Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

Optimization of Software Effort Estimation Using Hybrid Consistent Fuzzy Preference Relation and Least Squares Support Vector Machine

Ika Indah Lestari*[1], Adnan Purwanto[2], Sulistiyasni[3], Khoem Sambath[4]

[1,2,3]Informatics Engineering, STMIK Widya Utama, Indonesia
[4]Computer Science and Networking, National Polytechnic Institute of Cambodia, Cambodia

Email: [1]ikaindah22@swu.ac.id

Received : Nov 24, 2025; Revised : Dec 2, 2025; Accepted : Dec 2, 2025; Published : Dec 23, 2025

Abstract

The success of software project management hinges on the ability to reliably forecast development effort. However, achieving precise estimates is notoriously difficult, primarily due to inherent project complexities and numerous uncertain variables. While various techniques exist, no single method has proven consistently reliable, leading to inaccurate scheduling and cost overruns. This study aims to develop a more accurate and robust estimation model by hybridizing a multi-criteria decision-making (MCDM) method for handling uncertainty with a machine learning algorithm for predictive modeling. The proposed approach integrates the Consistent Fuzzy Preference Relation (CFPR) method to derive consistent weights for cost drivers from expert judgments. These weights are then used as Effort Adjustment Factors (EAF) to preprocess the COCOMO and NASA datasets, which are subsequently modeled using the Least Squares Support Vector Machine (LSSVM). Evaluation of the hybrid CFPR-LSSVM model confirmed its enhanced predictive accuracy. For the COCOMO dataset, the model yielded an MMRE of 28.463% and an RMSE of 0.4705. Its performance on the NASA dataset was particularly remarkable, with results indicating an MMRE of 1.104% and an RMSE of 0.4593, demonstrating a level of precision that underscores the model's effectiveness. This research contributes a novel hybrid framework that effectively combines consistent fuzzy preference handling with powerful non-linear regression. By providing a more structured and robust methodology for managing uncertainty, this approach offers a substantial advancement in software effort estimation, delivering more reliable predictions for improved project planning.

Keywords : *COCOMO, Consistent Fuzzy Preference Relation, Least Square Support Vector Machine, Machine Learning, NASA,*

1. INTRODUCTION

Accurate software effort estimation constitutes a cornerstone of successful software project management [1], [2] The accuracy of predicting effort encompassing time, labor, and cost, is paramount to a project's success[3]. This estimation is directly linked to budgeting, resource allocation, and risk management. However, predictions often deviate from the actual values, resulting in either underestimates or overestimates. Such inaccuracies lead to schedule delays, cost overruns, or diminished project quality. The primary challenges in effort estimation stem from project complexity, the variability of factors such as software size, team expertise, and technical complexity, and the frequent incompleteness of historical data. Conversely, accurate estimation enables optimal resource allocation, mitigates the risks of delays and cost overruns, and enhances the effectiveness of decision-making during the planning stage[4], [5].

Foundational approaches to effort estimation, including algorithmic models like COCOMO and Function Point Analysis, rely on mathematical formulas derived from historical data[6]. Although these models provide a structured and interpretable approach, their performance is limited when dealing with qualitative and uncertain data[7]. Moreover, the reliability of these estimates is highly contingent on the exactness of the input parameters and calibration constants used, factors that often lack generalizability from one project context to another[8]. To overcome these limitations, researchers have developed soft computing approaches that integrate fuzzy logic with machine learning techniques. Fuzzy logic facilitates the modeling of linguistic or imprecise variables—such as development experience, project complexity, or tool usage—that represent qualitative assessments [7], [9], [10]. To address these non-linear complexities, machine learning models—including Support Vector Machine (SVM) and its efficient variant, Least Squares Support Vector Machine (LSSVM)—prove especially capable of learning the relationship between project attributes and effort [11], [12], [13]. The hybridization of these two approaches is anticipated to yield more accurate and reliable predictions[14].

Previous research in Software Effort Estimation (SEE) has explored various techniques combining fuzzy logic and machine learning. A prominent machine learning algorithm in this domain is the Least Squares Support Vector Machine (LSSVM), which simplifies the underlying quadratic optimization problem by transforming it into a system of linear equations. Yet, a key limitation of LSSVM is the strong dependence of its results on the configuration of kernel parameters and input feature weights. As a result, researchers frequently turn to fuzzy logic techniques to calculate more accurate and representative weightings for this purpose.

Several relevant studies have been conducted, including the combination of fuzzy logic and machine learning[14] , Fuzzy-AHP and LSSVM[15], LFPP-LSSVM [16], Function Point Analysis[17], and a novel ensemble rule[18]. The results from these studies indicate that the integration of fuzzy Multi-Criteria Decision Making (MCDM) methods with LSSVM yields better performance compared to single-method approaches, such as linear regression, Artificial Neural Networks (ANN), or standard SVM.

