CENTIVE
Conference on Electrical Engineering, Informatics
, Industrial Technology, and Creative Media
ISBN : 978-602-53004-0-0

Applying Consistent Fuzzy Preference Relation in Weighting Software Effort Estimation Criteria

Ika Indah Lestari[1*], Adnan Purwanto[2], Sulistiyasni[3], and Singgih Briandoko[4]
[1,2,3,4]STMIK Widya Utama Purwokerto
[1,2,3,4]Jalan Sunan Kalijaga, Berkoh, Purwokerto Selatan 53146, Indonesia
[*]ikaindah22@swu.ac.id

Abstract — Software effort estimation (SEE) is a critical process in project planning, as it determines budget allocation, resource management, and timeline accuracy. The weighting of estimation criteria significantly influences the reliability of the estimation model. This study aims to determine the weights of SEE criteria using a fuzzy logic approach, specifically the Consistent Fuzzy Preference Relation (CFPR) method. As a Multi-Criteria Decision Making (MCDM) technique, CFPR offers an efficient mechanism for extracting consistent expert preferences by requiring only n-1 pairwise comparisons from n criteria, making it suitable for rapid weighting calculations. The study evaluates four main attributes: Product, Computer, Personnel, and Project. Expert assessments were conducted using crisp numbers on a 1-9 scale. The results show the following attribute weights: Product (0.372), Computer (0.275), Personnel (0.231), and Project (0.122). Furthermore, the top three ranked cost drivers are Required Reliability (0.1674), Product Complexity (0.1384), and Execution Time Constraint (0.1050). Conversely, the lowest weights were assigned to Programming Language Experience (0.0270), Virtual Machine Experience (0.0296), and Required Development Schedule (0.0303). The integration of CFPR into SEE models produces a stable and interpretable weight distribution, thereby enhancing the accuracy of effort estimation.

Keywords – Consistent fuzzy preference relation, multi-criteria decision making, project planning, software effort estimation, weighting criteria.

I. INTRODUCTION

The development of relevant and practical models remains a significant challenge in software engineering. A critical aspect of this challenge is to achieve an accurate estimate of the effort, which constitutes a fundamental process for predicting the time and cost required for software development. Accurate estimates are vital as overestimation can lead to unnecessary financial losses, while underestimation often results in compromised software quality. In the early stages of a project, reliable effort estimation provides crucial support to project managers in planning and decision-making [1], even when information is scarce and uncertain.

As an essential component of project management, software effort estimation (SEE) involves predicting the resources needed for major development or maintenance activities based on historical data. Accurate predictions allow companies and clients to effectively classify, prioritize, and allocate resources for a project [2], [3]. In response to this need, researchers and practitioners have extensively investigated SEE issues and proposed numerous estimation methods [4]–[6]. Although many of these developed models are suitable for specific development environments, frequent changes in technologies and user requirements continually exacerbate the challenge of producing accurate estimates. Although various approaches have been attempted to improve the accuracy of the prediction, no single technique has been consistently successful [5]. Consequently, some researchers have turned to hybrid methods that combine multiple techniques to mitigate the inaccuracy of individual models. A primary source of erroneous estimates is the use of incomplete, sparse, inconsistent, or poorly documented datasets from previous projects. Furthermore, the estimation process is influenced by a multitude of factors, both observable and latent [7], [8].

Recently, multi-criteria decision-making (MCDM) methods have emerged as a promising approach for addressing the multifactorial nature of effort estimation [9], [10]. These methods can be further enhanced through integration with machine learning techniques to improve the performance of estimation models. Among the established models, the Constructive Cost

Table 1. Cost drivers and attributes

ATTRIBUTES	NOTATION	COST DRIVERS
A. PRODUCT ATTRIBUTES	A1	**RELY** - Required Reliability
	A2	**DATA** - Database size
	A3	**CPLX** - Product Complexity
B. COMPUTER ATTRIBUTES	B1	**TIME** – Execution time constraint
	B2	**STOR** – Main storage constraint
	B3	**VIRT** – Virtual machine volatility
	B4	**TURN** – Computer turnaround time
C. PERSONEL ATTRIBUTES	C1	**ACAP** – Analyst capability
	C2	**AEXP** – Application experience
	C3	**PCAP** – Programmer capability
	C4	**VEXP** – Virtual Machine experience
	C5	**LEXP** – Programming language experience
D. PROJECT ATTRIBUTES	D1	**MODP** – Modern programming practices
	D2	**TOOL** – Use of software tools
	D3	**SCED** - Required development schedule

Model (COCOMO) is a well-known regression-based approach that uses lines of code (LOC) as its primary input [11]–[13]. Its extended version, Intermediate COCOMO, incorporates cost drivers to refine its estimates. These 15 cost drivers are categorized into four attributes, as presented in Table 1.

