van Hai, Ph.D.

Doctoral Thesis Summary

Tomas Bata University in Zlín

Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis

Optimalizace odhadu softwarového úsilí zlepšením analýzy funkčních bodů

Optimisation of Software Effort Estimation by Improving Functional Points Analysis

Author:	**Vo Van Hai, Ph.D.**
Degree programme:	Engineering Informatics / P3902
Degree course:	Software Engineering / 3902V023
Supervisor:	Assoc. Prof. Ing. Zdenka Prokopová, CSc.
Consulting Supervisor:	Assoc. Prof. Ing. Radek Šilhavý, Ph.D.
External examiners:	Prof. RNDr. Ing. Miloš Šeda, Ph.D.
	Assoc. Prof. Oldřich Trenz, Ph.D.
	Assoc. Prof. Ing. Bronislav Chramcov, Ph.D

Zlín, November 2022

Published by **Tomas Bata University in Zlín** in the Edition **Doctoral Thesis Summary.**

The publication was issued in the year 2022.

Key words in Czech: *odhad úsilí vývoje softwaru, analýza funkčních bodů, váhy funkční složitosti, kategoriální proměnné, klastrování dat, strojové učení*

Key words: *software effort estimation, function point analysis, calibration complexity weight, categorical variables, data clustering, machine learning*

Full text of the doctoral thesis is available in the Library of TBU in Zlín.

ISBN 978-80-7678-129-0

ABSTRAKT

Odhad softwarového úsilí je jedním ze základních úkolů v procesu projektového řízení. Tento výsledek z určování softwarového úsilí lze využít k různým účelům, jako je např. stanovení nákladů, plánování projektu nebo jeho použití pro účely nabídkového řízení projektu a mnoho dalších souvisejících aspektů. Existuje mnoho předchozích studií souvisejících s problémy odhadu úsilí, jako je zavádění nových modelů nebo vylepšení předchozích modelů za účelem zlepšení přesnosti procesu odhadu.

Od softwarové krize [1] bylo věnováno mnoho úsilí návrhu modelů vývoje softwaru, aby se zajistilo, že se tento proces stane strukturovanějším a udržitelnějším. V 80. letech byla vyvinuta metoda Function Points Analysis (FPA), která se stala jednou z kritických metod při odhadu velikosti softwaru. Postupem času však původní hodnoty této metody zastaraly a je třeba je aktualizovat.

S rychlým rozvojem oblasti umělé inteligence se techniky strojového učení staly základní složkou mnoha činností vývoje softwaru, aby uspokojily rostoucí potřeby lidí. Tyto techniky strojového učení nabízejí lepší budoucnost zejména pro informatiku a aspekty společnosti obecně. Skutečnost, že se staly také nástrojem pro predikci softwaru je v souladu s vývojovým trendem.

Předkládaná disertační práce navrhuje novou metodu využívající techniky strojového učení k návrhu nového systému pro kalibraci složitosti vah používaných v metodě FPA a zavádí nový rámec pro optimalizaci výsledků odhadu úsilí. Pro realizaci tohoto návrhu je nezbytný výběr vhodného algoritmu strojového učení. Kromě toho se také ukázalo, že velký vliv na výsledky odhadů má klastrování dat. Proto je navržený způsob klastrování dat, který je aplikován podle vhodného klastrovacího kritéria.

Výsledky získané v této práci byly hodnoceny podle některých nezaujatých hodnotících kritérií a dosáhly mnohem kvalitnějšího výsledku než původní metoda FPA.

Klíčová slova: *odhad úsilí vývoje softwaru, analýza funkčních bodů, váhy funkční složitosti, kategoriální proměnné, klastrování dat, strojové učení*

ABSTRACT

Estimating software effort is one of the essential tasks in the project management process. Various purposes can use this result from determining the software effort, such as determining the cost, planning the project, or using it for project bidding purposes and many other related aspects. There have been a lot of previous studies related to effort estimation problems, such as introducing new models or improving previous models to improve the accuracy of the estimation process.

Since the software crisis [1], efforts have been made to propose software development models to ensure this process becomes more structured and sustainable. In the 1980s, the Function Points Analysis (FPA) method was proposed and has become one of the critical models in software size estimation. However, over time, the original values of this method have become obsolete and need to be updated.

With the rapid development of the field of artificial intelligence, typically, machine learning techniques have become a core component for many software development activities to meet human's growing needs. These machine learning techniques offer a brighter future for computer science in particular and aspects of society in general. The fact that it has become a tool for software prediction is not out of the development trend.

This thesis proposes a new method using machine learning techniques to present a new calibration complexity weight system applied in the FPA method and introduces an optimization framework for optimizing the effort estimation results. To implement this proposal, choosing a suitable machine learning algorithm is indispensable. In addition, data clustering has also proven that has a great influence on the estimation results. Therefore, the proposed method is also applied according to a particularly suitable clustering criterion.

The results obtained in this study were evaluated against some unbiased evaluation criteria and achieved a much better result than the original FPA method.

Keywords: Software Effort Estimation, Function Point Analysis, Data Clustering, Categorical Variables, Machine Learning, Calibration Complexity Weight.

Contents Of the Thesis

Abstrakt ... i
Abstract ... ii
Contents Of the Thesis ... iii
1. Current state of effort estimation ... 1
2. Objectives of the Thesis ... 5
3. Proposed method ... 5
 3.1 Applying machine learning algorithms in effort estimation ... 6
 3.2 Applying segmentation techniques in effort estimation ... 8
  3.2.1 Using categorical variables ... 8
  3.2.2 Using segmentation algorithms ... 10
 3.3 Propose a New Calibration System and Optimization Framework ... 12
4. Main Results ... 13
 4.1 Finding the best suitable Machine Learning algorithm ... 14
  4.1.1 On the entire non-clustered dataset ... 14
  4.1.2 On the clustered dataset ... 15
  4.1.3 Summary ... 15
 4.2 Finding the best suitable clustering criterion ... 16
  4.2.1 With categorical variables ... 16
  4.2.2 With segmentation algorithms ... 16
  4.2.3 Summary ... 20
 4.3 New Calibration System and Optimization Framework ... 21
  4.3.1 Summary ... 25
5. Contribution of the thesis to science and practice ... 25
6. Conclusion and Future work ... 26
7. References ... 28
LIST OF PUBLICATIONS ... 30
Curriculum Vitae ... 33

1. CURRENT STATE OF EFFORT ESTIMATION

Software estimation techniques today involve three approaches: non-algorithmic, algorithmic, and machine learning [2], [3]. Each approach has its advantages and optimal domain of application. In this study, an algorithmic approach (in particular, IFPUG FPA) is used as the fundamental technique in which the size estimation stage is customized based on machine learning techniques to calibrate the system complexity weight. This section will examine the state of the current effort estimation studies. We will look at the workaround of these approaches.

The FPA method has made certain contributions to the software industry. Albrecht [4] first introduced FPA in 1979 and presented the Functional Point (FP) metric to measure the functionality of a project. It was proposed in response to a number of problems with other system size measures, such as lines of codes. Effective software development and maintenance management with FPA was advocated and made more widely known in 1986 by the International Function Point User Group (IFPUG) [5]

However, the FPA method has encountered some disputes from different researchers - in terms of advantages and limitations. Steven et al. [6] studied FPA from a manager´s and developer´s perspective-based on 13 attributes with three key findings: SLOC count is less complicated than FP; developers are better able to comprehend the benefits of FP than managers; the difference between managers and developers is in the Values Block Communication necessary to propound informed decisions. Some studies reported that the FPA method does not create consistent results when applied to different metrics [7], [8]. Meli [9] pointed out a mismatch between the complexities established for the Base Functional Components and the possible productivity estimates.

Many studies showed that the FPA scored the BFC incorrectly. For example, the same data function and/or the same transactional function with different combinations of DET and RET/FTR can be categorized with the same complexity. This leads to the same number of function points for the data functions and/or transactional functions. They also notice that - in some situations, functionalities that have very similar DET and RET / FTR can be categorized with different complexities; and thus, receive different FPA weightings.

Xia et al. - [10] doubted that the Unadjusted Function Points weight values, which were raised based on a study of the IBM data processing systems - (locally), could not reflect the software globally. In [11], they continually point out the existence of ambiguous classification, and the original method may not fully reflect the reality of the software complexity under the specific software application. In [12], they proved that there is no clear boundary between two classifications in FP counting. To resolve these problems, the authors suggested the merging of three techniques - (Fuzzy Logic, Artificial Neural Networks, and Statistical Regression) in a neuro-fuzzy function point calibration model.

Ahmed et al. [13] showed that many factors could be affected by the complexity of FP weight metrics values, like methodologies to develop software, support tools, and other factors. The authors proposed that new FPA weights were measured based on an adapted genetic algorithm. The proposed algorithm is based on a set of initial solutions - using biologically inspired evolution mechanisms to derive new - and possibly better, solutions.

According to Hajri et al. [14], the classification of function types into simple, average, and complex does not reflect the entire complexity necessary to develop user systems. The main improvement idea of this research is to establish a new weighting system for FP measurement using Artificial Neural Networks (ANN), that is to say, (the back-propagation technique). In the first step, they use the original weights system as baselines in order to establish the new weights. Next, they train one of the most popular Neural Networks techniques to predict the values of the new weights. And then, they apply the new weights and the original weights in the FP model. Finally, they calculate the FP count, depending on the original and new weights.

Ya-Fang et al. [15] studied that the BFCs weights - which were set by IFPUG, are said to reflect the functional size of the software, but actually - today's software differs drastically from the past; so it is no longer suitable. Authors also discovered that this inconsistency in a large number of BFCs, which lies on the specified intervals´ boundary areas, becomes even worse. The cause is due to the inaccurate classification of various system functionalities – which would distort its functional size.

The Function Points Measurement Process is not accurate in some specific cases, as demonstrated by Rao and Raju [16], and the number of referenced items, which establishes the lower limits of the high-complexity range, can lead to the same measurement accuracy issues, particularly in systems that reference a variety of data element types (DETs).

To learn about various machine learning (ML) strategies, their estimate accuracy, and the comparison between multiple models and estimation contexts, Wen et al. [17] looked through 84 original studies of ML techniques in SEE. This study discovered that eight different ML techniques have been used in SEE and concluded that ML models offer more accurate estimates than non-ML models. Case-Based Reasoning, Artificial Neural Networks, Decision Trees, Bayesian Networks, Support Vector Regression, Genetic Algorithms, Genetic Programming, and Association Rules are the eight ML subtypes mentioned above. They also discovered that DT, ANN, and CBR are utilized the most frequently.

Phannachitta [18] analyzed 13 different datasets using 14 machine learning algorithms frequently utilized in data science. According to the results, two algorithms, random forest and bagging, outperform the other algorithms. Additionally, the author suggests merging algorithms to get better outcomes. The author revisits another study's comparison of software effort adaptors based on heuristics and machine learning methods [19]. The authors integrated the seven separate methods. Ordinary least squares regression, classification and regression trees, SVR, ANN, deep random forests, and Gradient boosting machines are the algorithms employed in this work. The findings of this research suggest that a combination model is required to get more accurate estimates. The study's top performer was the analogy-based model, which adjusts to the effort by combining the Gradient boosting machine algorithm and a conventional adaptation method based on productivity adjustment.