To perform the weighting, this work implements the Consistent Fuzzy Preference Relation (CFPR) method, utilizing its fuzzy logic framework to handle uncertainty in expert judgments. The objective is to maintain consistency in the preferences among attributes, which can significantly influence the estimation value. Unlike some other methods that rely on subjective weighting without accounting for uncertainty in preferences, CFPR offers a more stable assessment structure that is less vulnerable to subjective inconsistencies. The weights derived from the CFPR process are then integrated into the LSSVM model, ultimately influencing the estimation output. Therefore, by hybridizing the CFPR method for weighting and LSSVM for predictive modeling, this research is expected to yield a more accurate and consistent SEE model.

2. METHOD

To achieve its objective, this research applies a hybrid strategy: it uses a Multi-Criteria Decision Making (MCDM) method to assign weights to the criteria and a Machine Learning (ML) algorithm to develop the predictive models. The MCDM method used is the Consistent Fuzzy Preference Relation (CFPR), which calculates the multiplicative factor, or Effort Adjustment Factor (EAF), for the software effort estimation model. Previous studies have utilized FAHP[19] and LFPP[16] as MCDM approaches. The CFPR method itself has been applied in model evaluation across several studies [20], [21], [22], [23], [24], [25], [26], [27]. The use of CFPR aims to consistently measure expert preferences concerning the factors influencing SEE. Through the CFPR process, a stable preference matrix is expected to be obtained, resulting in more representative input data for the model.

The weighting results from CFPR are then used as the Effort Adjustment Factor (EAF) for pre-defined datasets, which are subsequently modeled using machine learning. In this study, the COCOMO81 and NASA93 datasets were used as model input as can be seen in Table 1.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

Tabel 1. Datasets used for experiment

Dataset	Number of Project
COCOMO 81	63
NASA 93	93

The machine learning model was constructed by training on a dataset partitioned into training and testing subsets [28], [29]. In this study, we employed an 80-20 split for training and testing the model. A total of 15 cost drivers, categorized under four main attributes, were used in the analysis, as detailed in Table 2 [8].

Tabel 2 Cost Drivers and Attributes

ATTRIBUTES	NOTATION	COST DRIVERS
X1.PRODUCT ATTRIBUTES	X11	RELY is Required Reliability
	X12	DATA is Size of Database
	X13	CPLX is Complexity of Product
X2.COMPUTER ATTRIBUTES	X21	TIME is Constrain of Execution time
	X22	STOR is Constrain of Main storage
	X23	VIRT is Volatility of Virtual machine
	X24	TURN is Computer turnaround time
X3.PERSONNEL ATTRIBUTES	X31	ACAP is Capability of Analyst
	X32	AEXP is Experience on Application
	X33	PCAP is Capability of Programmer
	X34	VEXP is Experience on Virtual Machine
	X35	LEXP is Experience on Programming language
X4.PROJECT ATTRIBUTES	X41	MODP is Practice of Modern programming
	X42	TOOL is Use of software tools
	X43	SCED is Required development schedule

Machine learning, a subfield of computer science, is applicable for classification and prediction tasks. This approach has been widely employed in various projects, including credit card fraud detection [30], [31], [32], robot locomotion[33], [34], agriculture[35], [36] and natural language processing [37], [38]. Software effort estimation is another domain that can be addressed through machine learning modeling, as its application can reduce reliance on human effort and minimize manual errors [39]. Consequently, selecting an algorithm capable of delivering accurate and reliable estimations is crucial. The core predictive modeling in this research is built upon the Least Squares Support Vector Machine (LSSVM) algorithm.

The constructed model was subsequently evaluated to assess its performance level. While several performance evaluation methods exist—most of which are statistical approaches such as mean, standard deviation, and coefficient of variation[12][14], [15]—this study focuses specifically on two evaluation

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

metrics: Root Mean Square Error (RMSE) and Mean Magnitude of Relative Error (MMRE). The methodological steps of this research are as follows:

2.1. Consistent Fuzzy Preference Relation (CFPR)

According to [40], for n criteria, CFPR needs only n-1 pairwise comparisons. The CFPR method offers a major methodological benefit by greatly simplifying the evaluation process through a significant reduction in the necessary pairwise comparisons between drivers. In contrast, other methods, such as the Analytic Hierarchy Process (AHP), typically require n(n-1)/2 comparisons [41]. The procedural steps of this study are as follows:

2.1.1. *The Hierarchical Structure of Software Effort Estimation*

The systematic hierarchical structure of the 15 cost drivers, categorized into four attributes—namely Product, Computer, Personnel, and Project—is illustrated in Figure 1.