II. RESEARCH METHOD

This study employs the Consistent Fuzzy Preference Relation (CFPR) as a Multi-Criteria Decision Making (MCDM) method. According to [14], CFPR requires only $n-1$ pairwise comparisons for n criteria. A key advantage of using CFPR is its ability to significantly reduce the number of comparisons between cost drivers. In contrast, other methods, such as the Analytic Hierarchy Process (AHP), typically require $\frac{n(n-1)}{2}$ comparisons [15]. The procedural steps of this study are as follows:

a) Identifying the research problem and constructing the hierarchical structure.
b) Developing a pairwise comparison matrix for cost drivers and attributes using the CFPR method.
c) Determining the score and weight of each cost driver
d) Interpreting the results.

The pairwise comparison matrix for the cost drivers and criteria was established based on assessments from expert respondents. The relationships between elements can be represented as a relation R on the set of cost drivers A, where

$$R \subseteq A \times A,\ R = (r_{ij}), \tag{1}$$

$\forall i, j \in \{1, 2, 3, \ldots, n\}$ where A is the set of cost drivers or the criteria, r_{ij} is the importance ratio of the cost drivers between a_i and $a_j, a_{ij} \times a_{ji} = 1, \forall i, j \in \{1, 2, 3, \ldots, n\}$. Suppose that A is a pairwise comparison matrix in CFPR method,

$$A = \begin{pmatrix} 1 & a_{12} & \ldots & z \\ z & 1 & \ldots & z \\ \vdots & \vdots & \ddots & a_{n-1,n-1} \\ z & z & \ldots & 1 \end{pmatrix} \tag{2}$$

The main diagonal fills 1, while the expert assessment fills in a_{ij} above the main diagonal, while z can be calculated using proposition 1 and proposition 2 [16]–[19]. A Fuzzy preference relation P is represented by the $n \times n$ matrix $P = (p_{ij})$ with correlations with the matrix A, $p_{ij} = g(a_{ij}), \forall i, j \in \{1, 2, 3, \ldots, n\}$.

$$P = \begin{pmatrix} 1 & p_{12} & p_{13} & p_{14} \\ p_{21} & 1 & p_{23} & p_{24} \\ p_{31} & p_{32} & 1 & p_{34} \\ p_{41} & p_{42} & p_{43} & 1 \end{pmatrix} \tag{3}$$

Proposition 1: Consider a set of alternatives $A = \{a_1, a_2, ..., a_n\}$ and associated with a reciprocal multiplicate preference relation $A = (a_{ij})$ with $a_{ij} \in [\frac{1}{9}, 9]$. Then the corresponding reciprocal fuzzy preference relation $P = (p_{ij})$ with $p_{ij} = [0, 1]$ is given as:

$$p_{ij} = g(a_{ij}) = \frac{1}{2}(1 + \log_9\ a_{ij}) \tag{4}$$

where $\log_9(a_{ij})$ is considered because a_{ij} is between $1/9$ and 9. In general, if a_{ij} is between $1/n$ and n, then $\log_n\ a_{ij}$ is used.

Proposition 2: For a reciprocal fuzzy preference relation $P = (p_{ij})$, the following statements are equivalent:

$$p_{ij} + p_{jk} + p_{ki} = \frac{3}{2}, \quad \forall i < j < k \tag{5}$$

$$p_{i(i+1)} + p_{(i+1)(i+2)} + \cdots + p_{(i+k)i} = \frac{k+1}{2}, \quad \forall i < j \tag{6}$$

A decision matrix with entries that do not fall in the interval $[0, 1]$, but fall in an interval $[-k, 1+k]$, $k > 0$, can be obtained by transforming the calculated values using a transform function. The transform function is defined as:

$$f : [-k, 1+k] \to [0, 1],\ f(p) = \frac{p+k}{1+2k} \tag{7}$$

The weight of each criterion (w_{ij}) was calculated using equations (5) and (6) where n_c is the number of criteria and p_{ij} is the value in the row i and column j .

