TEMATIK APERTA NUSANTARA

Nusantara Journal Of Informatics (NIJ)

Journal homepage: https://e-journal.tematikapertanusantara.sch.id/index.php/Nusantara-Journal-Of-Informatics/

REGRESSION MODEL EXPLORATION IN SOFTWARE EFFORT ESTIMATION USING THE USE CASE POINT METHOD

Anggit Wirasto [1*]

[1]Universitas Harapan Bangsa, Banyumas, Central Java, Indonesia

Coresponding Author: *anggitwirasto@uhb.ac.id*

ABSTRACT

Software development effort estimation is a crucial aspect of project management as it directly affects scheduling, resource allocation, and cost control. The Use Case Point (UCP) method is widely used for early-stage estimation; However, its traditional approach has several limitations, particularly related to subjective assessments and the tendency toward overestimation or underestimation. This study aims to explore and compare the performance of various regression models in improving the accuracy of UCP-based effort estimation. The dataset consists of 71 completed software projects, using UAW, UUCW, TCF, and ECF as predictor variables, and actual effort as the target variable. The evaluated models include Polynomial Regression, Decision Tree, Random Forest, Gradient Boosting, and Ridge Regression. Model performance was assessed using Mean Absolute Error (MAE), Mean Balanced Relative Error (MBRE), and Mean Inverted Balanced Relative Error (MIBRE) with an 80:20 train–test split. The experimental results indicate that the optimized Random Forest model achieves the best balance between training accuracy and generalization ability on unseen data (test MAE of 11.38), significantly outperforming the traditional UCP calculation method (MAE of 90.33). These findings suggest that non-linear regression approaches, particularly ensemble-based methods, can substantially enhance the reliability of software effort estimation compared to the conventional UCP method.

ARTICLEINFO

Keywords:

Software Business Estimation, Use Case Point, Regression Model

1. Introduction

Amidst the ongoing digital transformation, software occupies a central position in various segments of modern life. This strategic role demands accelerated software development that is not only efficient but also responsive to dynamic market needs. Therefore, the ability to accurately estimate development effort, enabling decision-makers to allocate resources appropriately and plan development cycles more effectively, is crucial.

Effort estimation in software development is a fundamental process in project management, essential for planning, resource allocation, and project budgeting. The accuracy of effort estimates significantly impacts strategic and operational decisions, ultimately determining the success or failure of a software project. For example, overly optimistic estimates can lead to time and resource shortages, while pessimistic estimates can lead to project rejection due to uncompetitive costs [1].

Traditional estimation methods such as Function Point Analysis have been used since the 1970s to assess the effort required in software development. Although these methods have become industry standards, they are often not flexible enough to adapt to new technologies and rapidly changing development methodologies, such as Agile and DevOps development [2] –[6].

The Use Case Point (UCP) method, developed in the 1990s [7], is a technique that attempts to address some of the limitations of the Function Points technique. UCP measures the complexity of a software project by considering the use cases and actors involved, allowing for more dynamic and contextual estimation [8].

However, UCP also presents its own challenges. Subjective assessments in classifying complexity and interpreting technical and environmental adjustment factors can lead to inconsistencies and inaccuracies in effort estimates. Other challenges include a lack of sufficient public industry data, and potential overestimation and underestimation [1], [9] –[11]. The success of UCP applications depends heavily on the development team's expertise and experience in accurately analyzing and estimating use cases [12].

This study explores various regression methods to determine their effectiveness compared to traditional UCP calculation models. Regression analysis is expected to help identify areas where the UCP method may be less effective and provide insights for continuous improvement. This allows for adjustments to the UCP methodology based on feedback from previous projects, increasing its reliability and effectiveness as an estimation tool.

2. Method

2.1. Experimental Design

This experiment was designed to compare the software effort estimation performance between the traditional UCP method and various regression methods. The research design is presented in Figure 2 below.