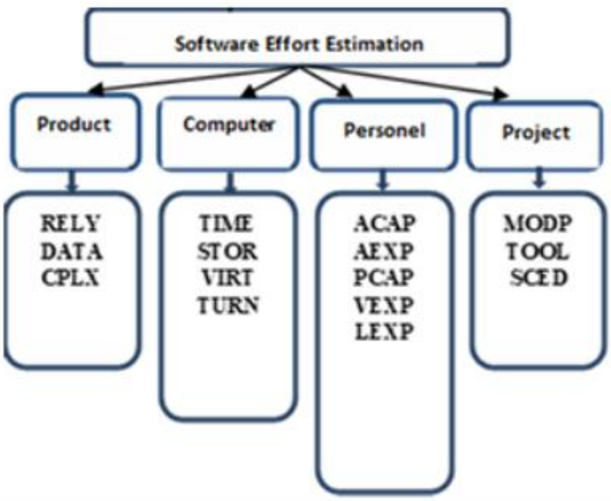


Figure 1. Hierarchical structure of Software Effort Estimation

Tabel 3 Linguistic Scale

Crisp Number	Definition
1	Equal Importance – Both elements have an identical contribution to the objective
2	Intermediate Value – Transition between Moderate Importance and Equal
3	Moderate Importance – One element is moderately preferred over the other.
4	Intermediate Value – Transition between Strong Importance and Moderate
5	Strong Importance – One element is strongly preferred over the other
6	Intermediate Value – Transition between Strong and Very Strong Importance
7	Very Strong Importance – One element is strongly dominant, with clear evidence of preference
8	Intermediate Value – Transition between Absolute Importance and Very Strong
9	Extreme Importance – The maximum degree of preference; the dominance of one element is beyond doubt

2.1.2. *Expert Judgment*

The values for the pairwise comparisons were obtained from assessments provided by five domain experts. These assessment values, expressed as crisp numbers on a scale of 1 to 9 according to Table 3, were then averaged to form the final comparison matrix.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

2.1.3. *Comparison Matrix*

Following the expert judgments, a pairwise comparison matrix was developed. This matrix, which applies the representative values from Table 3, is shown in Table 4. The pairwise comparison matrix for the cost drivers and criteria is constructed from expert assessments. Formally, the comparative relationships between elements are defined by a relation R on the set of cost drivers A, where $R \subseteq A \times A$. This relation is represented by a matrix $R=(r_{ij})$, for all i,j ε {1,2,3,…,n}, where $A=\{a_1,a_2,\ldots,a_n\}$ is the set of n cost drivers or criteria. Each entry r_{ij} denotes the relative importance of driver a_i compared to driver a_j a_j, and the matrix satisfies the reciprocal condition $r_{ij} \,.\, r_{ji} = 1$ for all i,j.

Table 4 The Pairwise Comparison matrix for criteria

Attributes	X1	X2	X3	X4
X1	1	3,48	p13	p14
X2	p21	1	1,76	p24
X3	p31	p32	1	4,05
X4	p41	p42	p43	1

Consider A as the pairwise comparison matrix used within the Consistent Fuzzy Preference Relation (CFPR) framework,

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

The matrix has 1s on its main diagonal. Expert assessments provide the values above the diagonal a_{ij}, and the remaining entries (z) are calculated using Propositions 1 and 2 [19],[25],[26]. A Fuzzy preference relation, denoted as P, is an n × n matrix $P=(p_{ij})$ with corresponds to matrix A, $\forall_{i,j} \in \{1, 2, 3, \ldots, n\}$.

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

- **Proposition 1:**

Consider a finite set of n alternatives, A={a1,a2, ...,an}. Let A=(aij) be the associated reciprocal multiplicative preference relation, where each entry aij ∈ [1/9,9]. The transformation to a reciprocal fuzzy preference relation P=(pij) with values pij=[0,1] is defined as:

$$p_{ij} = g(a_{ij}) = \frac{1}{2}\left(1 + log_9\, a_{ij}\right) \quad (1)$$

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

- **Proposition 2:**

The following conditions are equivalent and are satisfied by a reciprocal fuzzy preference relation P=(pij):

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

$$p_{i(i+1)} + p_{(i+1)(i+2)} + \ldots + p_{(i+k)i} = \frac{k+1}{2}, \forall i < j \quad (3)$$