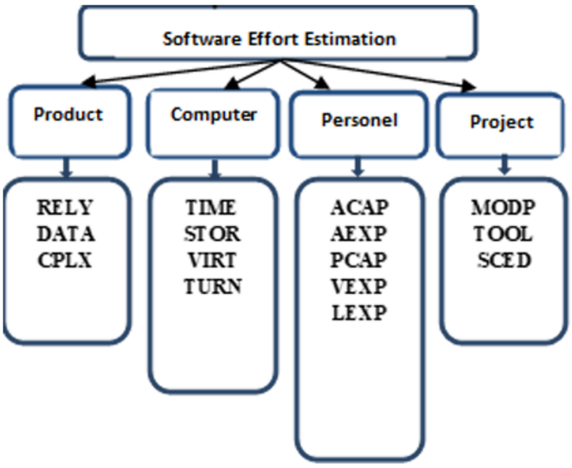

Fig. 1. The Hierarchical structure of Software Effort Estimation.

$$p_{ij} + p_{jk} + p_{ki} = \frac{3}{2}, \quad \forall i < j < k \tag{8}$$

$$p_{i(i+1)} + p_{(i+1)(i+2)} + \cdots + p_{(i+k)i} = \frac{k+1}{2}, \quad \forall i < j \tag{9}$$

Finally, the results obtained from the CFPR method for calculating the weights of the software effort estimation criteria were interpreted using descriptive analysis [20].

III. RESULT

plication of the Consistent Fuzzy Preference Relation (CFPR) method to determine the weights of cost drivers in Software Effort Estimation (SEE). The first step involved constructing the hierarchical structure, followed by developing a pairwise comparison matrix based on expert judgments. The experts involved in this study consisted of four academics from four different universities and one practitioner from an IT company. The pairwise comparison matrix derived from their judgments is presented in Table 3.

A. The Hierarchical Structure of Software Effort Estimation

The hierarchical structure was systematically established, organizing the 15 cost drivers into four main attributes: Product, Computer, Personnel, and Project. This structure is illustrated in Fig. 1.

B. Expert Judgement

Input values for the pairwise comparisons were obtained from the five experts mentioned previously. The assessment values, expressed as crisp numbers on a scale from 1 to 9 (as defined in Table 2), were averaged to form the final comparison matrix. This aggregated input ensures a balanced and representative reflection of expert opinion.

Meanwhile, the questionnaire form filled out by the expert is a comparative value of 2 cost drivers as shown

Table 2. Linguistic Scale

Crisp Numbers	Definition
1	Equal Importance – Two elements contribute equally to the objective.
2	Intermediate Value (Between Equal and Moderate Importance)
3	Moderate Importance – Experience and judgment slightly favor one element over another.
4	Intermediate Value (Between Moderate and Strong Importance)
5	Strong Importance – Experience and judgment strongly favor one element over another.
6	Intermediate Value (Between Strong and Very Strong Importance)
7	Very Strong Importance – One element is favored very strongly; its dominance is demonstrated in practice.
8	Intermediate Value (Between Very Strong and Extreme Importance)
9	Extreme Importance – The evidence favoring one element over another is of the highest possible order of affirmation

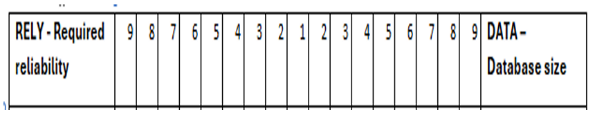

Fig. 2. Questionaire form for Expert Judgement.

below in Fig. 2. There are $n-1$ comparison cost drivers, where n is the cost drivers.

C. Comparison Matrix

The second step is building comparison matrix based on expert judgement. The pairwise somparison matrix shown in Table 3, which the value representation has been given Table 2.