S. Shukla and S. Khumar [20] use LR, SVM, KNN, and ANN algorithms to find a more feasible model for estimating software effort. The authors prove experimentally that the ANN algorithm is the best in this case. In another study [21], the authors used ANN with its ensembles (Ridge-MLPNN, Lasso-MLPNN, Bagging-MLPNN, and AdaBoost-MLPNN) to improve the software performance estimation process. The result signified that this model improves the performance compared to just ANN, and the combination of AdaBoost-MLPNN produced the highest accuracy. Priya Varshini et al. [22] used the ensemble technique to find the best suitable method with the same idea about using ensemble approaches for software estimation. Ensemble techniques studied for assessment were averaging, weighted averaging, stacking, boosting, and bagging. Single models considered for comparison were SVM, decision tree, random forest, neural net, ridge, LASSO, elastic net, and deep net algorithms. The proposed stacking using random forest provided the best result. This result was compared with the singles model and also got outperformed.

Additionally, Hammad et al. [23] examined four different algorithms (MLR, SVM, ANN, and K-Star) in estimating the real effort from the software features at the early phases of the software development life cycle to find a good ML technique for predicting software effort. The outcomes demonstrate the viability of using ML for software effort estimation. The results produced by SVM are the best of the four suggested algorithms.

Another issue related to this study is the application of data clustering techniques to improve the accuracy of the software effort estimation process.

Using parametric models with a mathematical foundation has some drawbacks, as demonstrated by Aroba et al. [24]. These restrictions can be overcome by combining segmentation models with the participation of other models to produce one model. Due to the fuzziness, it is crucial to take into account the fact that a project only fits within one segment. In order to estimate software cost, a segmentation model based on fuzzy logic is proposed. According to experimental findings, accuracy has dramatically increased.

Using three categorical variables—relative size, industrial sector, and organization type area - P. Silhavy et al. [25] created a novel categorical variable segmentation model based on dataset segmentation. The category variable of relative size serves as the segmentation parameter for the suggested approach. The proposed approach beats the IFPUG FPA model, spectral clustering-based models, and regression models in terms of estimation accuracy.

A soft computing approach to estimating software effort was suggested by Azath et al.[26] . The dataset clustered by the fuzzy-c-means clustering algorithm will be used to produce the rules. These learned rules will serve as the input for another neural network-based operation. This study's neural network model is an amalgam of optimization algorithms. This optimization algorithm will use the algorithms Artificial Bee Colony (ABC), Modified Cuckoo Search (MCS), and hybrid ABC-MCS. The experiment made use of the NASA 60, NASA 90, and Desharnais databases. The outcomes obtained using this suggested paradigm are excellent.

Prokopova et al. [27] analyze three different distance metrics using k-means, hierarchical, and density-based clustering techniques. The outcomes emphasize the significance of choosing the proper clustering type and distance metric. The authors demonstrate that hierarchical clustering results in inaccurate cluster distributions and can thus not be used. K-means clustering appears to be the segmentation technique that performs the best.

According to Bardisiri et al. [28], clustering as a method of dataset segmentation significantly impacts the accuracy of development effort estimation since it enables the removal of insignificant projects from historical data points. In that study, the authors put up a hybrid model that incorporates fuzzy clustering, artificial neural networks, and analogy-based estimation. The test made use of the Desharnais and Maxwell datasets. The experimental findings show promise, reaching up to 127% for the PRED (0.25) evaluation criterion.

Benala et al. [29] combined functional link artificial neural networks with unsupervised learning approaches in a study to forecast the software effort (clustering algorithms). To thoroughly examine the performance, the Functional Link Artificial Neural Networks (FLANNs) technique was employed in this instance. Chebyshev polynomials were chosen as the functional expansion method. The empirical evaluation of this proposed method took into account three real-world datasets related to software cost estimation. The experimental results demonstrate that the proposed method performs well for software cost estimates and can significantly increase the prediction accuracy of standard FLANN.

The above are some of the studies related to this study. In addition, there are many other studies with proposed solutions to increase the accuracy of the estimation process. However, with certain limitations, this study cannot fully address these contributions.

2. OBJECTIVES OF THE THESIS

With the aim of increasing the accuracy in software effort estimation, this study uses the IFPUG FPA method combined with machine learning techniques to allow effort estimation on groups segmented by Industry Sector categorical variable.

It is necessary to determine the appropriate clustering criteria and algorithm to solve this target. With the determination of the proper clustering criteria, the clustering algorithms and categorical variables are assessed to determine the most suitable clustering criteria. With the determination of the most suitable algorithm, a survey was conducted to identify some of the most commonly used algorithms recently and then performed an evaluation on these algorithms to find the most appropriate algorithm. In addition, the results of the FPA counting process will also be optimized again to improve the accuracy of the final estimate.

Hence, with the above problems, the aim of this thesis is:

1) A calibration complexity weight system for the IFPUG FPA method is proposed.
2) Create an effort estimation optimization framework based on regression models, machine learning, clustering, and the use of categorical variables.
3) Based on experiments, compare the proposed approaches with the reference method FPA and the tested approaches with each other.
4) Assess the contribution of the proposed optimization procedures to the refinement of the FPA method used for software estimation.

3. PROPOSED METHOD

The essential idea of this study is based on the combination of the IFPUG FPA method and the machine learning techniques. The FPA takes the basement role, and the machine learning techniques play an inference role. First, two following phases should be done as the premise for the whole process: 1) find the best suitable machine learning algorithm and 2) find the best suitable clustering criterion. After selecting the best suitable algorithm and clustering criterion, the calibration phase calibrates and proposes the functional complexity weight system.

The new project that needs the effort estimation uses the FPA for counting function points with the default complexity weight (of the FPA method) was replaced by the new complexity weight system. This phase's result (effort) will be optimized using the effort optimization framework.

The effort estimation framework uses the voting ensemble model with four base estimators (Random Forest Regressor, Bayesian Ridge Regressor, MLP Regressor, and LASSO). The result after this phase is the final result. All these processes can be illustrated in **Fig. 3-1**.

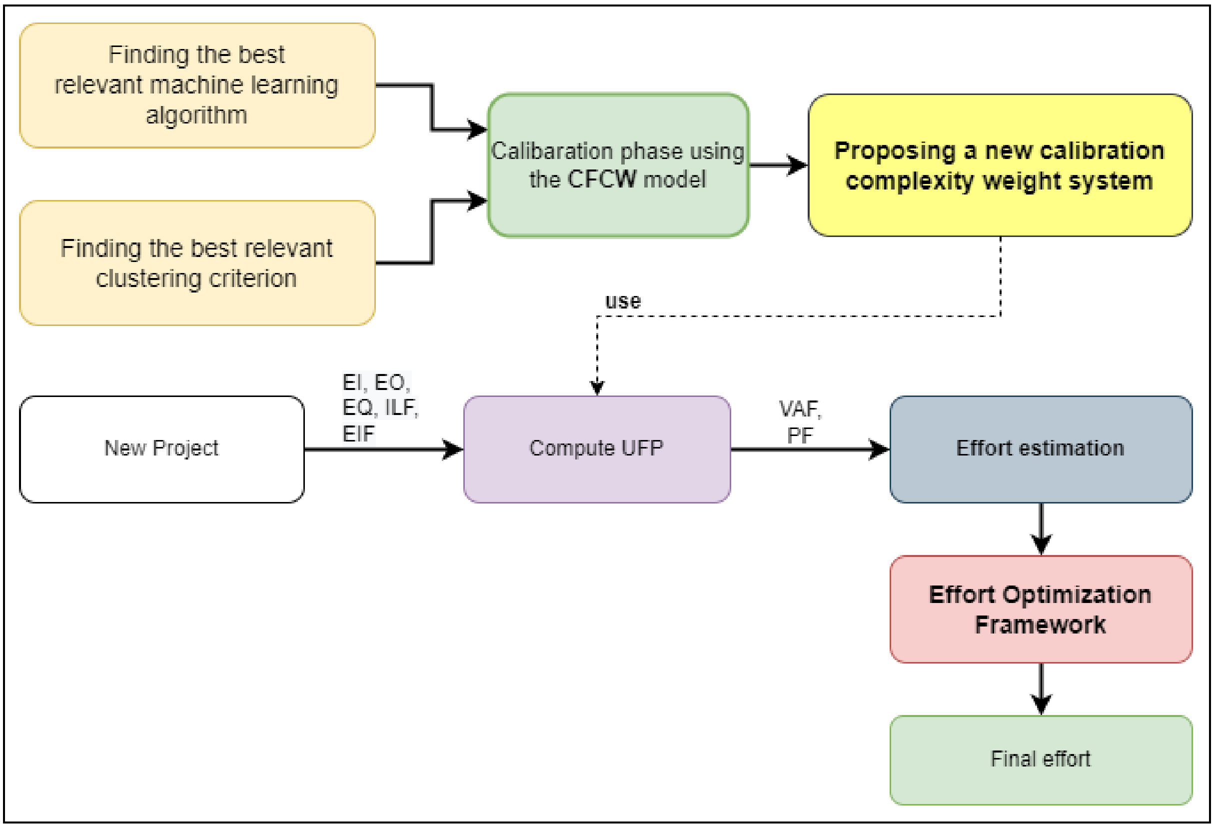

Fig. 3-1.Theoretical Framework

In the main step of proposing the new calibration complexity weight system, the Calibration of Functional Complexity Weight (CFCW) algorithm is proposed. This algorithm (CFCW) elicits the complexity weights from the EI, EO, EQ, EIF, and EIF variables using Bayesian Ridge Regression.

Another vital proposition is the effort optimization framework (named CFCW Optimization - CFCWO). This frame is constructed based on an ensemble algorithm with some base estimators. This framework takes a final role in the estimation of the effort.

3.1 Applying machine learning algorithms in effort estimation

Choosing the suitable algorithm for the estimator is a matter of first concern. Selecting a lousy algorithm can lead to estimating results that are not as expected. The development project can end up failing because of this selection. Moreover, for each different dataset, the suitable algorithm for it is also different. A practical algorithm for a dataset could not necessarily be adequate for another.

Many ML algorithms have been proposed and applied in recent years. Each algorithm proves its superiority in certain areas. Hence, between these algorithms, whether any algorithm will be suitable for the proposal in the research of this thesis. Another important thing is that we can hardly try all ML algorithms. So, choosing the test algorithms is also a challenging problem. In this study, we carry out a survey to find out the most used algorithms today and then select some algorithms to test.

Based on this survey, five most-used ML algorithms are Linear Regression, SVM, ANN, Ridge Regression, and LASSO. They are the algorithms used in turn in the CFCW model to infer the most suitable algorithm.

We perform an experiment to find the most suitable algorithm for proposing the new calibration functional complexity weight system based on these selected algorithms. This experimental process can be illustrated in **Fig. 3-2**.

