Missing Data Imputation using K-Nearest Neighbour for Software Project Effort Prediction

Sri Handayaningsih[a,1*], Ardiansyah[b,2]

[a]Department of Information System, Faculty Sains and Applied Technology Software Engineering
[b]Department of Informatics, Faculty of Industrial Technology, Data Knowledge Research Group
[1]sriningsih@is.uad.ac.id*; [2] ardiansyah@tif.uad.ac.id
* Corresponding Author

Received 8 September 2021; accepted 21 December 2021; published 21 January 2022

ABSTRACT

The accurate of software development effort prediction plays an important role to estimate how much effort should be prepared during the works of a software project so that it can be completed on time and budget. Achieving good prediction accuracy is rely on the quality of data set. Unfortunately, missing data is one of big problem regards to the software effort data set, beside imbalance, noisy and irrelevant problem. Low quality of data set would decrease the performance of prediction model. This study aims to investigating the accuracy of software effort prediction with missing data set by using KNN missing data imputation and List Wise Deletion (LWD) techniques. It was continued by applying stepwise regression with backward elimination for feature selection and implementing two effort prediction methods of Multiple Linear Regression (MLR) and Analogy. The result shows that missing data imputation using KNN and listwise deletion with multiple linear regression approach outperforms the Analogy approach significantly (p>0.05)

KEYWORDS
Missing data
KNN
analogy
multiple linear regression
software effort prediction

1. Introduction

The prediction of the software project's effort so far depends on the acquired data set. The data must be of high quality because it influences the high or low accuracy of the prediction model [1],[2]. Problems occurred within various data sets include imbalance, noise, missing, irrelevant. Missing data (MD) is one of the most researched problems in the field of software engineering, especially in software effort prediction and software defect prediction. The performance of the model's accuracy can be affected or biased [3] depending on the success rate of handling MD, as one of the cleaning data activities which is part of the preprocessing phase [4]. In general, there are five approaches that can be applied in order to handle MD: ignoring, replace with default value, fill missing value manually based on your domain knowledge, replace with variable mean (if numerical) or most frequent value (if categorical), and also by using methods of K-NN, Naïve Bayes, Decision Tree or Expectation-Maximization (EM).

Some researchers intensively try to increase the accuracy of prediction models for software projects effort by improving MD imputation techniques. For example with the work done in [5] by proposing low-rank recovery and semi-structured regression imputation (LRSRI) techniques to determine the factors that drive the emergence of MD as well as doing the imputation towards identified missing data of effort, and also work done by [6] with proposing column-wise Guided Data Imputation (cGDI). [7] proposed the Bayesian Regression and EM (BREM) algorithm which was embedded into the missing data imputation technique, and CrossValidation based K-NN Imputation (CVbKNNI) proposed by [1]. While [8] applied three MD techniques of toleration, deletion and KNN along with two software effort prediction methods: Classical Analogy and Fuzzy Analogy.

Unlike the others, this study aims to contribute in the part of investigating the accuracy of software effort prediction with missing data set by using K-NN missing data imputation and ListWise Deletion (LWD) techniques. It will be continued by applying a stepwise regression method with backward elimination for feature selection and implementing two effort prediction methods of Multiple Linear Regression (MLR) and Analogy.

2. Method

2.1. Prepocessing

2.1.1 Missing Data Imputation

Missing data can be removed by ignoring, deleting or imputation techniques. Ignoring means leaving the data blank as it is, while deletion means deleting one tuple whose attributes have lost its data. Imputation means using certain techniques to fill in the missing data. One of the imputation techniques is the k-Nearest Neighbor (KNN) method. KNN calculates the similarity between projects according to the attribute types. There are three distance measurement techniques for numerical types; Euclidean, Manhattan and Minkowski which are proven to produce good similarity measurements according to [9] and [10].

In this study, the technique used is the Manhattan distance. This technique calculates absolute distance on each attribute without rooting as denoted in equation (1). Tuples that are nearest (*K = 1*) to the class or target are selected and used to fill in the blank data.

$$D(P,P') = \sum_{i=1}^{n} w_i Dis(f_i, f') \quad (1)$$

Notation *p* is a new project that will be estimated and *p'* is the old project that has been completed. The f_i and f'_i show the *i*-attribute value of a project,while $w_i = \{0, 1\}$ is the weight of the *i* -attribute.