Herein, next calculation can be demonstrated as follows:

$$\begin{aligned}
p_{12} &= \log_9(3.48) = 0.784, \\
p_{23} &= \log_9(1.76) = 0.629, \\
p_{34} &= \log_9(4.05) = 0.818, \\
p_{21} &= 1 - p_{12} = 1 - 0.784 = 0.216, \\
p_{32} &= 1 - p_{23} = 1 - 0.629 = 0.371, \\
p_{43} &= 1 - p_{34} = 1 - 0.818 = 0.182, \\
p_{31} &= 1.5 - p_{12} - p_{23} = 0.088, \\
p_{42} &= 1.5 - p_{23} - p_{34} = 0.053, \\
p_{41} &= 2 - p_{12} - p_{23} - p_{34} = -0.230, \\
p_{13} &= 1 - p_{31} = 1 - 0.088 = 0.912, \\
p_{14} &= 1 - p_{41} = 1 - (-0.230) = 1.230, \\
p_{24} &= 1 - p_{42} = 1 - 0.053 = 0.947.
\end{aligned}$$

In the calculation above, there is p between interval $[-k, 1+k]$, $k > 0$, where the value is not in interval $[0, 1]$. Therefore, equation (4) will be used to normalized so that the value in interval $[0, 1]$. It can be seen $p_{14} = 1.230$ and $p_{41} = -0.230$, then using equation (4) which $k = 0.230$,

Table 3. The Pairwise Comparison matrix for criteria

Attributes	**A**	**B**	**C**	**D**
A	1	3.48	p_{13}	p_{14}
B	p_{21}	1	1.76	p_{24}
C	p_{31}	p_{32}	1	4.05
D	p_{41}	p_{42}	p_{43}	1

Table 4. The initial decision matrix of the criteria

Attributes	**A**	**B**	**C**	**D**
A	0.5	0.694	z	z
B	z	0.5	0.588	z
C	z	z	0.5	0.718
D	z	z	z	0.5

$$f(p_{12}) = f(0.784) = \frac{0.784 + 0.230}{1 + (2 \times 0.230)} = 0.694 \quad (10)$$

The others element is calculated with same method.

D. *The Initial Decision Matrix*

Herein, construct the initial criteria decision matrix. The initial preference ratio calculated according to preposition 1. It can be seen in Table 4 until Table 8. Whereas the others value initialized z.

E. *The Complete Decision Matrix*

The full transformation matrix which has been calculated can be seen in Table 9 – Table 13. Its shows the weight and rank of each cost drivers.

The weight of each cost driver for each attribute is presented in Fig. 3 – Fig. 7. Therefore,can be seen the cost drivers that have most dominant in each attribute.

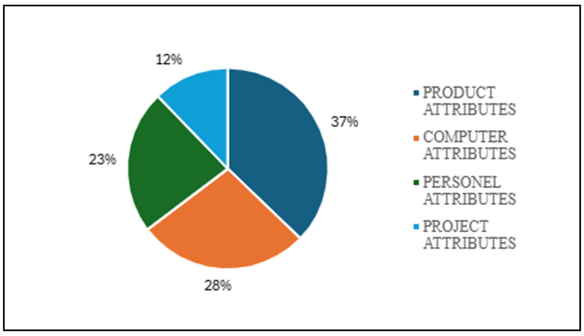

Fig. 3. Attribute Weight Percentage.

In Fig. 3, can be seen that the most considered attribute in software effort estimation is the product attribute 37%.

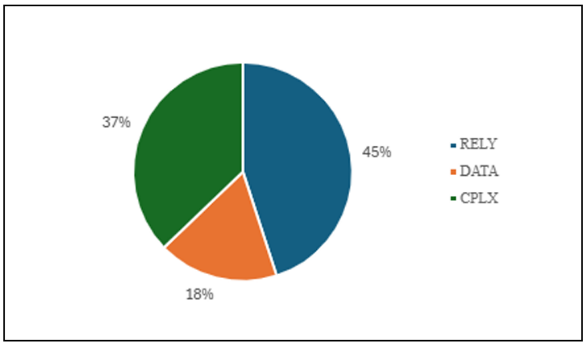

Fig. 4. Product Attributes Weight Percentage.

Table 5. The initial decision matrix of the Product Attributes

Cost Driver	**A1**	**A2**	**A3**
A1	0.5	0.908	z
A2	z	0.5	0.21
A3	z	z	0.5

Table 6. The initial decision matrix of the Computer Attributes

Cost Driver	**B1**	**B2**	**B3**	**B4**
B1	0.5	0.866	z	z
B2	z	0.5	0.608	z
B3	z	z	0.5	0.241
B4	z	z	z	0.5

Figure 4 shown that RELY (Requiered reliability) which 45% is dominant factor in produst attributes beside DATA (database size) and CPLX (product complexity).