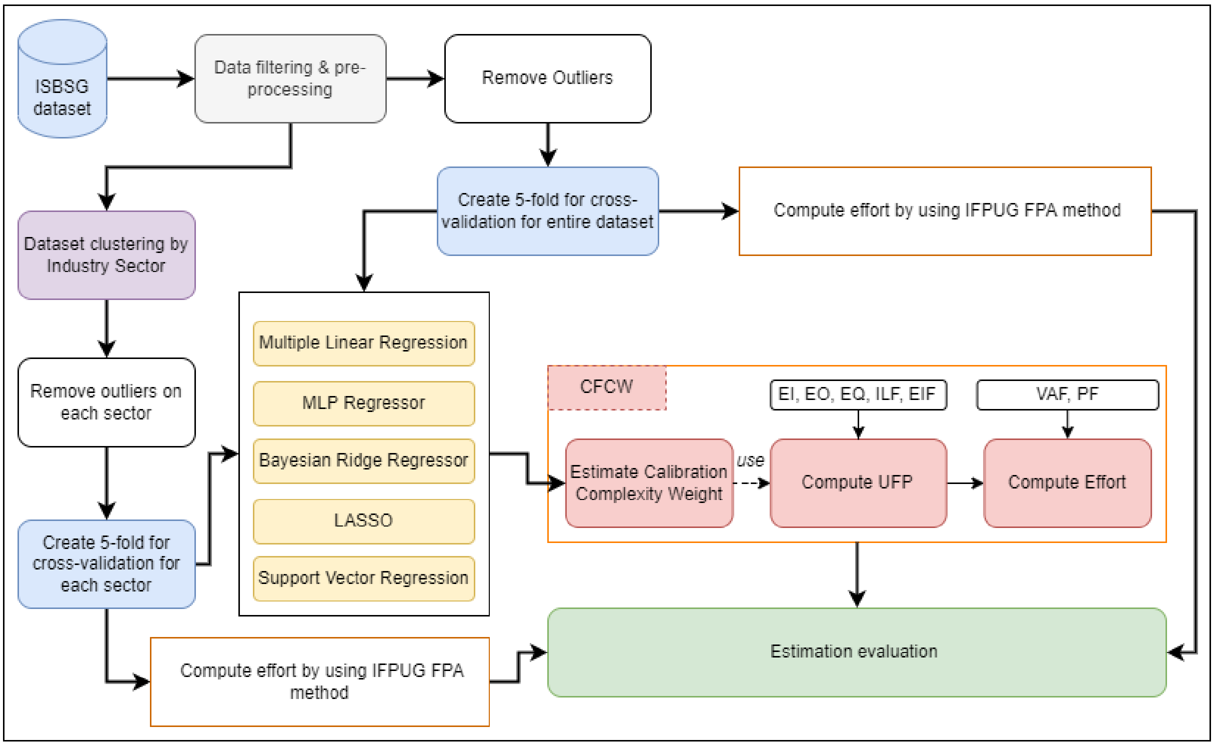

Fig. 3-2. Finding the best relevant algorithm

The following is a description of this procedure: First, the dataset will be filtered and preprocessed. After this step, the dataset is now called the tested dataset. Four cases need to be evaluated in this test. We can group them into two groups: the first group includes the first and second cases; the second group includes the third and fourth cases.

In the first case, the experiment will be conducted using the FPA method on the entire non-clustered dataset. This case is considered the baseline model for comparison with other models.

The second case will still be performed on the entire non-clustered dataset but now apply the CFCW algorithm in the effort estimation process. Specifically, this case will use five machine learning algorithms, MLR, MLP, BRR, LASSO, and SVR, respectively, to estimate complexity weight.

In the third case, the tested dataset will be segmented according to the Industry Sector categorical variable, then use the FPA method on the segments to estimate the effort.

The last case of this experiment is to use the CFCW algorithm on five machine learning algorithms on the clusters identified by the Industry Sector categorical variable segmentation.

These cases will be evaluated through the evaluation criteria mentioned in the section and compared. The ultimate goal is to find the algorithm with the lowest estimation error.

In short, the tested models used in this experiment are:

1. IFPUG FPA method on the entire non-clustered dataset.
2. CFCW method with the specific algorithm on the entire non-clustered dataset.
3. IFPUG FPA on the clusters formed by clustering the ISBSG dataset using the Industry Sector categorical variable.
4. CFCW method with the specific algorithm on the clusters formed by clustering the ISBSG dataset using the Industry Sector categorical variable.

3.2 Applying segmentation techniques in effort estimation

This section considers two segmentation approaches: 1) segmentation based on categorical variables; 2) segmentation based on clustering algorithms. This process aims to find the best suitable segmentation criterion for proposing the new calibration functional complexity weight system.

In general, the terms clustering and segmentation may have different meanings, but in this study, these two concepts are the same meaning and are interchangeable.

3.2.1 Using categorical variables

The same problem for algorithm selection; in the ISBSG dataset, many categorical variables can be used for segmentation. We can hardly test for all these variables. Therefore, through a survey to investigate the recent categorical variables, we will choose the representative variables for this study.

Based on the survey, the categorical variables used in this study are the most used categorical variables. They are Development Platform (DP), Industry Sector

(IS), Language Type (LT), Organization Type (OT), and Project Relative Size (RS). The test model is visualized in **Fig. 3-3**.

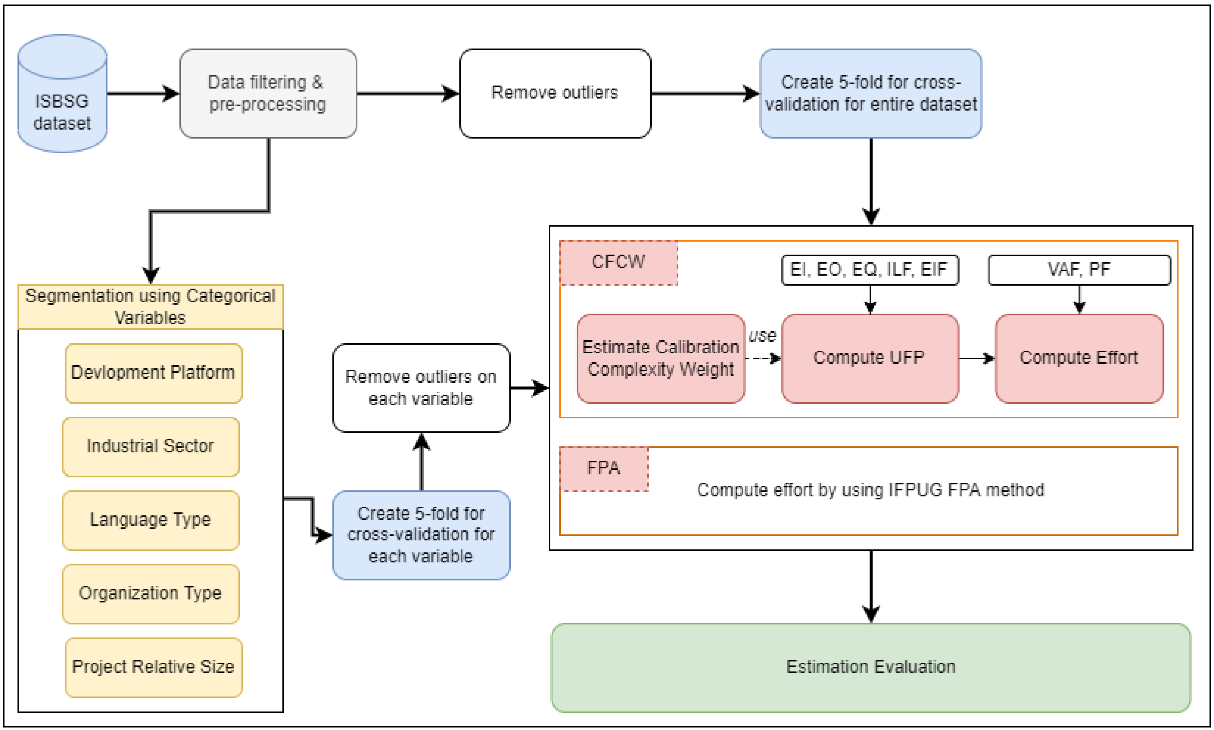

Fig. 3-3. Finding the best suitable categorical variables

There are four cases to evaluate in this model. The first case is still testing the IFPUG FPA method on an entire non-clustered dataset as a baseline model for comparison.

With the best suitable algorithm (BRR), the second case is tested on the entire dataset with the CFCW algorithm built on this best-fit machine learning algorithm.

The third and fourth cases start with segmenting the tested dataset according to the categorical variables mentioned. For each of these categorical variables, specific criteria are defined then the tested dataset will be segmented according to these criteria. The third case will be performed on the IFPUG FPA method, and the fourth case will be conducted on the CFCW method with the algorithm selected for the Complexity weight correction.

Tested models:

1. IFPUG FPA method on the entire non-clustered dataset.
2. CFCW method on the entire non-clustered dataset with suitable ML algorithm.
3. IFPUG FPA method on the clusters formed by clustering the ISBSG dataset using the specific categorical variable.
4. CFCW method on the clusters formed by clustering the ISBSG dataset using the specific categorical variable.

The segmentation criteria of the six categorical variables are: Development Platform (DP), Industry Sector (IS), Language Type (LT), Organization Type (OT), Project Related Size (RS)

3.2.2 Using segmentation algorithms

Previous researchers have proposed many clustering algorithms. People have tried to organize them into groups of relative similarity for easy access and application. However, this categorization is just relatively because there are algorithms that can belong to different groups.

Regarding the problem of categorizing clustering algorithms, there have been many studies looking at the classification of these clustering algorithms. In this study, according to these classifications, we try to choose algorithms so that they belong to different groups relatively. This choice is difficult because we cannot evaluate all existing clustering algorithms. A representative group is possible within the realm of possibility. Therefore, the following algorithms in **Table 3-1** are selected in the experiment of this study.

Table 3-1. Selected clustering algorithms

No.	Clustering Algorithms	Abbr.
1	Balanced Iterative Reducing and Clustering Hierarchies (BIRCH)	BIR
2	Fuzzy C-Means	FCM
3	Gaussian Mixture Model	GMM
4	k-means	KM
5	MeanShift	MS
6	Spectral	SC

The process of this experiment can be illustrated in **Fig. 3-4**. The features of all clustering algorithms in this study are EI, EO, EQ, ILF, EIF, and VAF.

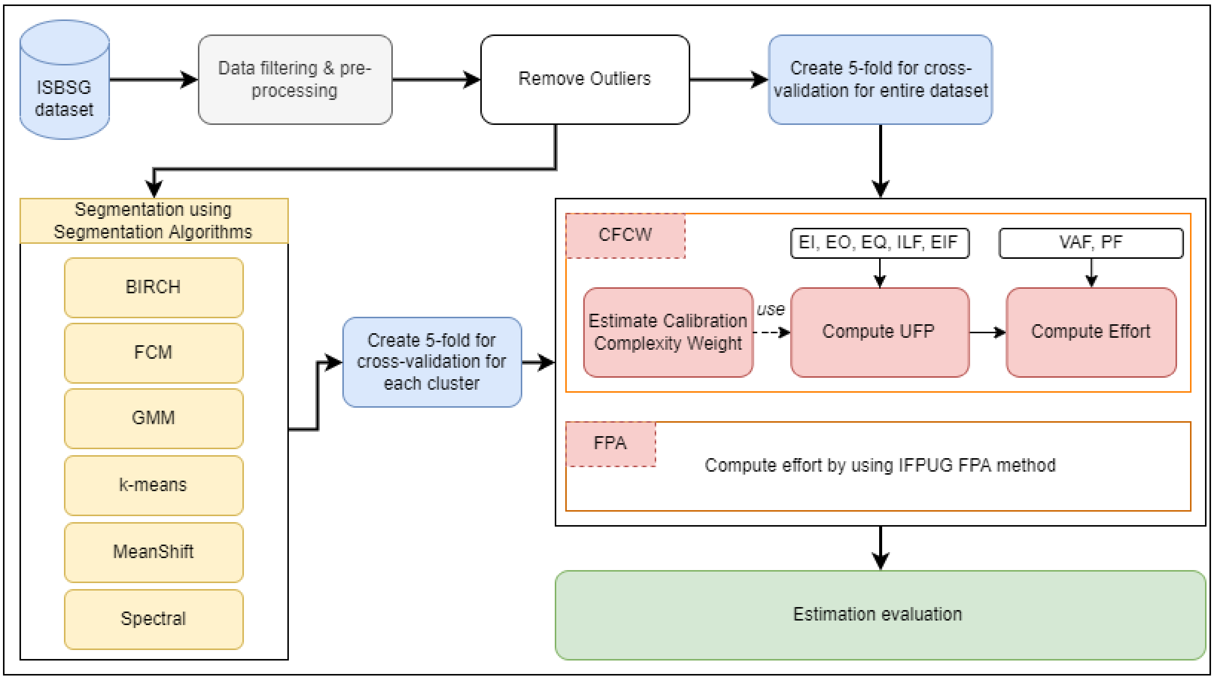

Fig. 3-4. Finding the best suitable segmentation algorithms

Firstly, the filtering and pre-processing of the ISBSG dataset should be done; Then, the outliers of the entire dataset should be detected and removed. After this step, the original dataset was called the tested dataset. Before performing the estimation step, the 5-fold cross-validation procedure was used for the full tested dataset in the baseline case. In the remaining cases, clustering algorithms were used to cluster the dataset. The tested dataset will be clustered into clusters, and the 5-fold cross-validation is also used on these clusters. The IFPUG FPA and CFCW estimate approaches were used in the estimation process. Finally, the results are compared using the estimation assessment with evaluation criteria. The CFCW algorithm, in this case also built on this best-fit machine learning algorithm (BRR).