2.1.2 Feature Selection

Feature selection towards independent variables is applied using the regression approach with stepwise regression and backward elimination. This technique was chosen compared to other techniques such as forward regression or backward regression because it is more popular than the two other techniques [11]. This technique chooses features that only have a strong and significant relationship (correlation) with the dependent variable. Thus, feature with the largest p-value or equal to $p > 0.05$ will be eliminated.

2.2. Effort Prediction Methods

This study employs two methods of prediction effort, namely Multiple Linear Regression (MLR) and Analogy. MLR is chosen due to its popularity among all algorithmic approach. While Analogy is a method with machine learning approach that produces the best accuracy among other methods with the same approach [12]. MLR requires data of previous project to be able to evaluate and predict the effort of a new project. MLR will calculate how strong the relationship (correlation) between dependent variable *(Y)* and independent variables (X_i) as defined in equation (2):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon \quad (2)$$

where *Y* is dependent variable or target; $X_1, X_2, ..., X_n$ are independent variables or features; β_0 is intercept parameter;$\beta_1, \beta_2, \cdots, \beta_n$ is a regression coefficient or slope; and ε is the error level.

Analogy is a method that predicts the effort of a new software project, based on certain similarities with past projects that have been done. Similarity measure is the first step to identify similar projects by using Euclidean, Manhattan or Minkowski distance techniques. The results of similarity measures are the list of past projects that have been sorted according to the smallest to the largest distance value. The smaller the distance value is considered to be more similar to the target project. The number of most similar K projects then being selected from this list. Fixed analogy selection with *K = 1* will be used in this study, which means that only an analogy is chosen (closest analogy) based on the smallest distance value. The last step is to calculate the target project effort through the adaptation rules mechanism which is the division between effort and size of old project multiplied by size of new project as defined by equation (3).

$$Effort_{new\ project} = \frac{Effort_{old\ project}}{Size_{old\ project}} Size_{new\ project} \tag{3}$$

2.3. Model Evaluation

In this study, Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE) and Pred (0.25) are used as the evaluation techniques to determine the accuracy of the model. Accuracy is obtained after model validation with an eight-fold cross validation technique. MMRE is generated by calculating the average MRE of each project in the data set. MMRE is one evaluation that is used to assess the efficiency of the effort to be estimated. While MRE is a statistical technique used to measure the accuracy of project estimates obtained from the absolute value of substraction of e_i and $\hat{e}_I$ divided by e_i, as shown in equation (4).

$$MRE = \frac{|e_i - e_i|}{e_i} \tag{4}$$

Symbol e_i shows the actual effort of the old project and $\hat{e}_i$ is the estimated effort of new project obtained using equation (3). MMRE is one of the accuracy measurements of a software project estimation model that calculates the average of MRE. The accuracy of the prediction model belongs to category of good if the MMRE value less or equal to 0.25.

$$MMRE = \frac{1}{n} \sum_{i=1}^{i=n} \frac{|e_i - e_i|}{e_i} \tag{5}$$

MdMRE measures the accuracy of software project prediction models by calculating the median of MRE. The accuracy of the estimation model is categorized as good if MdMRE is less or equal to 0.25 as denoted in equation (6).

$$MdMRE = median(MRE) \tag{6}$$

Pred (0.25) is an aggregate of the percentage of MRE which is less or equal to 0.25, as denoted in equation (7). The accuracy of the prediction model is in good category if Pred (0.25) is more or equal to 0.75.

$$Pred\ (0.25) = \frac{1}{n} x \sum_{i=1}^{n} (MRE < 0.25) \tag{7}$$

2.4 Experiments

The experimental framework which consists of Desharnais data sets is shown in Figure 1. Its preprocessing step consists of KNN missing data imputation and listwise deletion, data transformation using log and SQRT and selection features using stepwise regression techniques with backward elimination. Effort prediction using multiple linear regression and analogy, validation model with 8-fold cross validation and comparison model using independent t-Test.