The calculated values may fall within an interval of [-k, 1+k] for some k > 0. To map these values onto the standard [0, 1] interval, the following transformation function is applied:

$$f : [-k, 1+k] \mapsto [0, 1], f(p) = \frac{p+k}{1+2k} \quad (4)$$

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

$$s_i = \frac{1}{n_c}\left(\sum_{j=1}^{n_c} p_{ij}\right) \quad (5)$$

$$w_i = \frac{s_i}{\sum_{j=1}^{n_c} s_i} \quad (6)$$

2.2. Least Square Support Vector Machine (LSSVM)

The Support Vector Machine (SVM) is a predictive method for classification and regression that works by finding the hyperplane with the maximum margin between classes, typically requiring quadratic optimization. LSSVM introduces a key modification by applying a least squares cost function, which simplifies the solution to a system of linear equations and improves computational speed. However, to maintain SVM's effectiveness with non-linear data, LSSVM also relies on kernel functions, including linear, polynomial, RBF, and Laplace types[42]. For non-linear data, as found in software effort estimation in this study, the RBF kernel is employed. The machine learning modeling steps using LSSVM begin with adjusting the dataset by multiplying each cost driver by the Effort Adjustment Factor (EAF).

2.2.1. Integration of Effort Adjustment Factor (EAF) into the Dataset

During data preprocessing, every cost driver was scaled by its corresponding Effort Adjustment Factor (EAF)—a weight representing the attribute's relative influence on effort. These EAF weights were derived using the Consistent Fuzzy Preference Relation (CFPR) method. This adjustment procedure was implemented uniformly across both the COCOMO and NASA datasets.

2.2.2. Software Effort Model Development

The model was developed using LSSVM with the modified dataset. This dataset was used to train the model to learn the relationship between features and effort. The trained model was then tested to compute the effort estimation values. The precision of the final prediction outcomes was then assessed. This study utilized the Radial Basis Function (RBF) kernel for the COCOMO dataset and three kernels (polynomial, radial basis, linear) for the NASA dataset. The RBF kernel function is defined as:

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

$$K\,(x, x') = exp\left(\frac{|x - x_i|^2}{2\sigma^2}\right) \quad (7)$$

2.3. Performance Testing

In line with common practice for evaluating machine learning models, this study employs Root Mean Square Error (RMSE) and Mean Magnitude of Relative Error (MMRE). These metrics are calculated by comparing the model's predicted effort against the actual effort values from the dataset.

2.3.1. Root Mean Square Error (RMSE)

Measures the standard deviation of prediction errors, representing the average difference between estimated and actual values. Lower RMSE scores correspond to greater predictive accuracy. The RMSE formula is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y}_i)^2} \quad (8)$$

2.3.2. Mean Magnitude of Relative Error (MMRE)

Is defined as the average percentage of absolute error between the predicted and actual values over the entire dataset. The interpretation ranges for MMRE are detailed in Table 5. In many studies, an MMRE value below 25% is considered highly accurate[43]. The MMRE formula is:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \overline{y}_i}{y_i}\right| \quad (9)$$

With y_i= actual value, $\overline{y_i}$= predicted value and n = number of data points.

Table 5 MMRE Interpretation

MMRE Range	Accuracy Interpretation
< 0.25	highly accurate
0.25 - 0.5	acceptable accuracy
0.5 - 0.75	moderate accuracy
> 0.75	low accuracy

3. RESULT

The integration of machine learning with multi-criteria decision-making (MCDM) techniques offers a significant pathway to optimize software effort estimation (SEE) models. In this study, the computational outcomes from the Consistent Fuzzy Preference Relation (CFPR) method were used to establish the weights for the relevant cost drivers. This study involved five domain experts, comprising four academics from different universities and one IT practitioner. The pairwise comparison matrix derived from expert assessments, presented in Table 3, was subsequently completed to establish the initial decision matrix in accordance with the relevant propositions.

3.1. The Initial Decision Matrix

The initial decision matrices for the criteria can be seen in Tables 6 through 10. The remaining values, denoted as 'z', were initialized for subsequent calculations.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

Table 6 The initial decision matrix of the criteria

Attributes	X1	X2	X3	X4
X1	0.5	0.694	z	z
X2	z	0.5	0.588	z
X3	z	z	0.5	0.718
X4	z	z	z	0.5

Table 7 The initial decision matrix of the Product Attributes

Cost driver	X11	X12	X13
X11	0.5	0.908	z
X12	z	0.5	0.21
X13	z	z	0.5

Tabel 8 The initial decision matrix of the Computer Attributes

Cost driver	X21	X22	X23	X24
X21	0.5	0.866	z	z
X22	z	0.5	0.608	z
X23	z	z	0.5	0.241
X24	z	z	z	0.5

Table 9 The initial decision matrix of the Personnel Attributes

Cost driver	X31	X32	X33	X34	X35
X31	0.5	0.667	z	z	z
X32	z	0.5	0.653	z	z
X33	z	z	0.5	0.65	z
X34	z	z	z	0.5	z
X35	z	z	z	z	0.5

Tabel 10 The initial decision matrix of the Project Attributes

Cost driver	X41	X42	X43
X41	0.5	0.795	z
X42	z	0.5	0.545
X43	z	z	0.5

3.2. The Complete Decision Matrix

The full transformation matrix which has been calculated can be seen in Table 11 – Table 15. It shows the weight and rank of each cost driver.