Tested models in this experiment are:

1. IFPUG FPA method on the entire non-clustered dataset.
2. CFCW method on the entire non-clustered dataset.
3. IFPUG FPA method on clusters formed by clustering the ISBSG dataset using the specific clustering algorithm.
4. CFCW method on the clusters formed by clustering the ISBSG dataset using the specific clustering algorithm.

Some clustering algorithms don't need to determine the number of clusters as a parameter, while some are required. In selected clustering algorithms, BIRCH and MeanShift do not need to specify the number of clusters; Remains algorithms need this parameter. We will use the Silhouette methodology [30] in this study to determine the number of eligible clusters for each clustering methodology. **Fig.**

3-5 presents the results of determining the number of clusters with these algorithms.

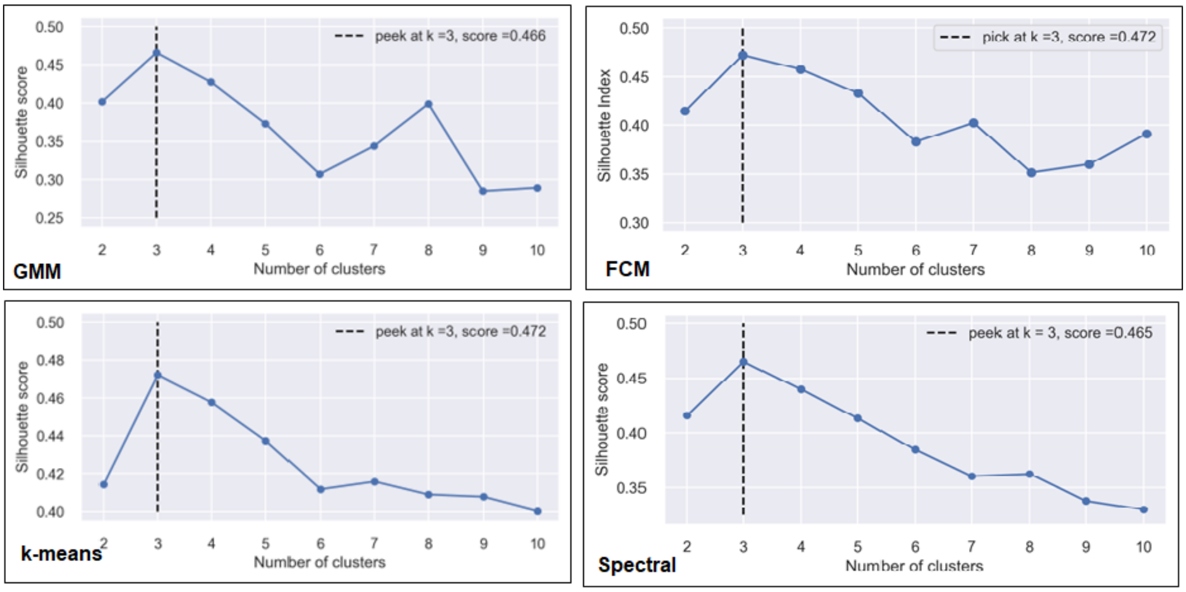

Fig. 3-5. Determining the number of k-optimal for algorithms

3.3 Propose a New Calibration System and Optimization Framework

After finding the best suitable algorithm and segmentation criteria, the next step will be to propose a new calibration complexity weight system and optimization framework. **Fig. 3-6** presents the experiment phases for this process.

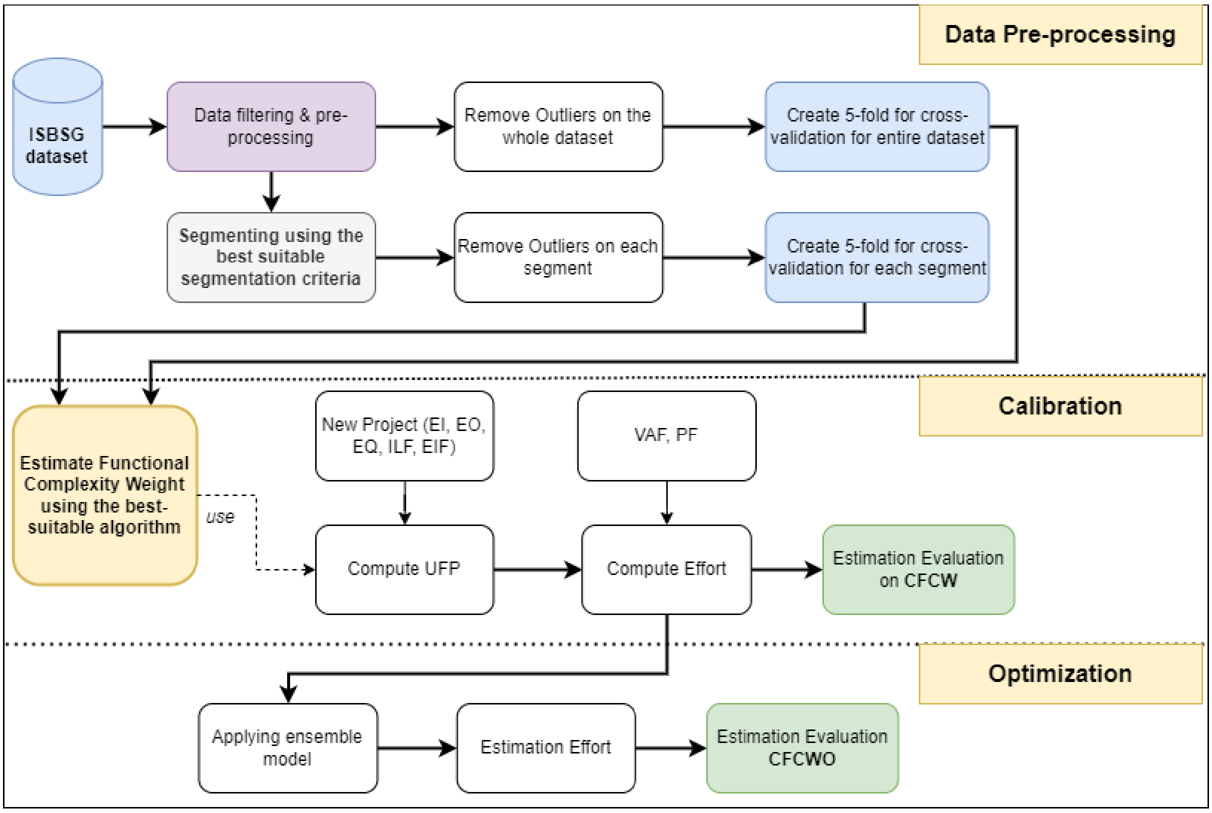

Fig. 3-6. Proposing new calibration and optimization algorithms

Basically, this process has three phases: 1) Data pre-processing, 2) Calibration, and 3) Optimization.

In the first phase, the dataset will be filtered as well as preprocessed. The data is now called the tested dataset. At this point, two experimental branches are deployed: one on the entire unsegmented tested dataset and the other on the tested dataset segmented according to the best-fit clustering criteria. Both branches are performed in the subsequent calibration phase. The CFCW algorithm will be applied in this calibration phase with the best-fit machine learning algorithm to propose a new calibration functional complexity weight system. This calibration functional complexity weight system will be used to calculate the UFP value, followed by the effort estimation. After that, the process transitions to the third phase, optimization. In this optimization phase, an ensemble model (Voting Regressor with base-estimators Random Forest Regressor, Bayesian Ridge, MLP Regressor, and LASSO) will be trained with the input of effort calculated in the second phase and then output the final result.

Four tested models are grouped into two groups in this process. The first group includes tested models 1 and 2, and the second group contains tested models 3 and 4.

1. Applying the FPA method to the entire unsegmented dataset.
2. Applying the CFCW method and then the CFCWO method to the entire unsegmented dataset.
3. Applying the FPA method to the segments yielded by the best-suitable segmentation criteria.
4. Applying the CFCW method and then the CFCWO method to the segments yielded by the best-suitable segmentation criteria.

4. MAIN RESULTS

This section presents the results archive from the experiment. Also, reiterate the goal of the investigation in this study is to find the most suitable algorithm and data segmentation criteria and then apply them to our proposal for a new calibration complexity weight system.

There are three subsections in this section: The first is finding the most suitable machine learning algorithm results, the second is the experimental result of determining the most appropriate segmentation criteria, and the third of proposing a new calibration complexity weight system.

4.1 Finding the best suitable Machine Learning algorithm

This section presents the results with the experimental model presented in section 6.2. In this section, there is a notation that we should mention. The CFCW model with the addition suffix is the notation for which algorithm means the CFCW model in which that algorithm replaces BRR. For example, CFCW-MLP means the CFCW model used by the algorithm MLP.

4.1.1 On the entire non-clustered dataset

In the first group (including 2 experimental tested models 1 and 2 in which the IFPUG FPA and CFCW methods are applied to the entire dataset without clustering). **Table 4-1** is the results compiled from the experiments.

Table 4-1. Mean values of evaluation results of all algorithms

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
FPA	379.89	10.524	623.38	0.879	0.106	0.087
CFCW-MLR	325.88	9.096	509.50	0.929	0.094	0.079
CFCW-MLP	354.99	9.917	568.10	0.900	0.100	0.083
CFCW-BRR	317.80	8.734	493.49	0.953	0.090	0.076
CFCW-LAS	325.83	9.094	509.41	0.929	0.094	0.079
CFCW-SVR	351.26	9.609	558.92	0.903	0.098	0.082

As we can see, the BRR algorithm is the best because it has the minimum estimation error. **Fig. 4-1** shows us the visualization of the experiment results.

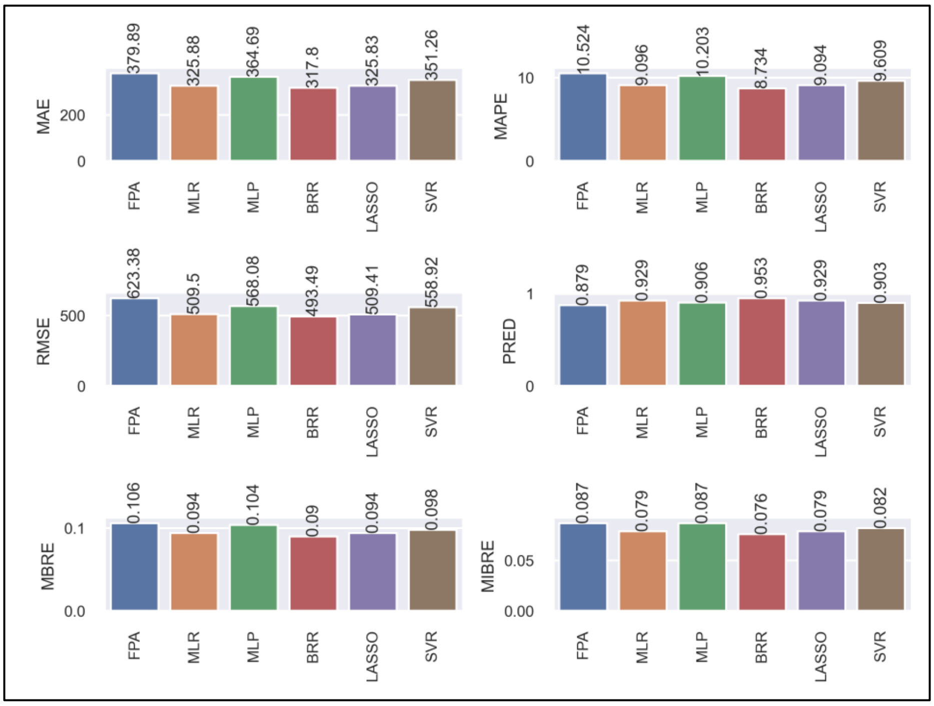

Fig. 4-1. evaluation results

4.1.2 On the clustered dataset

In the second group, the experiment includes two tested models (3 and 4). The IFPUG FPA and CFCW methods are applied to the segmenting dataset by Industry Sector categorical variables. **Table 4-2** are the results compiled from the experiment.