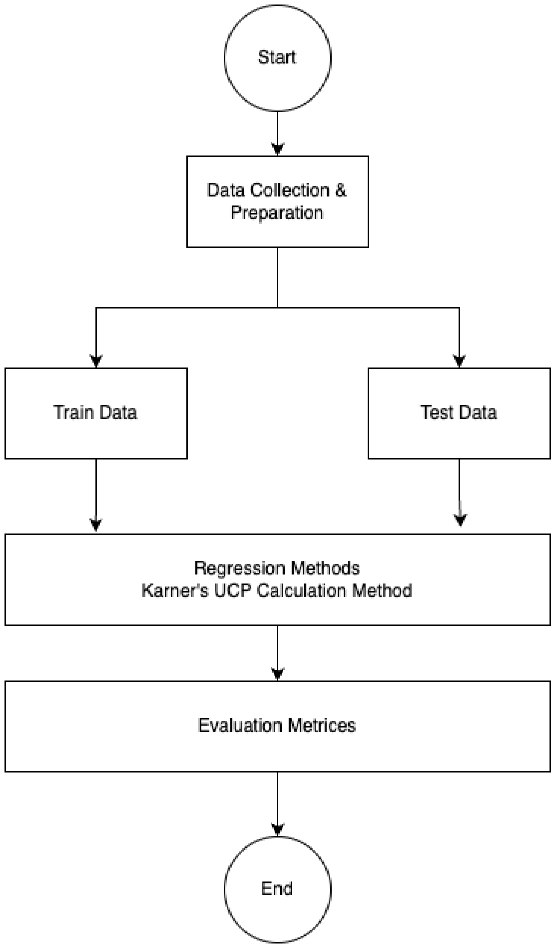

Figure 1. Research design

2.2. Data Collection and Preparation

Historical data from completed software projects will be collected from a software development company database. The data will include complete details of the use cases, including actors, interactions, and technical and environmental components relevant to UCP.

The dataset used is data from 71 software development projects collected by Silhavy [15]. This data contains eighteen variables: Project No., Simple Actors, Average Actors, Complex Actors, UAW, Simple UC, Average UC, Complex UC, UUCW, TCF, ECF, Real_P20, Real_Effort , Sector, Language, Methodology, Application Type, and DataDonator. Of these 18 variables, only the related variables will be used: UAW, UUCW, TCF, ECF as independent variables, and Real_P20 as dependent variable. The data is divided into training and test datasets.

2.3. Implementation of Regression Method

Various regression methods are then implemented on the training data. Methods that allow for parameter optimization will search for the best parameter combination. The regression methods used are polynomial regression, decision tree regression, random forest regression, gradient boosting regression, and ridge regression.

2.4. Testing Procedures

The optimal model obtained from each regression model is then tested on a test dataset. This is done to determine the model's performance on previously unknown data and thus to identify the model's tendency to overfit.

2.5. Evaluation and Interpretation of Results

The results are evaluated based on the Mean Absolute Error (MAE), Mean Balanced Relative Error (MBRE), and Mean Inverted Balanced Relative Error (MIBRE) criteria. MAE measures the average absolute error between the predictions and the actual values, providing an indication of how close the predictions are to the actual values.

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

MBRE is a metric used to measure the accuracy of predictions in software effort estimation, taking into account both overestimation and underestimation. The formula for MBRE is as follows:

$$MBRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{\min(\hat{y}_i, y_i)} \tag{2}$$

MIBRE is a metric used to measure the performance of an estimation model by considering the model's tendency to overestimate or underestimate.

$$MIBRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{\max(\hat{y}_i, y_i)} \tag{3}$$

2.6. Research Limitations

This study has several limitations, including the use of historical data from a dataset that may not reflect the current state of software development. Other limitations include potential bias in dataset selection and potential overfitting in model optimization.

3. Results and Discussion

Study This aim For modeling and predicting Use Case Points (UCP) in context development device soft . In the study This done a series experiment regression using a dataset consisting of from features such as Unadjusted Actor Weight (UAW), Technical Complexity Factor (TCF), Environmental Complexity Factor (ECF), and Unadjusted Use Case Weight (UUCW). Experiment This involving use various technique regression , start from a simple linear model to more ensemble models complex , with objective For find the best model that can generalize with good on data that is not seen previously .