Table 11 The complete decision matrix for Attributes

Cost drivers	X1	X2	X3	X4	Weight	Rank
X1	0.5	0.694	0.782	1	0.372	1
X2	0.306	0.5	0.588	0.806	0.275	2
X3	0.218	0.412	0.5	0.718	0.231	3
X4	0	0.194	0.282	0.5	0.122	4

The weight of each cost driver is presented in Figures 3–7, thus allowing for the identification of the most dominant cost drivers within each attribute.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

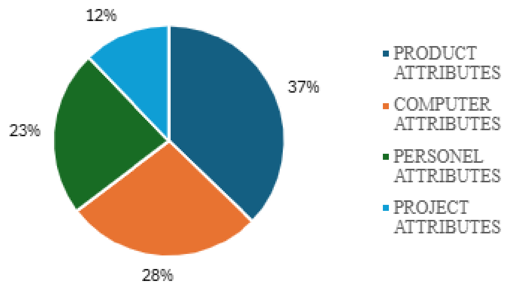


Figure 2 Percentage of Attributes in Software Effort Estimation using CFPR

Figure 2 shows that Product Attributes constitute the most significant factor in software effort estimation, accounting for 37% of the total consideration.

Tabel 12 The complete decision matrix for Product Attributes

Cost driver	X11	X12	X13	Weight	Rank
X11	0.5	0.908	0.618	0.45	1
X12	0.092	0.5	0.21	0.178	3
X13	0.382	0.79	0.5	0.372	2

Tabel 13 The complete decision matrix for Computer Attributes

Cost driver	X21	X22	X23	X24	Weight	Rank
X21	0.5	0.866	0.975	0.715	0.382	1
X22	0.134	0.5	0.608	0.349	0.199	3
X23	0.025	0.392	0.5	0.241	0.145	4
X24	0.285	0.651	0.759	0.5	0.274	2

Tabel 14 The complete decision matrix for Personnel Attributes

Cost driver	X31	X32	X33	X34	X35	Weight	Rank
X31	0.5	0.667	0.82	0.97	1	0.317	1
X32	0.33	0.5	0.653	0.803	0.83	0.25	2
X33	0.18	0.347	0.5	0.65	0.68	0.188	3
X34	0.03	0.197	0.35	0.5	0.53	0.128	4
X35	0	0.167	0.32	0.47	0.5	0.117	5

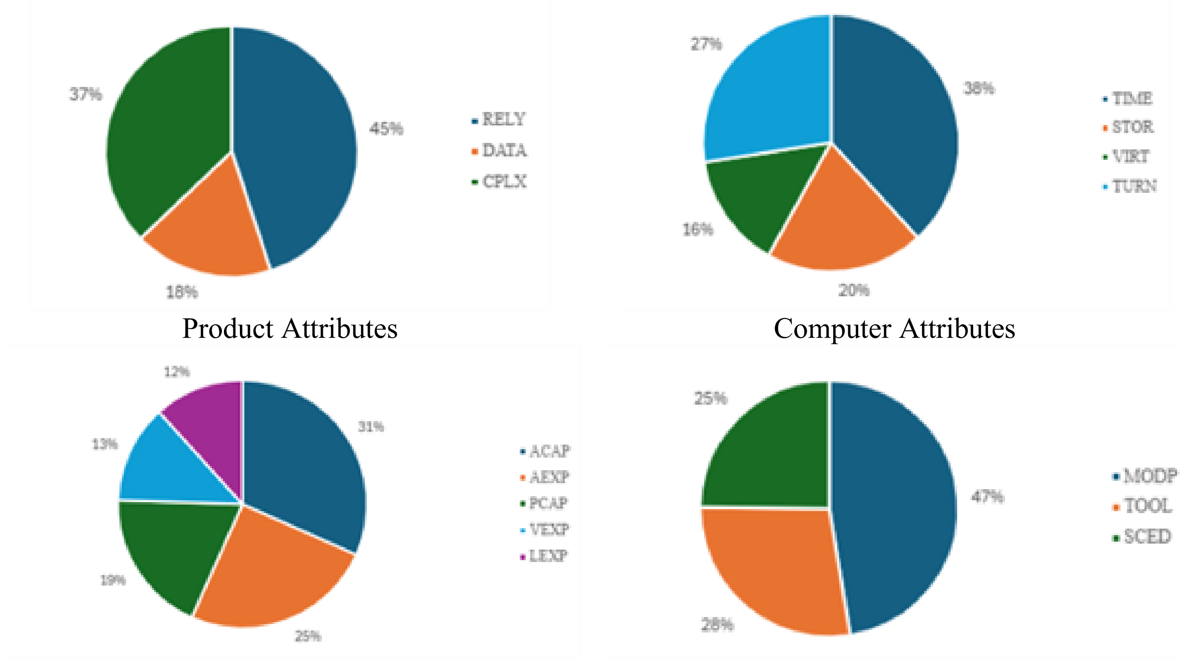
Tabel 15 The complete decision matrix for Project Attributes