Table 4-2. Mean of all algorithms

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
FPA	459.53	13.085	722.37	0.868	0.132	0.104
CFCW-MLR	342.15	10.246	541.77	0.920	0.106	0.085
CFCW-MLP	415.16	11.745	667.03	0.897	0.120	0.095
CFCW-BRR	295.90	9.245	454.25	0.937	0.095	0.076
CFCW-LAS	341.47	10.232	540.32	0.920	0.106	0.085
CFCW-SVR	349.67	10.290	539.33	0.921	0.107	0.086

As we can see, the BRR algorithm also is the best because it has the minimum estimation error. **Fig. 4-2** shows us the visualization of the experiment results.

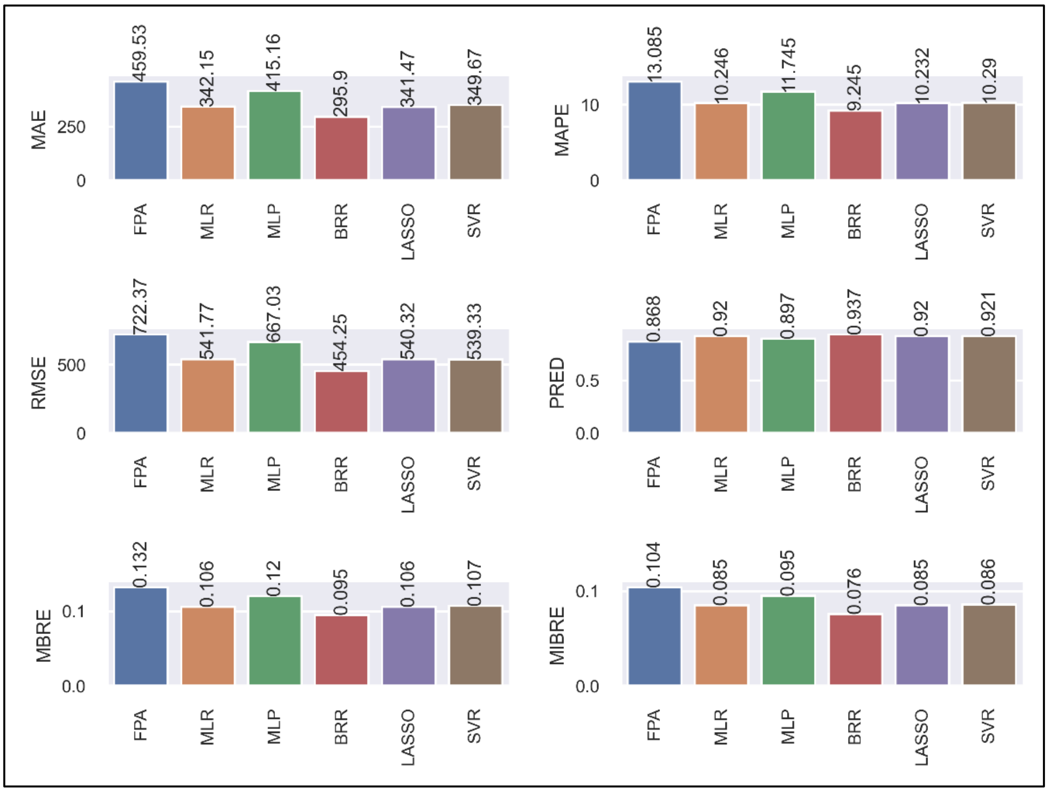

Fig. 4-2. Summary evaluation results

4.1.3 Summary

From the experiment, we can easily see that the BRR algorithm achieves the most optimal results with four tests in this group. This assertion is made based on all used evaluation criteria. In all tests, the BRR algorithm consistently achieves the lowest estimation error. To conclude this section, we can confirm that the BRR algorithm is the most suitable algorithm for this study.

4.2 Finding the best suitable clustering criterion

This section presents the results of finding the best segmentation criteria. The finding for the best suitable segmentation criterion is based on two main directions: segmentation by categorical variables and segmentation by clustering algorithms.

4.2.1 With categorical variables

For an overview, **Table 4-3** and **Table 4-4** are used to represent the combined results of FPA and CFCW methods on the categorical variables applied in this study. Each row in this table is the mean of all subgroups in each segment variable.

Table 4-3. The estimation results of the FPA method on all categorical variables

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
DP	362.02	10.430	583.92	0.906	0.105	0.085
IS	288.37	8.489	478.53	0.950	0.086	0.073
LT	366.46	9.763	579.91	0.897	0.099	0.083
OT	329.63	8.860	532.47	0.943	0.091	0.077
RS	350.75	9.168	536.74	0.917	0.094	0.078

Table 4-4. The estimation results of the CFCW method on all categorical variables

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
DP	274.30	8.131	410.44	0.957	0.085	0.071
IS	204.63	6.768	327.40	0.980	0.072	0.062
LT	300.36	8.514	442.27	0.971	0.088	0.076
OT	245.40	7.222	365.78	0.980	0.077	0.067
RS	290.95	7.970	415.98	0.934	0.084	0.072

Accordingly, we can easily see that the IS categorical variable criterion achieves the slightest estimation error on the FPA and CFCW methods. From this point, we can conclude that the categorical variable "IS" is the most suitable segmentation criterion among all the assessed categorical variables.

4.2.2 With segmentation algorithms

The experiment's segmentation algorithm findings are presented in this section. There are four tested models corresponding to each segmentation algorithm. With the first test model (IFPUG FPA on the entire non-clustered dataset). The results obtained corresponding to this model are shown in **Table 4-5**.

Table 4-5. FPA method on the whole dataset

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Ex 1	579.46	14.380	1294.07	0.816	0.146	0.103
Ex 2	588.86	14.339	1305.18	0.805	0.145	0.102
Ex 3	572.42	13.281	1308.24	0.839	0.134	0.094
Ex 4	579.38	12.815	1322.90	0.851	0.130	0.101
Ex 5	624.73	15.308	1356.72	0.816	0.155	0.106
mean	*588.97*	*14.025*	*1317.42*	*0.825*	*0.142*	*0.101*

With the second tested model (CFCW over the entire dataset), the obtained results corresponding to this model are shown in **Table 4-6**.

Table 4-6. CFCW method on the whole dataset

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Ex 1	518.02	13.225	1136.31	0.897	0.138	0.102
Ex 2	518.27	13.271	1132.10	0.885	0.137	0.101
Ex 3	547.42	12.947	1205.92	0.874	0.133	0.098
Ex 4	525.43	11.354	1179.33	0.920	0.118	0.095
Ex 5	565.27	14.301	1197.26	0.874	0.147	0.105
mean	*534.88*	*13.020*	*1170.19*	*0.890*	*0.135*	*0.100*

Fig. 4-3 shows the evaluation result of the first and second tested models' evaluation results in visual form. As we can see, the orange column (CFCW) always gets a better result than the blue (FPA) in all evaluation criteria.

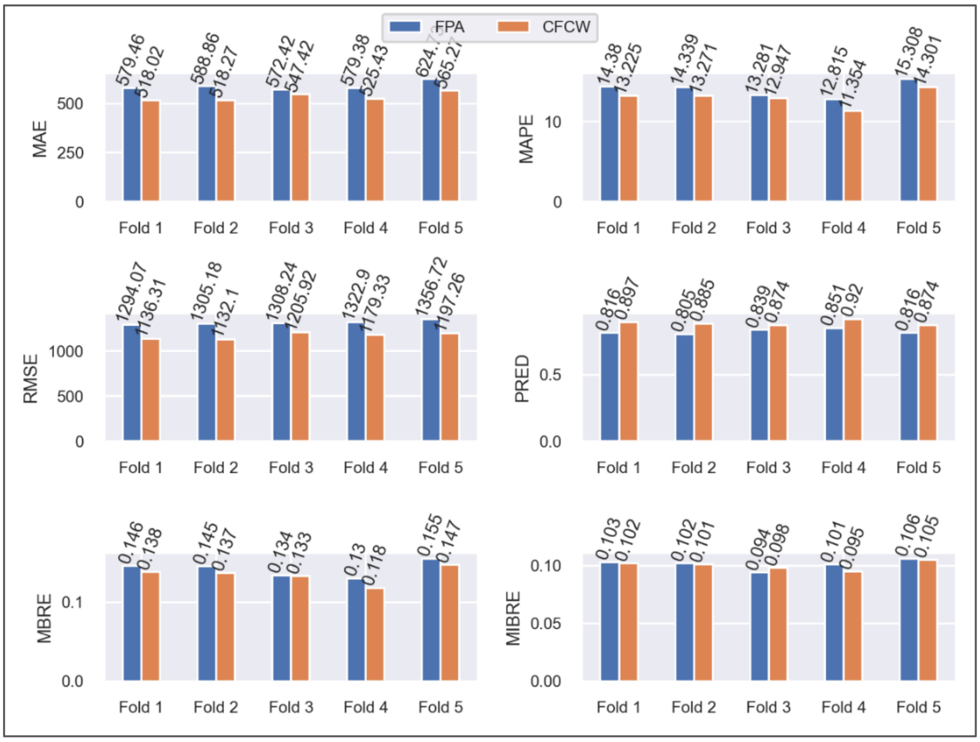

Fig. 4-3. The evaluation result of the first tested model

The detailed results above were summed up by taking the mean row across all algorithms. Then the results of FPA and CFCW methods on the algorithms are shown in **Table 4-7** and **Table 4-8**, respectively.

Table 4-7. FPA method on clusters formed by clustering algorithms

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Non-clustered	588.97	14.025	1317.42	0.825	0.142	0.101
BIRCH	373.35	9.317	656.88	0.888	0.094	0.079
FCM	378.12	8.881	682.41	0.892	0.089	0.075
GMM	369.14	8.648	657.01	0.897	0.087	0.073
k-means	371.40	9.373	771.34	0.854	0.094	0.076
MeanShift	376.66	9.700	626.92	0.876	0.098	0.081
Spectral	396.12	9.353	691.35	0.896	0.094	0.079

Table 4-8. CFCW method on clusters formed by clustering algorithms

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Non-clustered	534.88	13.020	1170.19	0.890	0.135	0.100
BIRCH	324.24	8.115	526.97	0.976	0.084	0.073
FCM	312.85	7.527	527.83	0.966	0.077	0.067
GMM	321.90	7.470	534.90	0.976	0.076	0.067
k-means	317.16	8.202	604.09	0.948	0.084	0.072
MeanShift	321.30	8.360	494.17	0.973	0.086	0.074
Spectral	353.78	8.334	579.83	0.958	0.085	0.075

Fig. 4-4 gives an overall view of all comparisons on all evaluation criteria: 1) between FPA and CFCW methods and 2) between all selected algorithms.