Figure 5 shown that TIME (Execution time constrain) with percentage 38% have highest priority factor in computer attributes.

Figure 6 shown that Analyst capability with percentage 31% have highest priority factor in personel attributes.

Figure 7 shown that Modern Programming practices with percentage 47% have highest priority factor in project attributes.

IV. DISCUSSION

This study successfully determined the precise weights of cost drivers for Software Effort Estimation (SEE) by applying the Consistent Fuzzy Preference Relation (CFPR) method. The final weight for each individual cost driver (Wi) was calculated as the product of the weight of its parent attribute and its local weight within that attribute, as defined by the equation:

$$W_i = w_i(\text{attributes}) \times w_i(\text{the cost driver}) \quad (11)$$

The comprehensive results, including the calculated weights and the subsequent ranking of all cost drivers, are presented in Table 14.

A primary application of these CFPR-derived weights is their use as a reliable Effort Adjustment Factor (EAF) within established SEE models. The authors propose that these weights can be integrated into historical datasets, such as the NASA or the COCOMO dataset, to enhance their accuracy. Specifically, the

Table 7. The initial decision matrix of the Personel Attributes

Cost Driver	**C1**	**C2**	**C3**	**C4**	**C5**
C1	0.5	0.667	z	z	z
C2	z	0.5	0.653	z	z
C3	z	z	0.5	0.65	z
C4	z	z	z	0.5	z
C5	z	z	z	z	0.5

Table 8. The initial decision matrix of the Project Attributes

Cost Driver	**D1**	**D2**	**D3**
D1	0.5	0.795	z
D2	z	0.5	0.545
D3	z	z	0.5

Table 9. The complete decision matrix for Attributes

Cost Driver	A	B	C	D	Weight	Rank
A	0,5	0,694	0,782	1	0,372	1
B	0,306	0,5	0,588	0,806	0,275	2
C	0,218	0,412	0,5	0,718	0,231	3
D	0	0,194	0,282	0,5	0,122	4

Table 10. The complete decision matrix for Product Attributes

Cost Driver	A1	A2	A3	Weight	Rank
A1	0,5	0,908	0,618	0,45	1
A2	0,092	0,5	0,21	0,178	3
A3	0,382	0,79	0,5	0,372	2

Table 11. The complete decision matrix for Computer Atributes

Cost Driver	B1	B2	B3	B4	Weight	Rank
B1	0,5	0,866	0,975	0,715	0,382	1
B2	0,134	0,5	0,608	0,349	0,199	3
B3	0,025	0,392	0,5	0,241	0,145	4
B4	0,285	0,651	0,759	0,5	0,274	2

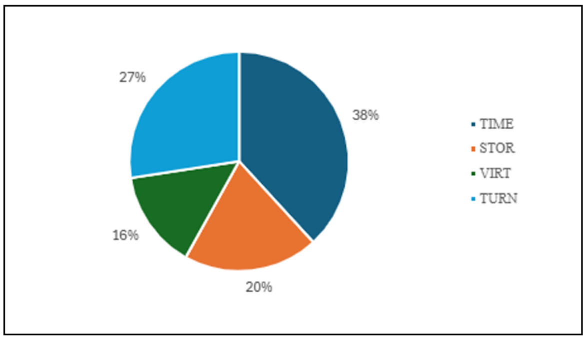

Fig. 5. Computer Attributes Weight Percentage.

Table 12. The complete decision matrix for Personel Atributes

Cost Driver	C1	C2	C3	C4	C5	Weight	Rank
C1	0,5	0,667	0,82	0,97	1	0,317	1
C2	0,33	0,5	0,653	0,803	0,83	0,25	2
C3	0,18	0,347	0,5	0,65	0,68	0,188	3
C4	0,03	0,197	0,35	0,5	0.53	0,128	4
C5	0	0,167	0,32	0,47	0,5	0,117	5

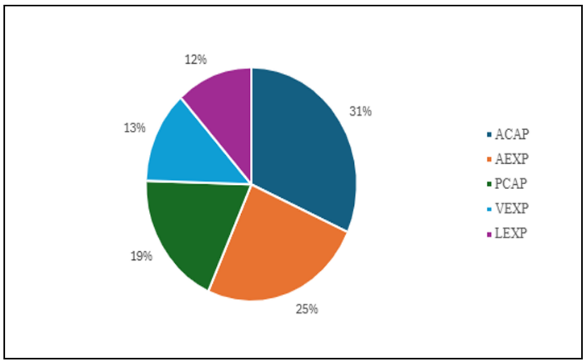

Fig. 6. Personel Attributes Weight Percentage.