Cost driver	X41	X42	X43	Weight	Rank
X41	0.5	0.795	0.84	0.474	1
X42	0.205	0.5	0.545	0.278	2
X43	0.16	0.455	0.5	0.248	3

Figure 3 illustrates comparative weight each attribute derives from the CFPR method, aiming to present results in percentage.

The analysis of attribute weights reveals key drivers of software effort. Within Product Attributes, RELY (Required Reliability) is the dominant concern at 45%, ahead of DATA (Database Size) and CPLX (Product Complexity). For Computer Attributes, TIME (Execution Time Constraint) carries the greatest weight at 38%. Among Personnel Attributes, Analyst Capability is the most significant factor at 31%. In the Project Attributes category, Modern Programming Practices emerges as the highest-priority factor, accounting for 47% of the weight. The complete set of calculated weights and the overall ranking of all cost drivers are provided in Table 16.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465



Product Attributes | Computer Attributes

Personnel Attributes | Project Attributes

Figure 3 Percentage Chart of Each Attribute

Table 16 The weight of cost drivers using CFPR Method for Software Effort Estimation

ATTRIBUTE	COST DRIVER	WEIGHT ATTRIBUTES	WEIGHT COST DRIVERS IN ATTRIBUTES	WEIGHT COST DRIVERS	RANK
PRODUCT(X1)	RELY(X11)	0.372	0.45	0.1674	1
	DATA(X12		0.178	0.0662	6
	CPLX(X13)		0.372	0.1384	2
COMPUTER(X2)	TIME(X21)	0.275	0.382	0.1050	3
	STOR(X22)		0.199	0.0547	9
	VIRT(X23)		0.145	0.0399	11
	TURN(X24)		0.274	0.0754	4
PERSONNEL(X3)	ACAP(X31)	0.231	0.317	0.0732	5
	AEXP(X32)		0.25	0.0578	8
	PCAP(X33)		0.188	0.0434	10
	VEXP(X34)		0.128	0.0296	14
	LEXP(X35)		0.117	0.0270	15
PROJECT(X4)	MODP(X41)	0.122	0.474	0.0578	7
	TOOL(X42)		0.278	0.0339	12
	SCED(X43)		0.248	0.0303	13

3.3. Least Square Support Vector Machine

The weights derived from the CFPR method were subsequently utilized as the Effort Adjustment Factor (EAF) for the available datasets. The datasets, modified with these CFPR weights, were then used to construct the model through an initial training process. The evaluation measures used in this study were Root Mean Squared Error (RMSE) and Mean Magnitude of Relative Error (MMRE). The hyperparameters explored for γ, σ, and c spanned the values [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1.0e3], whereas parameter d was varied from 1 to 24. The kernels examined included RBF, Polynomial, and Linear. The results for each dataset are as follows:

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 6, December, 2025, Page. 5590-5608
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

3.3.1. COCOMO

The Constructive Cost Model (COCOMO) dataset, introduced by Barry W. Boehm (1981), is a seminal benchmark in software effort estimation research. Each record includes essential attributes like Lines of Code (LOC), development mode, Effort Adjustment Factors, and the actual effort. The performance outcomes obtained by applying the CFPR-weighted COCOMO data to the LSSVM model are summarized in Table 17.

Table 17. COCOMO Dataset Result of Performance Test

No	Actual	LSSVM Testing			
		Predicted	Pred (10)	Pred (5)	Pred (1)
1	2040	2030.19159	1	1	1
2	1600	1594.548026	1	1	1
3	43	51.78541539	0	0	0
4	1075	1070.219775	1	1	1
5	423	421.8594938	1	1	1
6	321	311.0127239	1	1	0
7	201	207.9545966	1	1	0
8	79	86.72738174	1	0	0
9	40	17.09538707	0	0	0
10	9	25.49803086	0	0	0
11	12	19.86659891	0	0	0
12	87	95.83259402	0	0	0
13	50	57.71043103	0	0	0

The evaluation of the COCOMO dataset yielded an MMRE of 28.463% (0.2846) and an RMSE of 0.47047. An MMRE of 28% suggests the model's predictions are reasonably aligned with actual project efforts. Furthermore, the low RMSE value signifies a small average squared error, indicating that the model provides both consistent and precise estimates.