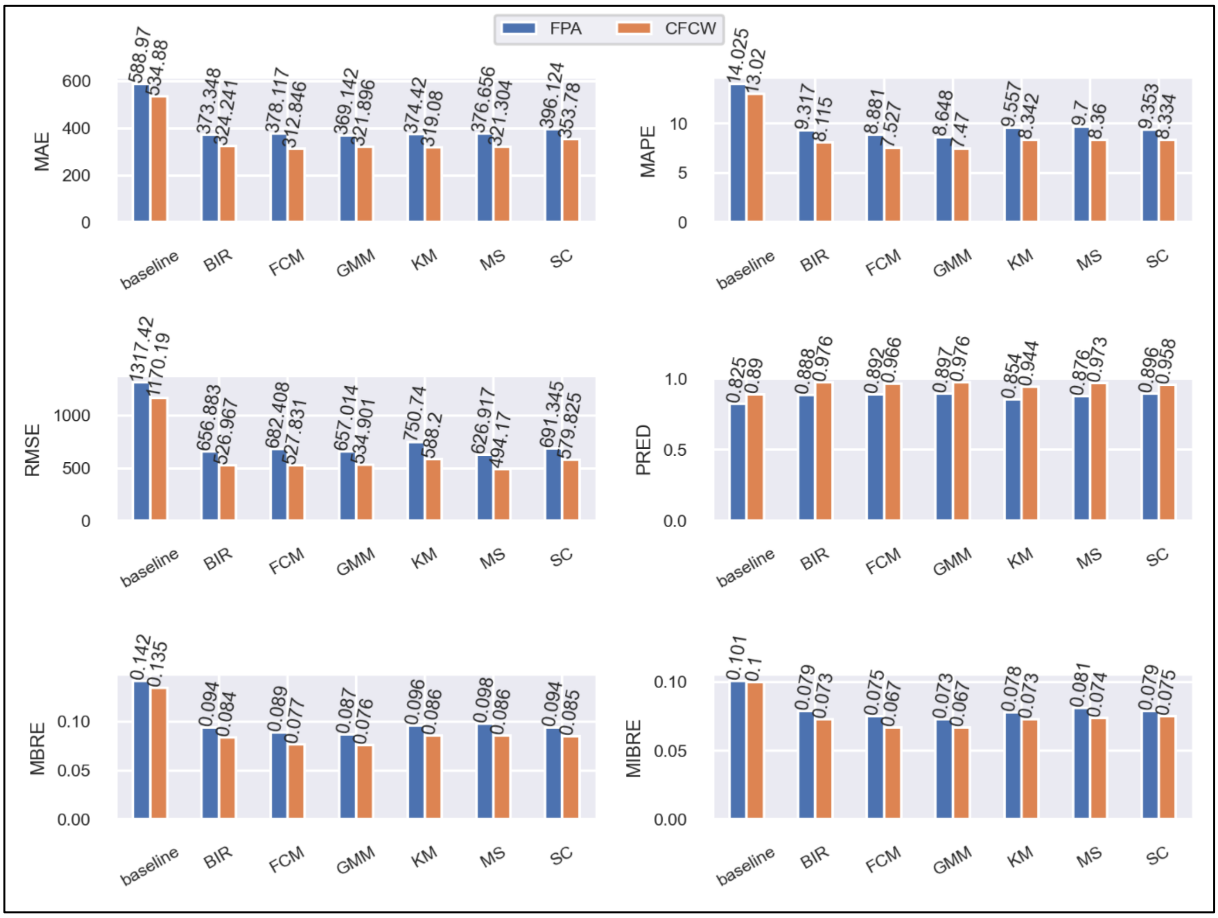

Fig. 4-4. Evaluation results of the FPA and CFCW-CA methods on clusters using the clustering algorithms

Corresponding to all algorithms, we consistently found that 1) applying clustering algorithms will give estimation accuracy better results than not applying, and 2) applying the CFCW method gives estimation accuracy better results than the FPA method.

To confine which algorithm is the best suitable for the evaluated dataset, a ranking table is created with the rating of each algorithm according to each evaluation criterion. A mean value of the evaluation criteria (EC) will also be determined, then considering the ranking position of each algorithm.

Table 4-9. The rank of algorithms with the FPA method

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE	mean of all EC	rank
WS	7	7	7	7	7	7	7.00	7
BIRCH	3	3	2	4	3	4	3.17	3
FCM	5	2	4	3	2	2	3.00	2
GMM	1	1	3	1	1	1	1.33	1
k-means	2	5	6	6	3	3	4.17	5
MeanShift	4	6	1	5	6	6	4.67	6
Spectral	6	4	5	2	3	4	4.00	4

Table 4-10. The rank of algorithms with the CFCW method

	MAE	MAPE	RMSE	PRED	MBRE	MIBRE	mean of all EC	rank
WS	7	7	7	7	7	7	7.00	7
BIRCH	5	3	2	1.5	3	4	3.08	3
FCM	1	2	3	4	2	1	2.17	2
GMM	4	1	4	1.5	1	1	2.08	1
k-means	2	4	6	6	3	3	4.00	4
MeanShift	3	6	1	3	6	5	4.00	4
Spectral	6	5	5	5	5	6	5.33	6

The results of the ranking process are presented in **Table 4-9** and **Table 4-10**. Accordingly, with both the FPA and CFCW methods, the GMM clustering algorithm has the highest accuracy.

4.2.3 Summary

The experiment's goal in this section is to find which segmentation criterion is best suitable for the analysed dataset. We examine two aspects of segmentation: 1) segmentation based on segmentation variables and 2) segmentation based on clustering algorithms. There are four tested models for this experiment. The final composite results are shown in the following tables:

Table 4-11. The most-suitable results when applying the FPA method to categorical variables and segmentation algorithms

Types	Criteria	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Categorical Variables	IS	288.37	8.489	478.53	0.950	0.086	0.073
Clustering Algorithms	GMM	369.14	8.648	657.01	0.897	0.087	0.073

Table 4-11 is built based on selecting the results of applying the CFCW method on the best suitable segmentation criteria (IS) and the best suitable

algorithm (GMM) based on six evaluation criteria. We can easily see that the FPA method applied to the clusters segmented by the IS categorical variable has a more miniature estimation error than the FPA method on the segments based on the GMM clustering algorithm.

Table 4-12 is built based on the selection of the results of applying the CFCW method on the best suitable segmentation criteria (IS) and the best suitable algorithm (GMM) based on six evaluation criteria. Accordingly, we can see that the estimation results using the CFCW method on the IS categorical variable have a smaller estimation error than the result achieved on clusters formed by the GMM algorithm.

Table 4-12. The most-suitable results when applying the CFCW method to categorical variables and segmentation algorithms

Types	Criteria	MAE	MAPE	RMSE	PRED	MBRE	MIBRE
Categorical Variables	IS	204.63	6.768	327.40	0.980	0.072	0.062
Clustering Algorithms	GMM	321.90	7.470	534.90	0.976	0.076	0.067

According to this result, with the segmentation criterion according to the IS categorical variable, the estimation accuracy consistently achieves better results than the remaining criteria on both FPA and CFCW methods. Thus, the classification variable "IS" is the most suitable segmentation criterion in this study.

4.3 New Calibration System and Optimization Framework

After determining the best-suitable machine learning algorithm (the BRR algorithm) and the best-relevant segmentation criterion (the IS categorical variable), this section aims to propose a new calibration complexity weight system. After applying this new system to software effort estimation, an optimization step will be applied to optimize the obtained results to give a better new result. Following are the results of this process.

In the case of the first tested group, the CFCW model and CFCWO model are sequentially applied to the unsegmented dataset. For comparison purposes, a baseline model is also applied in this case. The base model is generated by applying the FPA model to this unsegmented dataset.

The evaluation results based on six evaluation criteria of this group are listed in **Table 4-13**.

Table 4-13. Evaluation results of the first group experiment

Criteria	Methods	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	**mean**
MAE	FPA	695.25	732.93	639.30	671.35	654.79	**678.72**
	CFCW	654.14	667.36	607.66	633.22	646.01	**641.68**
	CFCWO	565.39	571.50	555.00	570.61	563.25	**565.15**
MAPE	FPA	12.692	14.947	13.721	15.099	14.445	**14.181**
	CFCW	12.425	14.033	13.267	14.174	14.168	**13.613**
	CFCWO	11.402	12.755	12.677	13.589	13.128	**12.710**
RMSE	FPA	1580.32	1633.22	1591.47	1537.69	1551.89	**1578.92**
	CFCW	1420.73	1447.52	1427.09	1383.34	1395.47	**1414.83**
	CFCWO	1089.72	1087.42	1077.51	1042.25	1027.52	**1064.88**
PRED	FPA	0.826	0.767	0.849	0.791	0.826	**0.812**
	CFCW	0.872	0.860	0.884	0.872	0.872	**0.872**
	CFCWO	0.872	0.860	0.884	0.872	0.872	**0.872**
MBRE	FPA	0.129	0.153	0.138	0.153	0.149	**0.144**
	CFCW	0.134	0.147	0.136	0.149	0.153	**0.144**
	CFCWO	0.120	0.132	0.131	0.144	0.139	**0.133**
MIBRE	FPA	0.099	0.114	0.096	0.107	0.105	**0.104**
	CFCW	0.103	0.113	0.099	0.110	0.112	**0.107**
	CFCWO	0.097	0.105	0.099	0.108	0.106	**0.103**

As we can see, after applying the CFCW and then CFCWO methods, the estimation error is consistently more minor using the FPA method. It means that the proposed methods get more accurate than the FPA method. The calibration complexity weight system of this case is proposed in **Table 4-14**. The columns Ex 1 to Ex 5 are the five experiments on 5-folds cross-validation. The “mean” column is the mean value of five experiments.

Table 4-14. The calibration complexity weight system on the unsegmented dataset

Components	Complexity Level	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	**mean**
EI	Low	3.63	2.85	3.27	4.02	3.63	**3.48**
	Avg.	0.6	0.68	1.72	0.44	2.4	**1.17**
	High	10.5	9.66	6.3	7.38	7.92	**8.35**
EO	Low	3.08	3.28	3.64	4.2	3.24	**3.49**
	Avg.	4.1	4.5	4.2	4.55	4.75	**4.42**
	High	4.27	6.37	7.21	3.08	3.78	**4.94**
EQ	Low	4.08	4.44	3	4.05	3.36	**3.79**
	Avg.	4.6	6.04	7.84	6.04	5.16	**5.94**
	High	3.3	0.66	1.38	1.26	4.8	**2.28**

ILF	Low	4.8	5.2	5.6	6.1	3.95	**5.13**
	Avg.	5.74	7.56	7.91	7.35	7.56	**7.22**
	High	12.9	9.4	7.7	8.9	9	**9.58**
EIF	Low	8.89	7.28	7.07	6.79	5.6	**7.13**
	Avg.	4.3	8.5	6.4	6.6	9.4	**7.04**
	High	15.6	13.05	15.45	17.25	17.4	**15.75**

In the case of the second group, the tested dataset was segmented by the IS categorical variable (including Banking, Communication, Financial, Government, Insurance, Manufacturing, Service Industry, and Others). After the segmentation phase, each segment was applied to the CFCW method to calculate the effort. These results will be applied to the CWCFO framework for the optimization phase.

Like the first group, a baseline model is also created for comparison purposes. This baseline model is based on applying the FPA method to each segment. Based on six evaluation criteria in Section 6.5, the results of this phase are shown as follows.