3.1. Initial Review

Experiment done against a dataset containing 71 project data development device soft . For needs validate this dataset shared into training data sets and test data sets with 80:20 ratio or a total of 57 training data and 14 test data. Use approach This allows For train a model on a single data set and then test how much both models working on unprocessed data Once seen previously . With thus trend *overfitting* each model can seen .

The performance of each model is compared with mark *actual effort* and measured use MAE, MBRE, and MIBRE values . In the UCP calculation , the values *estimated effort* obtained with multiplying UCP and factor productivity which is constant with value 20. For the sake of simplification calculation comparison done to UCP value and value *actual effort* divided by 20.

3.2. Regression Polynomial

Analysis First done with Regression model Polynomial of degree-3. Although this model in a way theory capable catch non-linear relationship between features and targets, the results shows significant overfitting . This model fits very well with training data but fail generalizing to the test data, it is seen from Mean Absolute Error (MAE) for training data of 22.46 and a very high value in the test data of 394.27. This possible Because excessive model complexity the height that makes it sensitive to variation small in training data .

3.3. Decision Tree

Furthermore done exploration Use of Decision Tree Regression. Initial results shows perfect fit on the training data with MAE value 0.0, but performance poor on test data with an MAE of . 12.12, indicating overfitting. Through parameter adjustments , namely depth maximum tree changed to 5 and the minimum sample split is set of 4, overfitting is successful A little reduced . The MAE value after parameter adjustment is of 3.58 on training data and 11.94 on test data.

3.4. Random Forest

Furthermore done analysis with Random Forest Regression method , a ensemble method that combines prediction from many decision trees. In In general , Random Forest shows more balance Good between accuracy on training data and capability generalization on test data, shown with MAE value is 4.55 on training data and 11.20 on test data.

Then done more parameter optimization carry on For increase performance of this model . The best parameters found through the optimization process is the max depth with value 7, min samples split with value 2, and number of estimators with value 200. Obtained MAE value of training data of 3.41 and test data of 11.38. Optimization This show little performance more Good compared to with the previous model . However , it still There is indication of overfitting, although in larger scale small .

3.5. Gradient Boosting

The next method analyzed is Gradient Boosting Regression, with potential similar to the Random Forest model. This model show impressive performance on training data but experience challenge in generalization . Adjustment of parameters such as reduce rate learning and improving amount tree give marginal improvements that do not significant on ability model generalization .

3.6. Ridge Regression

Exploration final done with Ridge Regression, where this model produce warning about the "ill-conditioned matrix", which indicates problem with data conditions such as scale features that are not balanced or highly correlated features . Although parameter optimization through RandomizedSearchCV give improvement in performance , fundamental problems in data are not resolved completely .

3.7. Discussion

Evaluation results each regression model summarized in

Table 1All models explored show excellent performance on training data , with relative MAE value small . The result of Decision Tree and Gradient Boosting methods even give values 0.00 and 0.48 respectively .

However very small MAE value especially on training data Can So is indication existence *overfitting* . This is happen when the regression model too perfect in studying training data until to the specific details , including noise or fluctuations random , which is not represent pattern general in the data . As a result, the model fails generalize with good on the test data. This proven by the MAE value of each model when the test data has mark relatively more big from MAE value of training data .

Worst performance obtained in the regression model polynomial , with MAE value of remote test data more big from other models that is amounting to 394.27. This is due to polynomial models , especially those that have high degree tend studying training data in detail, and able to catch a very complex relationship between variables . Models that are too complex tend flexible and adaptable with every data points in the training data , which causes performance bad on data that is not seen previously .

The Ridge Regression model provides MAE value of 20.53 on the test data, far more small from the regression model polynomial However relatively more tall from other models that provide The MAE value of the test data is in the range of 11 to 12. Ridge Regression is a linear regression model that uses L2 regularization , which adds punishment on big coefficient For prevent *overfitting* . However mark lack of regularization appropriate make the model not Enough flexible For catch existing patterns in data.

In addition , the Ridge Regression model assumes linear relationship between features and target variables . If the actual data own non-linear relationship or complex , Ridge Regression models may be No capable catch connection the with effective . Warning *ill-conditioned matrix* during the regression process show existence problem with data, such as highly correlated features or scale features that are not balanced .