As illustrated in Figure 4, the predicted values closely approximate the actual values, signifying a reasonably good level of accuracy. Furthermore, the correlation coefficient between the actual and predicted values was calculated to be 0.99. Since this value approaches 1, it can be concluded that the estimation method exhibits high precision. A comparative evaluation of MMRE and RMSE with several previous research methods [12] is presented in Table 18.

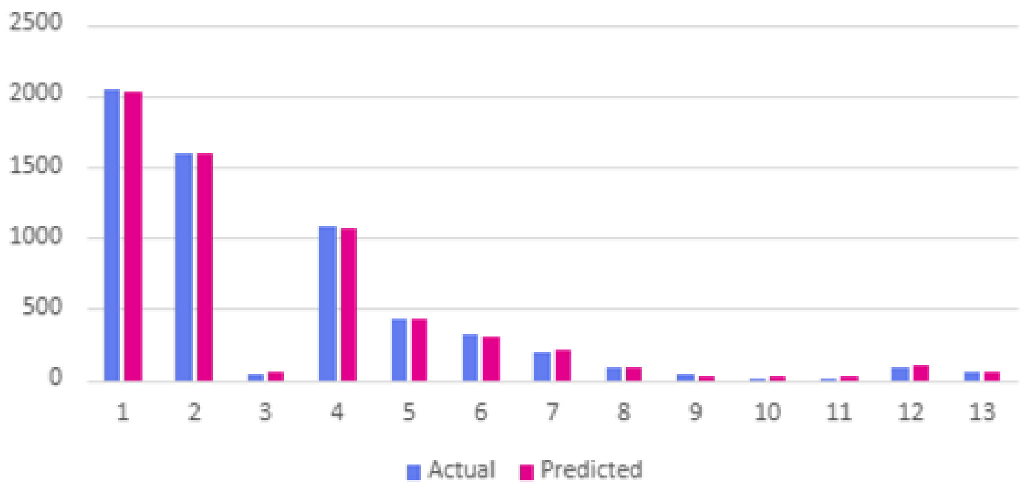


Figure 4 Comparison between Actual Effort and Predicted Effort for Dataset COCOMO

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

Table 18 Comparison of MMRE and RMSE Values Using RBF-LSSVM, FAHP-RBF-LSSVM, LFPP-RBF-LSSVM, and CFPR-RBF-LSSVM Methods on the COCOMO Dataset

	Method	Result
MMRE		
1	RBF-LSSVM	0.82
2	FAHP-RBF-LSSVM	0.57
3	LFPP-RBF-LSSVM	0.015
4	CFPR-RBF-LSSVM	0.2846
RMSE		
1	RBF-LSSVM	630.78
2	FAHP-RBF-LSSVM	569.43
3	LFPP-RBF-LSSVM	1.703
4	CFPR-RBF-LSSVM	0.47047

As shown in Table 18, the combined CFPR-RBF-LSSVM approach delivers more stable and accurate results overall. Additionally, it enhances the consistency of input preferences and optimizes the predictive performance of the LSSVM model.

3.3.2. NASA

In addition to the COCOMO dataset, another standard dataset widely used in Software Effort Estimation research is the data developed by the NASA Metrics Data Program (MDP). This dataset contains historical information from various NASA software development projects, including code size (Lines of Code), number of modules, complexity, and actual effort. The results obtained from testing this dataset on the LSSVM model are presented in Table 19.

Table 19 NASA Dataset Result of Performance Test

No	Actual	Pengujian LSSVM			
		Predicted	Pred (10)	Pred (5)	Pred (1)
1	117.6	118.1095419	1	1	1
2	8.4	9.01863285	1	0	0
3	10.8	11.41623525	1	0	0
4	72	72.55500094	1	1	1
5	360	360.2673841	1	1	1
6	36	36.59106042	1	1	0
7	324	324.3033481	1	1	1
8	48	48.57907241	1	1	0
9	60	60.5670844	1	1	1
10	48	48.57907241	1	1	0
11	82	82.54510638	1	1	1
12	444	444.183468	1	1	1
13	98.8	99.32832316	1	1	1
14	636	635.9916598	1	1	1
15	300	300.3273242	1	1	1
16	300	300.3273242	1	1	1
17	756	755.8717797	1	1	1
18	1200	1199.428223	1	1	1
19	703	702.9247268	1	1	1

For the NASA dataset, the MMRE and RMSE values were 0.01104 (1.104%) and 0.4593, respectively. The MMRE of just 1.1% is exceptionally low and very close to the true values. As noted by Conte et al. [43], an estimation model is considered reliable when MMRE < 0.25. With an RMSE of

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

0.4593, the model shows minimal average prediction error, thereby confirming its high accuracy and effective performance.