Table 4-15. The evaluation result of the second group experiment

Criteria	Methods	Banking	Communication	Financial	Government	Insurance	Manufacturing	Service Industry	others
MAE	FPA	463.96	391.92	219.86	567.01	516.91	207.29	319.78	354.73
	CFCW	301.02	213.70	195.12	508.22	376.64	192.84	294.48	268.76
	CFCWO	244.56	195.21	153.93	490.69	350.50	168.38	256.66	234.33
MAPE	FPA	10.625	14.416	8.152	8.413	13.369	11.191	7.413	10.151
	CFCW	7.359	10.062	7.722	7.708	11.473	10.422	6.638	8.161
	CFCWO	6.289	8.616	6.573	7.308	10.749	9.597	5.928	7.609
RMSE	FPA	828.43	574.89	315.90	1210.29	813.23	362.54	461.49	611.07
	CFCW	441.71	276.41	279.20	1095.56	548.40	311.59	370.71	381.67
	CFCWO	322.69	257.52	250.32	1039.66	523.46	233.31	340.40	329.97
PRED	FPA	0.909	0.820	1.000	0.933	0.738	0.782	0.867	0.875
	CFCW	1.000	0.940	1.000	0.933	0.862	0.855	0.956	0.975
	CFCWO	1.000	0.940	1.000	0.933	0.862	0.855	0.956	0.975
MBRE	FPA	0.106	0.144	0.089	0.087	0.134	0.113	0.074	0.102
	CFCW	0.076	0.114	0.087	0.081	0.126	0.107	0.068	0.085
	CFCWO	0.065	0.094	0.071	0.076	0.115	0.099	0.060	0.080
MIBRE	FPA	0.089	0.120	0.077	0.070	0.107	0.085	0.063	0.086
	CFCW	0.068	0.097	0.074	0.066	0.102	0.084	0.060	0.075
	CFCWO	0.060	0.081	0.061	0.063	0.094	0.080	0.052	0.071

As we can observe, the evaluation results always decrease from FPA to CFCW and then to CFCWO in all segments for each evaluation criterion. That means that the CFCW method achieves higher accuracy than the FPA method, and the CFCWO method is consistently more accurate than the CFCW.

The proposed calibration complexity weight system from this experiment is shown in **Table 4-16**. The first column is the name of the components (EI, EO, EQ, ILF, and EIF), and the second column is the complexity level (including Low (L), Average (A), and High (H)). The third column contains the complexity weight values from the IFPUG FPA method. The unsegmented column is the complexity weight corresponding to the unsegmented dataset. The remains columns are the complexity weight of the segments in IS categorical variable.

Table 4-16. Proposed Calibration Complexity Weight system on segments of the IS categorical variable

Components	Level	FPA	Unsegmented	Banking	Communication	Financial	Government	Insurance	Manufacturing	Service Industry	others
EI	L	3	3.48	1.73	1.10	3.38	0.76	1.81	3.49	3.95	3.33
	A	4	1.17	1.21	4.04	4.15	7.13	4.48	0.99	5.50	1.15
	H	6	8.35	7.90	4.07	4.68	7.33	3.60	8.54	6.72	9.53
EO	L	4	3.49	3.06	3.42	5.24	2.87	4.19	3.58	3.83	3.07
	A	5	4.42	5.37	2.49	1.99	5.40	5.68	5.28	4.74	4.31
	H	7	4.94	8.09	7.60	6.37	5.50	5.42	6.89	6.37	7.20
EQ	L	3	3.79	0.71	1.51	3.41	2.90	3.55	2.00	2.42	2.83
	A	4	5.94	4.39	3.45	3.71	4.18	6.40	4.81	4.26	6.11
	H	6	2.28	9.48	4.37	5.90	3.32	6.10	5.78	4.81	3.38
ILF	L	5	5.13	6.26	4.92	4.14	7.91	4.73	5.52	3.07	3.07
	A	7	7.22	2.24	10.65	9.58	5.84	4.65	7.99	3.53	10.15
	H	10	9.58	15.96	9.82	7.24	7.20	9.08	6.38	5.82	11.30
EIF	L	7	7.13	2.23	6.93	8.64	6.12	10.26	8.55	5.82	6.34
	A	10	7.04	10.16	10.82	13.04	11.96	2.58	1.88	17.32	4.94
	H	15	15.75	24.90	8.19	10.23	19.53	9.42	22.41	16.59	12.99

Fig. 4-5 presents the evaluation results graphically.

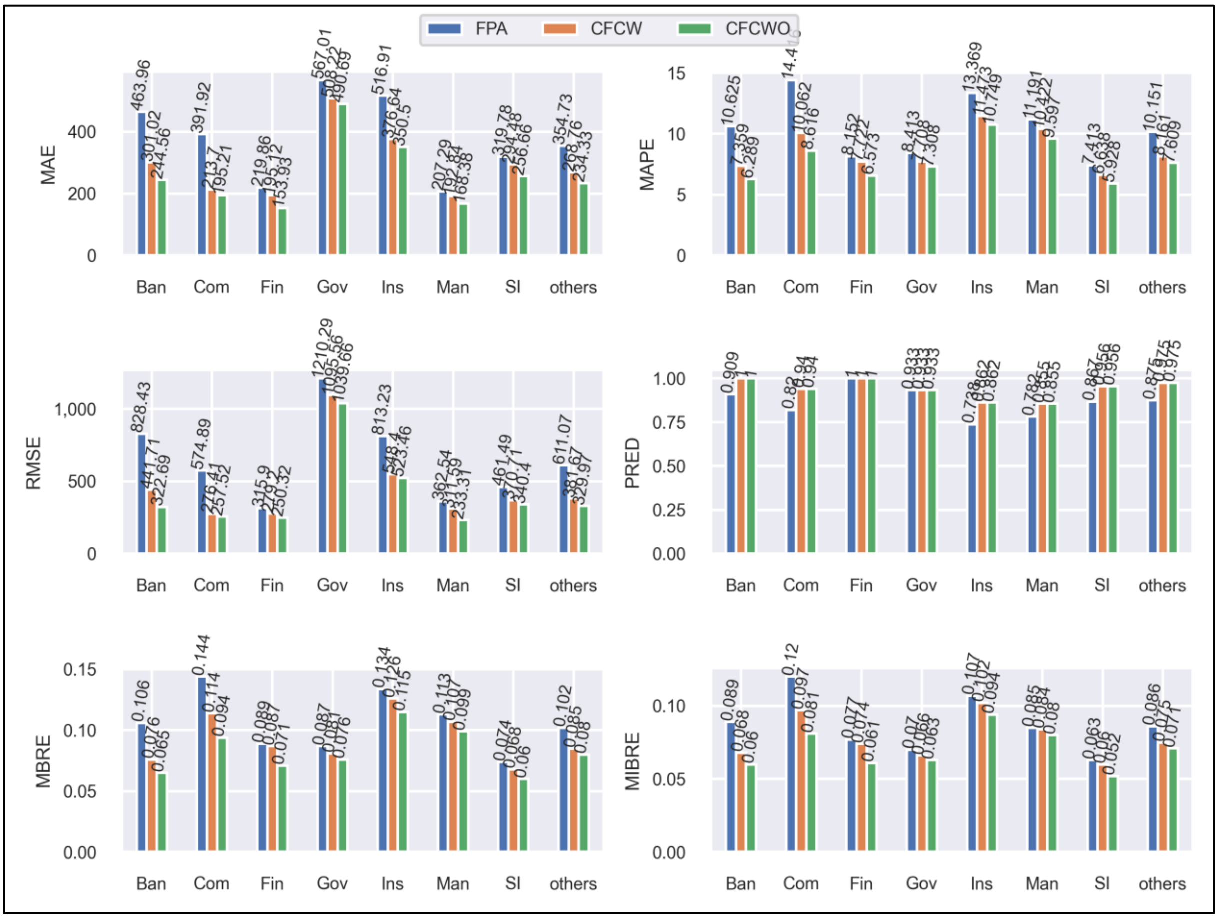

Fig. 4-5. Evaluation results of the experiment of the second group

4.3.1 Summary

This experiment has given a new calibration complexity weight system to be applied with the data segmented according to the IS categorical variable. When a new project needs to estimate the effort, it uses this new calibration complexity weight system to estimate the effort. The result of this phase then applies the optimization framework CFCWO to get more accurate results. Experimental results also confirmed that the proposed method accomplishes higher accuracy than the FPA method.

5. CONTRIBUTION OF THE THESIS TO SCIENCE AND PRACTICE

The main contribution of the thesis is the proposal of procedures for more accurate software effort estimation by improving the Functional Point Analysis method. The FPA method was born and is widely applied to the software industry. However, many reasons lead to this method being inadequate, as mentioned in section 4. That leads to the method needing to be updated to meet the evolving trends of the modern software industry.

Overall, it is possible to summarize the contributions as follows:

- The results of the performed experiments clearly showed that the estimates of the development effort using the new calibration complexity weight algorithm are more accurate than the estimates using the IFPUG FPA reference method.
- The effect of clustering has been demonstrated, allowing new algorithms to be applied to clustered data with the benefit of increasing the accuracy of effort estimation.
- The most suitable clustering algorithm and categorical variable were determined in the context of the study.
- A new framework has been created to optimize effort estimation based on improved FPA using regression models, machine learning, and clustering.
- Based on experimental results, the Bayesian Ridge Regressor (BRR) algorithm is the most appropriate approach to a new framework for optimizing software effort estimation.

6. CONCLUSION AND FUTURE WORK

FPA was proposed and has made significant contributions to the software industry. Besides, machine learning has also brought a big revolution in the field of computer science; Software development effort estimation is no exception. The application of machine learning in software effort estimation has been achieving many remarkable achievements. This study combines FPA (traditional) and machine learning (modern) methods to create a new method. In which, the effort estimate in principle is still based on FPA but with the complexity weight system on the basis of machine learning (CFCW). In addition, the results of the FPA-based estimation process are once again optimized to achieve higher accuracy (CFCWO). It has been demonstrated experimentally that with the proposal of this study, the accuracy can be improved markedly.

Because software engineering is a continually changing field, today's actual values may not correctly reflect software values tomorrow. As a result, the weights proposed in this work must be revised to reflect the new trend. The ISBSG dataset is a current database of companies from all over the world. It represents the dynamic nature of today's software industry. As a result, when project data is updated in the future, the IFPUG FPA weighting values should be recalculated to reflect the most recent software industry trends. Cause the coefficients are coherently related to data, the calibration process should be re-performed when using another dataset differ ISBSG.

In this thesis, the main work is focused on improving the effort estimation accuracy based on the functional complexity weight calibration. In fact, two other factors in effort estimation need to be considered, VAF and productivity factor. Future work will focus on these factors. With VAF, 14 GSCs are assessed as potentially obsolete, or some of these properties are not suitable for the current situation. It is necessary to find the proper criteria for modern software industry trends and their influence. With the productivity factor, new development technologies help significantly improve productivity. The determination of this factor is also another work that needs attention. In addition, other estimation methods such as COSMIC, FiSMA, and NESMA will also be studied as alternatives to the IFPUG FPA method.

7. REFERENCES

[1] P. Naur, B. Randell, Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968.

[2] T. Vera, S.F. Ochoa, and D. Perovich. Survey of software development effort estimation taxonomies. Technical Report, Computer Science Department, University of Chile, Chile, 2017.

[3] Khan, B., Khan, W., Arshad, M., & Jan, N. (2020). Software cost estimation: Algorithmic and non-algorithmic approaches. International Journal of Data Science and Advanced Analytics (ISSN 2563-4429), 2(2), 1-5.

[4] A. J. Albrecht, "Measuring application development productivity," Proc. IBM Applications Develop. Symp., pp. 83, 1979.

[5] IFPUG, Function Point Counting Practices Manual, Release 4.3.1, International Function Point Users Group, Westerville, Ohio, USA, 2010.