Based on summary results MAE value for each model that has been tested , Random Forest Regression was the best model with show good balance between accuracy on training data and capability For generalize to test data, especially after the parameter optimization process . Random Forest model, with his approach combines Lots tree decisions , tend to more stand to *overfitting* compared to with a single model like Decision Tree, and also capable of catch data complexity with more Good compared to with linear models such as Ridge Regression.

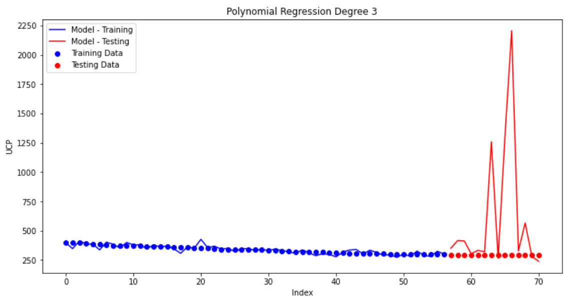

Figure 2. Regression model polynomial

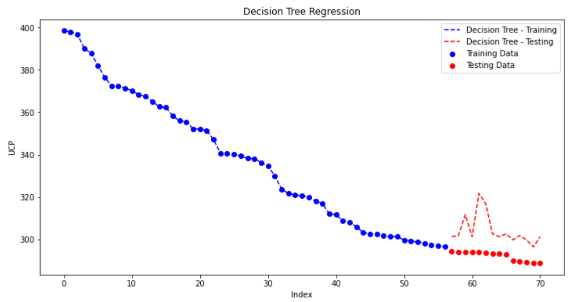

Figure 3. Regression model tree decision

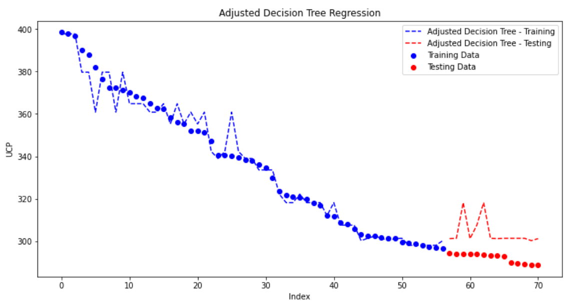

Figure 4. Regression model tree decision with parameter optimization

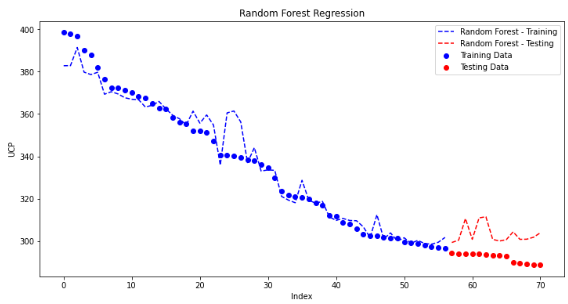

Figure 5. Random forest regression model

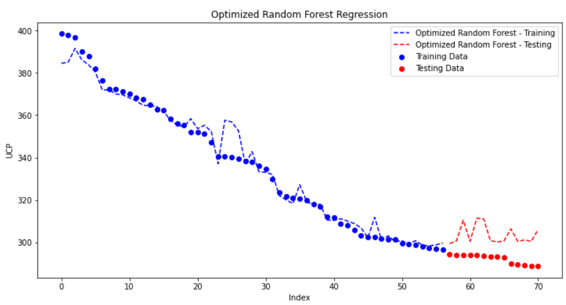

Figure 6. Optimized random forest regression model

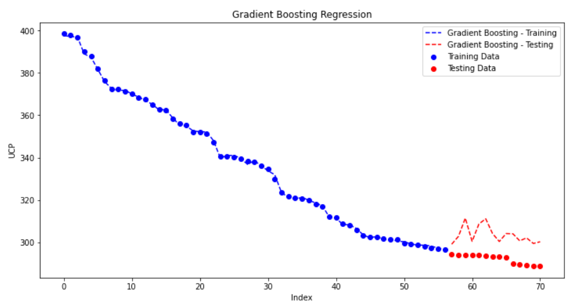

Figure 7. Gradient boosting regression model

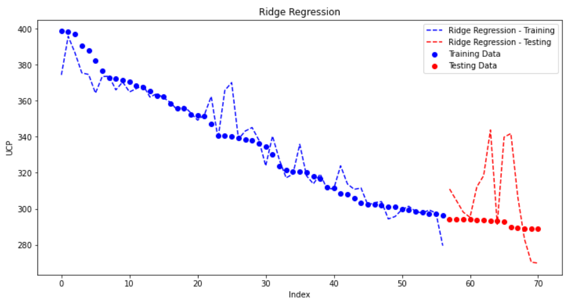

Figure 8. Ridge regression model

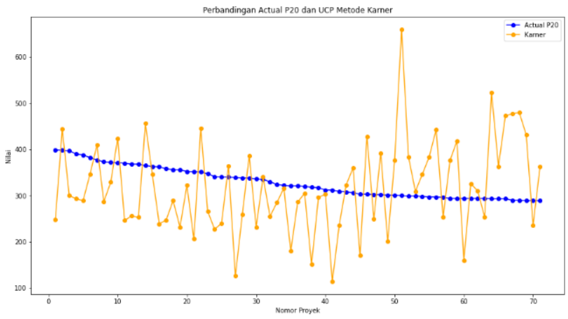

Figure 9. Comparison UCP value of Karner's calculation of actual effort value divided by 20

Table 1Metrics evaluation each model

Model	MAE (train)	MAE (test)	MBRE (train)	MBRE (test)	MIBRE (train)	MIBRE (test)
Polynomial Regression	22.46	394.27	0.0670	1.3530	0.0680	0.3310
Decision Tree	0.00	12.12	0.0000	0.0415	0.0000	0.0396
Decision Tree (adjusted)	3.58	11.94	0.0102	0.0409	0.0102	0.0390
Random Forest	4.55	11.20	0.0131	0.0384	0.0130	0.0368
Random Forest (optimized)	3.41	11.38	0.0099	0.0390	0.0098	0.0373
Gradient Boosting	0.48	11.34	0.0014	0.0388	0.0014	0.0372
Ridge Regression	5.80	20.53	0.0170	0.0703	0.0169	0.0644