Table 13. The complete decision matrix for Project Attributes

Cost Driver	D1	D2	D3	Weight	Rank
D1	0.5	0,795	0,84	0,474	1
D2	0.205	0,5	0,545	0,278	2
D3	0.16	0,455	0,5	0,248	3

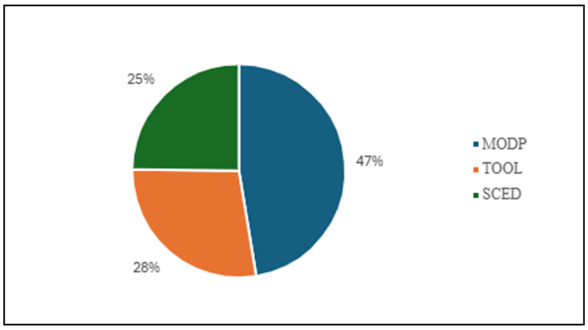

Fig. 7. Project Attributes Weight Percentage.

calculated EAF can be substituted into the Intermediate COCOMO model to calibrate its effort multiplier values, which are often based on less rigorous weighting techniques.

Furthermore, the weights generated by the CFPR method provide a robust, empirically-grounded foundation for machine learning algorithms. Instead of relying on default or assumption-based weights, algorithms like Regression or Least Squares Support Vector Machine (LSSVM) Regression can utilize these optimized weights as input features. This integration is anticipated to significantly improve the predictive precision of these models, as it incorporates expert judgment directly into the algorithmic learning process. The efficiency of the CFPR method, which requires fewer comparisons than techniques like AHP, ensures that this expert input is both consistent and practical to obtain.

V. CONCLUSION

The results of this study demonstrate that Required Reliability is the most influential weighting factor in software effort estimation. As detailed in Table 14, the top three cost drivers are Required Reliability (0.1674), Product Complexity (0.1384), and Execution Time Constraint(0.1050). This ranking suggests that factors directly related to software integrity and operational complexity are perceived by experts as the most critical determinants of development effort.

Conversely, the bottom three cost drivers—Required Development Schedule (0.0303), Virtual Machine Experience (0.0296), and Programming Language Experience (0.0270)—were assigned significantly lower weights, indicating their relatively minor impact on effort estimation compared to productcentric attributes.

The use of the Consistent Fuzzy Preference Relation (CFPR) method proved to be highly efficient, requiring only n – 1 pairwise comparisons while still producing a fully consistent set of weights. The sum of all normalized weights equaling 1 further confirms the internal consistency of the results and the absence of over-weighting. These validated weights can be directly utilized as Effort Adjustment Factors (EAF) in software effort estimation models, offering a robust and

Table 14. The weight of cost drivers using CFPR Method for Software Effort Estimation

ATTRIBUTE	COST	WEIGHT ATTRIBUTES	WEIGHT COST DRIVERS IN ATTRIBUTES	WEIGHT COST DRIVERS	RANK
PRODUCT	RELY	0,372	0,45	0,1674	1
	DATA		0,178	0,0662	6
	CPLX		0,372	0,1384	2
COMPUTER	TIME	0,275	0,382	0,1050	3
	STOR		0,199	0,0547	9
	VIRT		0,145	0,0399	11
	TURN		0,274	0,0754	4
PERSONEL	ACAP	0,231	0,317	0,0732	5
	AEXP		0,25	0,0578	8
	PCAP		0,188	0,0434	10
	VEXP		0,128	0,0296	14
	LEXP		0,117	0,0270	15
PROJECT	MODP	0,122	0,474	0,0578	7
	TOOL		0,278	0,0339	12
	SCED		0,248	0,0303	13

empirically-grounded basis for enhancing estimation accuracy in both traditional and machine learningbased approaches.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to all the experts who participated in this study. We also extend our thanks to STMIK Widya Utama for their institutional support. Additionally, we acknowledge with appreciation the funding provided by Ministry of Education, Culture, Research, and Technology, which made this research possible.