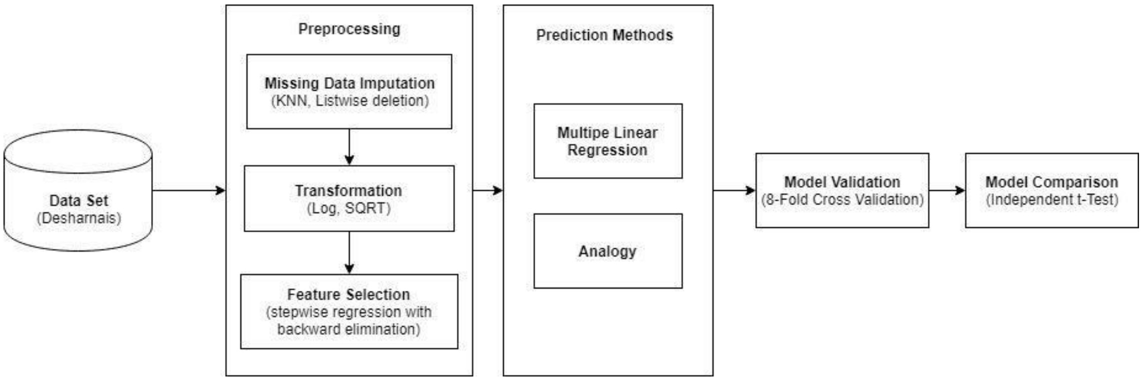

Figure 1. The Experimental Framework

3. Results and Discusion

3.1 Data set Description

The Desharnais data set consists of 81 software project data and has often been used in research related to estimating the effort of software projects such as research [13], [14]. Desharnais consists of 12 attributes including Project, YearEnd, TeamExp, ManagerExp, Length, Effort, Transactions, Entities, PointsAdjust, Envergure, PointsNonAdjust and Language. 5 categorical type of attributes (ordinal or nominal) and 7 quantitative attributes (interval or ratio). In order to meet the requirements using multiple linear regression, the data must be interval or ratio, so that after missing data handling the five categorical attributes will be eliminated. The selection of the Desharnais data set is based on the fact that the data set is widely used in research related to the prediction of the effort of the software project and also because it has four tuples with missing data, that is suitable for this research.

3.2. Missing Data Imputation

The application of KNN for missing data imputation produces the value taken from the most similar project (*K = 1*). The TeamExp and ManagerExp attributes in Project 38 are filled with values 2 and 4 because they are taken from the most common project of Project 23. While TeamExp attribute in Project 44 filled with value 4 because the most recent project is Project 21. Attributes ManagerExp in Project 66 is filled with value 1 because it was taken from Project 45, and the ManagerExp attribute in Project 75 is filled with value 4, which is taken from the project similar with Project 25. Table 1 shows the project whose data was successfully filled in based on *K = 1* project.

Table 1 The Result of MD imputation KNN towards feature TeamExp and ManagerExp

Project ID	TeamExp	ManagerExp	K=1
38	**2***	**4***	Project ID 23
44	**4***	4	Project ID 21
66	2	**1***	Project ID 45
75	0	**4***	Project ID 25

3.3 Data Transformation

The results of normality test with Kolmogorov Smirnov obtained Length, Effort, Transactions, Entities, Point Adjust, and PointsNonAdjust features had value of $p = 0.000$ ($p < 0.05$), which cause the six features are not normally distributed. While the Envergure *feature* has $p = 0.200$ ($p > 0.05$), causes the feature is normally distributed. These five features will then be transformed so that the data is normally distributed. The Feature Length and Effort data transformation uses Log10 since the histogram forms substantial positive skewness. While the data transformation of Transactions, Entities, Point Adjust and PointNonAdjust are using SQRT since the histogram forms a moderate positive skewness. Six feature has $p > 0.05$ which means that it is normally distributed as shown in Table 2.

Table 2. The Results of Data Transformation with p>0.05

Feature	Histogram	Data Transformation	.sig(p>0.05)
Leght	Substantial Positive Skewness	Log 10(x)	0.89
Effort	Substantial Positive Skewness	Log 10(x)	0.200
Transactions	Moderate Positive Skewness	SQRT (X)	0.82
Entities	Moderate Positive Skewness	SQRT (X)	0.88
PointAdjust	Moderate Positive Skewness	SQRT (X)	0.81
Point Non Adjust	Moderate Positive Skewness	SQRT (X)	0.200

3.4 Feature Selection

Pearson correlation test is done first to determine the correlation between independent variables with the dependent variable, namely the Effort feature. The results show that all features are strongly correlated, straight-line, positive and significant, and only Transactions feature that has correlation of sufficient. In the coefficient value, it is known that the five features have *p*-value>0.05 and only the *feature* Lengths which are smaller than 0.05 (*p* = 0.003). As the consequences, these five features are potential to eliminate. But for the first iteration, PointNonAdjust feature is delimited because it has the largest *p*-value of 0.717. In the second iteration, PointAdjust feature has the largest pvalue of 0.606 so it needs to be eliminated. In the third iteration, the biggest p-values is 0.090 in Transactions feature, so it needs to be eliminated. While in the fourth iteration obtained three features that have *p*-value<0.05, which are Envergure, Length and Entities. Thus, the feature selection results has successfully selected three of the six features in the data set as presented in Table 3.