Karner	90.33	-	0.2799	-	0.3327	-

4. Conclusion

Of the various regression models that have been tested can concluded the Random Forest model that has been optimized show sufficient balance Good between accuracy on training data and test data with MAE values of 3.41 and 11.38, MBRE of 0.0099 and 0.0390, and MIBRE of 0.0098 and 0.0373 respectively For each training data and test data.

However in a way overall regression model show far- reaching results more Good when juxtaposed with method traditional UCP calculations that have MAE, MBRE, and MIBRE values were 90.33, 0.2799, and 0.3327, respectively. The research furthermore can done optimization more continue on the Random Forest Regression model for see its performance on different datasets .

References

[1] M. Azzeh, A. Bou Nassif, and IB Attili, "Predicting software effort from use case points: A systematic review," *Science of Computer Programming* , vol. 204. 2021. doi: 10.1016/j.scico.2020.102596.

[2] M. Azzeh, AB Nassif, and CL Martín, "Empirical analysis on productivity prediction and locality for use case points method," *Softw. Qual. J.* , vol. 29, no. 2, pp. 309–336, 2021, doi: 10.1007/s11219-021-09547-0.

[3] N. Govil and A. Sharma, "Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors," *Adv. Eng. Softw.* , vol. 172, p.

103209, Oct. 2022, doi: 10.1016/J.ADVENGSOFT.2022.103209.

[4] D. De Silva, S. Vidanarachchi, P. Wilagama, R. Gunaratne, H. Hewa, and D. Nas, *Software Complexity Metrics for Emerging Technologies: Challenges and Opportunities* . 2023.

[5] S. Sharma and S. Vijayvargiya, "Modeling of software project effort estimation: a comparative performance evaluation of optimized soft computing-based methods," *Int. J. Inf. Technol.* , vol. 14, no. 5, pp. 2487–2496, 2022, doi: 10.1007/s41870-022-00962-5.