[6] Steven D. Sheetz, David Henderson, Linda Wallace, Understanding developer and manager perceptions of function points and source lines of code, Journal of Systems and Software, Volume 82, Issue 9, 2009, Pages 1540-1549, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2009.04.038.

[7] Kampstra, P, Verhoef. C: Reliability of function point counts — Department of Computer Science, VU University Amsterdam, Amsterdam, The Netherlands (2010).

[8] Kemerer, C. F: Reliability of function points measurement: A field experiment. Commun. ACM, vol. 36, no. 2, pp. 85–97 (1993).

[9] Meli, R: Functional metrics: Problems and possible solutions. Proc. 1st Eur. Software Meas. Conf., Antwerp, Belgium (1998).

[10] Xia, W., Capretz, L. F., Ho, D.: Neuro-fuzzy approach to calibrate function points, Proc. 8th WSEAS Int. Conf. Fuzzy Syst., pp. 116-119, (2007)

[11] Xia, W., Capretz, L. F., Ho, D., Ahmed, F.: A new calibration for function point complexity weights. Inf. Softw. Technol., vol. 50, no. 7–8, pp. 670-683, (2008).

[12] Xia, W., Ho, D., Captrez, L. F.: A neuro-fuzzy model for function point calibration. WSEAS Trans. Inf. Sci. Appl., vol. 5, no. 1, pp. 22-30, (2008).

[13] Ahmed, F., Bouktif, S., Serhani, A., Khalil, I.: Integrating function point project information for improving the accuracy of effort estimation. Proc. 2nd Int. Conf. Adv. Eng. Comput. Appl. Sci., pp. 193-198, (2008).

[14] Hajri, M. A., Ghani, A. A. A., Sulaiman, M. N., Selamat, M. H.: Modification of standard function point complexity weights system. J. Syst. Softw., vol. 74, no. 2, pp. 195-206, (2005).
[15] Ya-Fang, F., Xiao-Dong, L., Ren-Nong, Y., Yi-Lin, D., Yan-Jie, L.: A software size estimation method based on improved FPA. Proc. 2nd World Congr. Softw. Eng., pp. 228-233, (2010).
[16] Rao, K. K., Raju, G. S.: Error correction in function point estimation using soft computing technique. Proc. Int. Conf. Adv. Comput. Artif. Intell., pp. 194-198, (2011).
[17] J. Wen, S. Li, Z. Lin, Y. Hu, C. Huang, "Systematic literature review of machine learning based software development effort estimation models," Inf. Softw. Technol., 54 (1) (2012), pp. 41-59.
[18] Phannachitta, P., Matsumoto, K., "Model-based software effort estimation - A robust comparison of 14 algorithms widely used in the data science community," (2019) International Journal of Innovative Computing, Information, and Control, 15 (2), pp. 569-589.
[19] Phannachitta, P, "On an optimal analogy-based software effort estimation," Information and Software Technology, Volume 125, 2020, 106330, ISSN 0950-5849.
[20] S. Shukla and S. Kumar, "Applicability of Neural Network Based Models for Software Effort Estimation," 2019 IEEE World Congress on Services (SERVICES), 2019, pp. 339-342.
[21] S. Shukla, S. Kumar, and P. R. Bal, "Analyzing Effect of Ensemble Models on Multi-Layer Perceptron Network for Software Effort Estimation," 2019 IEEE World Congress on Services (SERVICES), 2019, pp. 386-387.
[22] A G, Priya Varshini, Anitha Kumari K and Vijayakumar Varadarajan. “Estimating Software Development Efforts Using a Random Forest-Based Stacked Ensemble Approach.” Electronics 10 (2021): 1195.
[23] M. Hammad and A. Alqaddoumi, "Features-Level Software Effort Estimation Using Machine Learning Algorithms," 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2018, pp. 1-3, DOI: 10.1109/3ICT.2018.8855752.
[24] Aroba, J., Cuadrado-Gallego, J. J., Sicilia, M. Á., Ramos, I., & Garcia-Barriocanal, E. (2008). Segmented software cost estimation models based on fuzzy clustering. Journal of Systems and Software, 81(11), 1944-1950.
[25] P. Silhavy, R. Silhavy and Z. Prokopova, "Categorical Variable Segmentation Model for Software Development Effort Estimation," in IEEE Access, vol. 7, pp. 9618-9626, 2019.

[26] Azath, H., Mohanapriya, M., & Rajalakshmi, S. (2020). Software Effort Estimation Using Modified Fuzzy C Means Clustering and Hybrid ABC-MCS Optimization in Neural Network. Journal of Intelligent Systems, 29(1), 251-263.

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

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

[29] T. R. Benala, R. Mall, S. Dehuri and K. Chinna Babu, "Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks," 2012 World Congress on Information and Communication Technologies, 2012, pp. 115-120, doi: 10.1109/WICT.2012.6409060.

[30] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math, vol. 20, no. C, pp. 53-65, 1987.

LIST OF PUBLICATIONS

Journal:

1. **Hai V.V**, H.L.T.K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "A New Approach to Calibrating Functional Complexity Weight in Software Development Effort Estimation," Computers 11, no. 2: 15, DOI: 10.3390/computers11020015, 2022.
2. **Hai V.V.**, H.L.T.K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "Towards improving the efficiency of software development effort estimation via clustering analysis," IEEE Access, vol. 10, pp. 83249-83264, DOI: 10.1109/ACCESS.2022.3185393, 2022.
3. H.LT.K. Nhung, **V.V. Hai**, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric Software Effort Estimation Based on Optimizing Correction Factors and Multiple Linear Regression, " IEEE Access, vol. 10, pp. 2963-2986, DOI: 10.1109/ACCESS.2021.3139183, 2022.

Conference:

4. **V.V. Hai**, H.L.T.K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, “Analysing the effectiveness of the Gaussian Mixture Model clustering algorithm in Software Enhancement Effort Estimation,” ACIIDS 2022.

5. **V. V. Hai**, Mohsin Javed, Zuhair Abbas, Meryem Ari, and Michaela Bílá, "On the Software Projects' Duration Estimation using Support Vector Regression," Software Engineering Perspectives in Intelligent Systems, CoMeSySo 2022, Advances in Intelligent Systems and Computing, Springer, Cham.
6. **V.V. Hai**, H.L.T.K. Nhung, and R. Jasek, "Toward applying agglomerative hierarchical clustering in improving the software development effort estimation," Lecture Notes in Networks and Systems, vol. 501 LNNS, pp. 353-371, DOI: 10.1007/978-3-031-09070-7_30, 2022.
7. **V.V. Hai**, H.L.T.K. Nhung, H.T. Hoc, "Calibrating Function Complexity in Enhancement Project for Improving Function Points Analysis Estimation," Lecture Notes in Networks and Systems, 232 LNNS, pp. 857-869, DOI: 10.1007/978-3-030-90318-3_67, 2021.
8. **V.V. Hai**, H.L.T.K. Nhung, H.T. Hoc, "Empirical Evidence in Early-Stage Software Effort Estimation Using Data Flow Diagram," Lecture Notes in Networks and Systems, 230, pp. 632-644, DOI: 10.1007/978-3-030-77442-4_53, 2021.
9. **V.V. Hai**, H.L.T.K. Nhung, H.T. Hoc, "A Productivity Optimising Model for Improving Software Effort Estimation," Advances in Intelligent Systems and Computing, 1294, pp. 735-746, DOI: 10.1007/978-3-030-63322-6_62, 2020.
10. **V.V. Hai**, H.L.T.K. Nhung, H.T. Hoc, "A Review of Software Effort Estimation by Using Functional Points Analysis," Advances in Intelligent Systems and Computing, 1047, pp. 408-422, DOI: 10.1007/978-3-030-31362-3_40, 2019.
11. H.LT.K. Nhung, **V.V. Hai**, and R. Jasek, "Towards a Correction Factors-based Software Productivity using Ensemble approach for Early Software Development Effort Estimation," Lecture Notes in Networks ans Systems, vol. 501 LNNS, pp. 413-425, DOI: 10.1007/978-3-031-09070-7_35, 2022.
12. H.LT.K. Nhung, **V.V. Hai**, and H.T. Hoc, "Analyzing Correlation of the relationship between Technical Complexity Factors and Environmental Complexity Factors for Software Development Effort Estimation", Lecture Notes in Networks and Systems, 232 LNNS, pp. 835-848, DOI: 10.1007/978-3-030-90318-3_65, 2021.
13. H.LT.K. Nhung, **V.V. Hai**, and H.T. Hoc, "Evaluation of Technical and Environmental Complexity Factors for Improving Use Case Points Estimation," Advances in Intelligent Systems and Computing Springer, 1294, pp. 757–768, DOI: 10.1007/978-3-030-63322-6_64, 2020.
14. H.L.T.K. Nhung, H.T. Hoc, and **V.V. Hai**, "A Review of Use Case-Based Development Effort Estimation Methods in the System Development Context," Advances in Intelligent Systems and

Computing, 1046, pp. 484-499, DOI: 10.1007/978-3-030-30329-7_44, 2019.
15. H.T. Hoc, **V.V. Hai**, H.L.T.K. Nhung, "An Approach to Adjust Effort Estimation of Function Point Analysis," Lecture Notes in Networks and Systems, 230, pp. 522-537, DOI: 10.1007/978-3-030-77442-4_45, 2021.
16. H.T. Hoc, **V.V. Hai**, H.L.T.K. Nhung, "AdamOptimizer for the Optimisation of Use Case Points Estimation," Advances in Intelligent Systems and Computing, 1294, pp. 747-756, 2020.
17. H.T. Hoc, **V.V. Hai**, H.L.T.K. Nhung, " A Review of the Regression Models Applicable to Software Project Effort Estimation," Advances in Intelligent Systems and Computing, 1047, pp. 399-407, 10.1007/978-3-030-31362-3_39, 2019.

CURRICULUM VITAE

Name and tile:	Vo Van Hai, M.Sc.
Date and place of birth:	23/06/1977, Quang Nam, Vietnam.
Address:	Nam T.G. Masaryka 1281, 76001, Zlín, Czechia.
Email:	vo_van@utb.cz, vovanhai@iuh.edu.vn.

Education:

2018-Recent	Tomas Bata University in Zlín, Czech Republic. Doctoral studies, Engineering Informatics, Software Engineering.
2011-2014	Master of Computer Science. Information Technology Faculty, Hue University of Science, Vietnam.
1996-2000	Bachelor of Information System. Information Technology Faculty, University of Dalat, Vietnam.

Work experience

Lecturer, Faculty of Software Engineering, University of Industrial, Ho Chi Minh City, Vietnam, 2001-2014.

Visited lecturer, 2004-2017.

- Faculty of Information Technology, Tran Dai Nghia University, Ho Chi Minh City, Vietnam.
- VTC Academy, Vietnam.
- Faculty of Information Technology, Gia Dinh University, Ho Chi Minh City, Vietnam.
- FPT University and FPT Aptech, Vietnam

Software Developer, Monet+, Zlín, Czech Republic 2019-2021.

Consultant, Manulife Vietnam, 2014-2017.

Isolation Developer, 2008-2014

Manager, Nhan Tri Hop company, Vietnam, 2004-2008

Language skills

English – proficient in spoken and written

French – basic communication skills

Optimalizace odhadu softwarového úsilí zlepšením analýzy funkčních bodů

Optimisation of Software Effort Estimation by Improving Functional Points Analysis

Doctoral Thesis Summary

Published by: Tomas Bata University in Zlín,

nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic

Edition: published electronically

Typesetting by: Vo Van Hai, Ph.D.

This publication has not undergone any proofreading or editorial review.

First Edition

Publication year: 2022

ISBN 978-80-7678-129-0