Table 3. Iteration of Stepwise Regression with Backward Elimination

Feature	Iteration(Sig)			
	1	2	3	4
Envergure	.656	.002	.001	.000
Leght	.003	.003	.003	.000
Transactions	.462	.382	.090	-
Entitites	.168	.131	.000	.001
Point Adjust	.610	.606	-	-
Point non adjust	.717	-	-	-

In order to be implemented into multiple linear regression as denoted in equation (2), it is necessary to define intercept parameters and regression coefficients first. Table 4 presents a list of intercept parameters and regression coefficients used in KNN and LWD of MD imputation techniques at 8-fold cross validation.

Table 4. List of intercept parameter and Regression Coefficient

Set	β_0		β_1		β_2		β_3	
	KNN	LWD	KNN	LWD	KNN	LWD	KNN	LWD
1	2.380	2.364	.550	.574	.032	.031	.010	.010
2	2.453	2,448	.492	.505	.032	.031	.011	.010
3	2.41	2.398	.438	.458	.039	.038	.012	.012
4	2.48	2.490	.495	.484	.029	.026	.011	.012
5	2.416	2.392	.420	.444	.035	.034	.012	.012
6	2.394	2.400	.522	.549	.028	.027	.013	.012
7	2.554	2.578	.502	.451	.028	.029	.009	.009
8	2.442	2.402	.507	.515	.029	.032	.011	.011

3.5 Model Accuracy

An absolute residual (AR) value measures the accuracy of software project effort prediction model. The smaller the value of AR means that the estimated value with the actual value is almost equal. Table 5 presents the comparison of the results of the MRE and AR of the four models: MLR-KNN, MLR-LWD, Analogy-KNN and Analogy-LWD. From the MLR-KNN model, the best combination is obtained at set-8 due to its lowest MRE of 7%. The MLR-LWD model is the best combination on set-1 with MRE 9.8%, the Analogy-KNN model and Analogy-LWD both at set-6 with 4.7% of MRE.

Table 5. Model accuracy through the four models

	MLR-KNN			**MLR-LWD**			**Analogy-KNN**			**Analogy-LWD**		
	MRE	**AR**	**Actual**	**MRE**	**AR**	**Actual**	**MRE**	**AR**	**Actual**	**MRE**	**AR**	**Actual**
Set 1	.094	.353	3,75	**.098**	**.366**	**3,75**	.32	1,147	3,59	.32	1,147	3,59
Set 2	.129	.378	2,92	.119	.347	2,92	.39	1,371	3,50	.39	1,371	3,50
Set 3	.19	.541	2,85	.19	.542	2,85	.72	2,061	2,85	.72	2,061	2,85
Set 4	.18	.519	2,81	.19	.545	2,81	.90	3,023	3,37	.90	3,023	3,37
Set 5	.10	.422	4,16	.10	.427	4,16	.32	1,123	3,56	.80	2,603	3,24
Set 6	.07	.286	3,83	.09	.278	3,10	**.47**	**1,639**	**3,47**	**.47**	**1,639**	**3,47**
Set 7	.21	.665	3,15	.21	.647	3,15	.54	2,119	3,96	.66	2,487	3,77
Set 8	**.07**	**.324**	**4,38**	.08	.307	3,90	.42	1,543	3,71	.32	1,147	3,59
MMRE		**.060**			**.061**			**.21**			**.22**	
MdMRE		**.047**			**.052**			**.19**			**.19**	
Pred(.25)		**1**			**1**			**.69**			**.73**	

Figure 2 shows the accuracy comparison of four models based on MMRE, MdMRE and Pred (25). It is shown that MMRE and MdMRE of the MLR-KNN method produce better values than the Analogy-KNN and Analogy-LW but has a slight difference with MLR-LWD. Similarly does with Analog-KNN which has a slight advantage with Analogy-LWD. Whereas Pred (25) for MLRKNN and MLR-LWD have the same value of 100% and far outperform Analogy-KNN and Analogy LWD which are 69% and 73% respectively. It also turns out that the Pred (25) value of AnalogyLWD is outperform the Analogy-KNN.