[6] J. Leong, K. May Yee, O. Baitsegi, L. Palanisamy, and RK Ramasamy, "Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability," *Sustainability* , vol. 15, no. 2, p. 1121, Jan. 2023, doi: 10.3390/su15021121.

[7] G. Karner, "Resource estimation for objectory projects," *Object. Syst. SF AB* , pp. 1–9, 1993, Accessed: Dec. 03, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=17b5f04743cd13f6077fbdec227719e5d83dba10

[8] S. Kusumoto, F. Matukawa, K. Inoue, S. Hanabusa, and Y. Maegawa, "Estimating effort by use case points: method, tool and case study," in *10th International Symposium on Software Metrics, 2004. Proceedings.* , 2004, pp. 292–299. doi: 10.1109/METRIC.2004.1357913.

[9] Y. Mahmood, N. Kama, and A. Azmi, "A systematic review of studies on use case points and expert-based estimation of software development effort," *J. Softw. Evol. Process* , vol. 32, no. 7, p. e2245, Jul. 2020, doi: 10.1002/SMR.2245.

[10] S. Shukla and S. Kumar, "Know-UCP: locally weighted linear regression based approach for UCP estimation," *Appl. Intel.* , vol. 53, no. 11, pp. 13488–13505, Oct. 2022, doi: 10.1007/S10489-022-04160-5.

[11] S. Shukla and S. Kumar, "Towards ensemble-based use case point prediction," *Softw. Qual. J.* , vol. 31, no. 3, pp. 843–864, Sept. 2023, doi: 10.1007/S11219-022-09612-2/METRICS.

[12] M. Sultan, T. Tawill, M. Hagal, and S. Elakeili, "Improving Use Case Points Techniques to Support Size Estimation of Software Projects," in *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)* , 2022, pp. 267–272. doi: 10.1109/MI-STA54861.2022.9837735.

[13] Ardiansyah, R. Ferdiana, and AE Permanasari, "Use case points based software effort prediction using regression analysis," *2019 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2019* , pp. 15–20, Oct. 2019, doi: 10.1109/ICACSIS47736.2019.8979851.

[14] S. Shukla and S. Kumar, "Towards non-linear regression-based prediction of use case point (UCP) metrics," *Appl. Intel.* , vol. 53, no. 9, pp. 10326–10339, May 2023, doi: 10.1007/S10489-022-04002-4/METRICS.

[15] R. Silhavy, P. Silhavy, and Z. Prokopova, "Analysis and selection of a regression model for the Use Case Points method using a stepwise approach," *J. Syst. Softw.* , vol. 125, pp. 1–14, March. 2017, doi: 10.1016/J.JSS.2016.11.029.

[16] A. Ardiansyah, R. Ferdiana, and A. Permanasari, "Optimizing complexity weight parameters of use case points estimation using particle swarm optimization," *Int. J. Adv. Intel. Informatics* , vol. 8, p. 165, Jul. 2022, doi: 10.26555/ijain.v8i2.811.

[17] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays," *IEEE Access* , vol. 9, pp. 26926–26936, 2021, doi: 10.1109/ACCESS.2021.3057807.

[18] HT Hoc, V. Van Hai, and H. Le Thi Kim Nhung, "AdamOptimizer for the Optimization of Use Case Points Estimation BT - Software Engineering Perspectives in Intelligent Systems," 2020, pp. 747–756.

[19] Yulianeu, A. (2017). Aplikasi perhitungan stok barang di waserda koperasi unit desa minarasa batukaras dengan menggunakan metode average. JUTEKIN (Jurnal Teknik Informatika), 1(2).

[20] Yulianeu, A., & Jakaria, D. A. (2017). Sistem Informasi Manajemen Keamanan Rukun Warga (Suatu Kajian Ilmiah yang diterapkan di RW 03 Kampung Babakan Talang Desa Cimari Kecamatan Cikoneng Ciamis). JUMIKA (Jurnal Manajemen Informatika), 1(1).