REFERENCES

[1] J. G. Rivera Ibarra, G. Borrego, and R. R. Palacio, "Early estimation in agile software development projects: A systematic mapping study," *Informatics*, dec 2024.

[2] A. Dogga, R. Gatte, B. Kiran Kumar, and A. Rapaka, "A synergistic approach to software effort estimation using fuzzy logic and machine learning," *Journal of Information Systems Engineering and Management*, vol. 10, pp. 2468–4376, 2025.

[3] H. Rastogi, S. Dhankhar, and M. Kakkar, "A survey on software effort estimation techniques," *International Journal of Computer Applications*, pp. 826–830, 2014. Journal name implied from common indexing, original text omitted journal name.

[4] A. Purwanto and L. P. Manik, "Software effort estimation using logarithmic fuzzy preference programming and least squares support vector machines," *Scientific Journal of Informatics*, vol. 10, pp. 1–12, dec 2022.

[5] M. Rahman, P. P. Roy, M. Ali, T. Gonçalves, and H. Sarwar, "Software effort estimation using machine learning technique," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, p. 2023, 2023.

[6] S. Alturki and E.-A. Fazal, "Comprehensive analysis of software effort estimation techniques: Evolving trends, key challenges, and prospective directions," 2025.

[7] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Computational Intelligence and Neuroscience*, vol. 2019, 2019.

[8] A. Trendowicz, J. Münch, and R. Jeffery, "State of the practice in software effort estimation," in *Software Project Effort Estimation*, pp. 232–245, Springer, 2015.

[9] R. Sarno, J. Sidabutar, and Sarwosri, "Improving the accuracy of cocomo's effort estimation based on neural network and fuzzy logic model," in *International Conference on Information Communication Technology and Systems (ICTS)*, pp. 197–202, 2015.

[10] P. V. G. D. P. R. C. V. Hari, "A fine parameter tuning for cocomo 81 software effort estimation using particle swarm optimization," *Journal of Software Engineering*, 2011.

[11] R. K. Yadav, "Optimized model for software effort estimation using cocomo-2 metrics with fuzzy logic," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 7, pp. 121–125, 2017.

[12] C. S. Reddy and K. V. S. V. N. Raju, "An improved fuzzy approach for cocomo's effort estimation using gaussian membership function," *Journal of Software*, vol. 4, no. 5, pp. 452–459, 2009.

[13] S. Chakraverti, S. Kumar, S. C. Agarwal, and A. K. Chakraverti, "Modified cocomo model for maintenance cost estimation of real time system software," *International Journal of Computer Science and Network*, vol. 1, no. 1, pp. 2277–5420, 2012.

[14] E. Herrera-Viedma, F. Herrera, F. Chiclana, and M. Luque, "Some issues on consistency of fuzzy preference relations," *European Journal of Operational Research*, vol. 154, pp. 98–109, apr 2004.

[15] X. Deng, F. T. S. Chan, R. Sadiq, S. Mahadevan, and Y. Deng, "D-cfpr: D numbers extended consistent fuzzy preference relations." ArXiv, mar 2014.

[16] T. Wahyuningrum and R. Pandiya, "Logarithmic fuzzy preference programming approach for evaluating university ranking optimization," *Scientific Journal of Informatics*, vol. 4, no. 1, 2017.

[17] Y. H. Chen and R. J. Chao, "Supplier selection using consistent fuzzy preference relations," *Expert Systems with Applications*, vol. 39, pp. 3233–3240, feb 2012.

[18] T. Wahyuningrum, A. Azhari, and Suprapto, "Improving e-commerce severity rating measurement using consistent fuzzy preference relation," in *International Conference on Soft Computing in Data Science*, (Iizuka, Japan), pp. 1–12, Springer Nature, 2019.

[19] T. C. Wang and Y. H. Chen, "Applying consistent fuzzy preference relations to partnership selection," *Omega*, vol. 35, no. 4, pp. 384–388, 2007.

[20] I. I. Lestari, N. A. Ekowati, and S. Sulistiyasni, "Descriptive analysis and comparison of reasoner using onti measures," *Jurnal Teknik Informatika (Jutif)*, vol. 5, pp. 301–312, feb 2024.