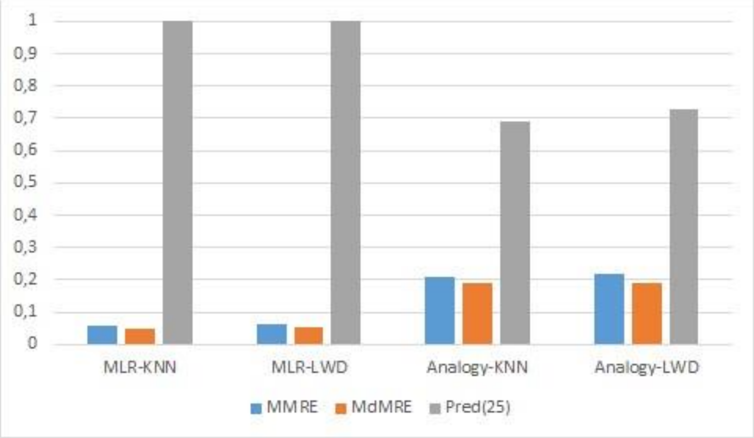

Figure 2. Accuracy Comparison of the Four Models

3.6 Model Comparison

The last stage is to verify the existence of significant differences between the models that implement KNN MD imputation with listwise deletion. Independent t-test on absolute residual mean [15] is done by comparing each of the two models, and combination of the results at significance level (α) 0.05 can be seen in Table 6. Verification of data normality has been carried out as a condition for conducting t-Test. The result shows that the absolute residual value of the four models is not normally distributed, so data transformation is done with SQRT. Table 6 shows that of the six t-tests there were four ttests that showed significant differences (p>0.05) and two tests that did not have significant differences (p<0.05). The four results of the t-test differ significantly if the method used is MLR with Analogy. Thus, MLR is outperformed the Analogy with both the KNN missing data imputation and listwise deletion.

Table 6. The Results of Model Comparison with independent t-Test

Model	*P value of t-Test*	Result
MLR-KNN MLR-LWD	0.769	Not Sig. (P > 0.05)
MLR-KNN Analogy-KNN	**0.000**	**Sig. (P < 0.05)**
Analogy-KNN Analogy-LWD	0,451	Not Sig. (P > 0.05)
MLR-LWD Analogy-LWD	**0.000**	**Sig. (P < 0.05)**
MLR-KNN Analogy-LWD	**0.000**	**Sig. (P < 0.05)**
MLR-LWD Analogy-KNN	**0.000**	**Sig. (P < 0.05)**

4. Conclusion

As discussed in this study, missing data was one of the causes of poor prediction problems. This study has implemented missing data imputation with the KNN method and listwise deletion which then compared the results using multiple linear regression and Analogy approaches. The results shows that MD imputation with MLR-KNN and MLR-LWD far outperformed Analogy-KNN and Analogy-LWD in terms of the accuracy of MMRE, MdMRE and Pred (25). This result has also statistically proven that there are significant differences between MLR-KNN approach and MLR-LWD with Analogy-KNN and Analogy-LWD. However, there are no significant difference between MLR-KNN and MLR-LWD or Analogy-KNN with Analogy-LWD. Thus, it can be concluded that missing data imputation using KNN and listwise deletion with multiple linear regression approach outperforms the Analogy approach.

This creates an interesting result since several studies of [16], [12] stated that Analogy is superior to algorithmic approaches such as linear regression with MMRE 50% average, MdMRE 28% and Pred (25) 48%. This study shows the contrasting results seen from MLR-KNN which reached 6% MMRE, 4.7% MdMRE and even Pred (25) far outperformed Analogy-KNN. Yet the results achieved by this linear regression approach are actually in line with what was expressed by [17], that if the basic assumptions of data sets have fulfilled the aspect of a strong relationship between dependent variable and independent variable, not having a significant outlier, and normally distributed, then the results of accuracy can reach the optimum and this research has fulfilled all of these assumptions.

It is recommended to conduct a further research to strengthen the results obtained using more than three data sets as well as applying other machine learning methods such as Naïve Bayes, Artificial Neural Network, SVM and Decision Tree for missing data imputation.

Acknowledgment

This research was supported by Ahmad Dahlan University's Competitive Research Grant Program (PHB-UAD) in academic year 2017/2018 with the contract number of PHB-028 / SP3 / LPPM-UAD / VI / 2018.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] J. Huang *et al.*, “Cross-validation based K nearest neighbor imputation for software quality datasets: An empirical study,” *J. Syst. Softw.*, vol. 132, pp. 226–252, Oct. 2017, doi: 10.1016/J.JSS.2017.07.012.

[2] I. Abnane and A. Idri, “Evaluating Fuzzy Analogy on incomplete software projects data,” *2016 IEEE Symp. Ser. Comput. Intell. SSCI 2016*, Feb. 2017, doi: 10.1109/SSCI.2016.7849922.

[3] I. Myrtveit, E. Stensrud, and U. H. Olsson, “Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods,” *IEEE Trans. Softw. Eng.*, vol. 27, no. 11, pp. 999–1013, Nov. 2001, doi: 10.1109/32.965340.

[4] J. Han, M. Kamber, and J. Pei, “Data Mining: Concepts and Techniques,” *Data Min. Concepts Tech.*, 2012, doi: 10.1016/C2009-0-61819-5.

[5] X. Y. Jing, F. Qi, F. Wu, and B. Xu, “Missing data imputation based on low-rank recovery and semi-supervised regression for software effort estimation,” *Proc. - Int. Conf. Softw. Eng.*, vol. 14-22-May-2016, pp. 607–618, May 2016, doi: 10.1145/2884781.2884827.

[6] A. Petrozziello and I. Jordanov, “Column-wise Guided Data Imputation,” *Procedia Comput. Sci.*, vol. 108, pp. 2282–2286, Jan. 2017, doi: 10.1016/J.PROCS.2017.05.008.

[7] W. Zhang, Y. Yang, and Q. Wang, “Using Bayesian regression and EM algorithm with missing handling for software effort prediction,” *Inf. Softw. Technol.*, vol. 58, pp. 58–70, Feb. 2015, doi: 10.1016/J.INFSOF.2014.10.005.

[8] A. Idri, I. Abnane, and A. Abran, “Missing data techniques in analogy-based software development effort estimation,” *J. Syst. Softw.*, vol. 117, pp. 595–611, Jul. 2016, doi: 10.1016/J.JSS.2016.04.058.

[9] 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.

[10] M. Shepperd and C. Schofield, “Estimating software project effort using analogies,” *IEEE Trans. Softw. Eng.*, vol. 23, no. 11, pp. 736–743, 1997, doi: 10.1109/32.637387.

[11] “(PDF) Software Effort Estimation with Multiple Linear Regression: Review and Practical Application.” https://www.researchgate.net/publication/250308495_Software_Effort_Estimation_with_Multiple_Linear_Regression_Review_and_Practical_Application (accessed Dec. 28, 2022).

[12] A. Idri, F. A. Amazal, and A. Abran, “Analogy-based software development effort estimation: A systematic mapping and review,” *Inf. Softw. Technol.*, vol. 58, pp. 206–230, Feb. 2015, doi: 10.1016/J.INFSOF.2014.07.013.

[13] E. Khatibi and V. K. Bardsiri, “An Improved Algorithmic Method for Software Development Effort Estimation,” ⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰ ⍰⍰⍰⍰⍰⍰⍰⍰ ⍰⍰ ⍰⍰⍰⍰⍰⍰⍰ ⍰⍰⍰⍰, vol. 9, no. 1, Apr. 1399, Accessed: Dec. 28, 2022. [Online]. Available: https://civilica.com/doc/1030096/certificate/print/

[14] M. Azzeh, A. B. Nassif, and S. Banitaan, "A better case adaptation method for case-based effort estimation using multi-objective optimization," *Proc. - 2014 13th Int. Conf. Mach. Learn. Appl. ICMLA 2014*, pp. 409–414, Feb. 2014, doi: 10.1109/ICMLA.2014.73.

[15] B. Kitchenham, S. L. Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," *J. Syst. Softw.*, vol. 64, no. 1, pp. 57–77, Oct. 2002, doi: 10.1016/S0164-1212(02)00021-3.

[16] E. Mendes, S. Counsell, and N. Mosley, "Measurement and Effort Prediction for Web Applications," pp. 295–310, 2001, doi: 10.1007/3-540-45144-7_28.

[17] B. Kitchenham and E. Mendes, "Why comparative effort prediction studies may be invalid," *ACM Int. Conf. Proceeding Ser.*, 2009, doi: 10.1145/1540438.1540444.