Similar to the results on the COCOMO dataset, visual inspection of Figure 5 shows a close alignment between predicted and actual values, which validates the model's good accuracy. A comparative evaluation using MMRE and RMSE on the NASA dataset with several previous research methods [12] is presented in Table 20.

Table 20 further demonstrates that the model using the NASA dataset overall provides better and more accurate prediction results. The MMRE value for the NASA dataset is significantly smaller than that of the COCOMO dataset, approaching nearly zero. The difference in RMSE values between the NASA and COCOMO datasets is not particularly significant. As illustrated in Figure 6, the estimation model performs relatively more accurately on the NASA dataset compared to the COCOMO dataset.

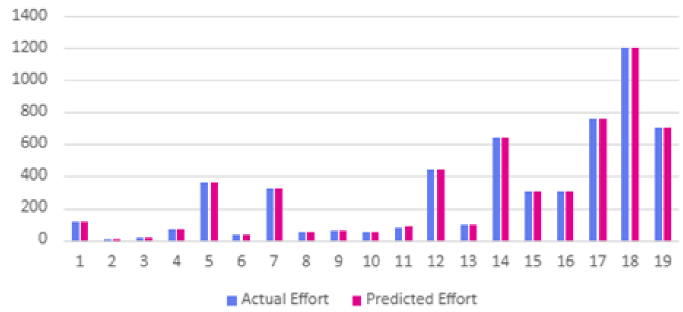


Figure 5 Comparison between Actual Effort and Predicted Effort for Dataset NASA

Table 20 Comparison of MMRE and RMSE Values Using RBF-LSSVM, FAHP-RBF-LSSVM, LFPP-RBF-LSSVM, and CFPR-RBF-LSSVM Methods on the NASA Dataset

	Method	Result
MMRE		
1	RBF-LSSVM	0.5
2	FAHP-RBF-LSSVM	0.19
3	LFPP-RBF-LSSVM	0.0073
4	CFPR-RBF-LSSVM	0.0110
RMSE		
1	RBF-LSSVM	19.74
2	FAHP-RBF-LSSVM	5.99
3	LFPP-RBF-LSSVM	6.039
4	CFPR-RBF-LSSVM	0.4593

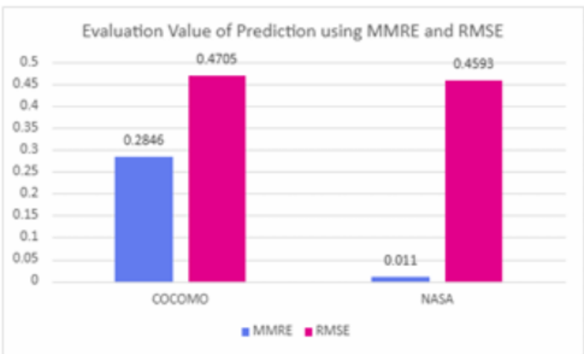


Figure 6 Evaluation Values of prediction using MMRE and RMSE

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

4. DISCUSSIONS

The primary objective of this research was to develop and evaluate a hybrid software effort estimation (SEE) model that integrates the Consistent Fuzzy Preference Relation (CFPR) method for criteria weighting with the Least Squares Support Vector Machine (LSSVM) for predictive modeling. The results presented in the previous section demonstrate that the proposed CFPR-LSSVM approach successfully enhances the accuracy and robustness of effort predictions. This discussion interprets these findings, explores the reasons behind the model's performance, and considers the implications for both research and practice.

4.1. Interpretation of the CFPR Weighting Results

The application of the CFPR method provided critical insights into the relative importance of various cost drivers. The derived weights, as summarized in Table 16, reveal that Product Attributes (37.2%) are considered the most significant factor group by domain experts, with RELY (Required Reliability) emerging as the single most influential cost driver overall (16.74%). This aligns with intuitive project management wisdom, as the fundamental quality and complexity of the product itself are often the primary determinants of development effort. The high ranking of CPLX (Product Complexity) and TIME (Execution Time Constraint) further underscores that technically challenging and performance-critical projects demand substantially more resources.

A key advantage of using CFPR, as opposed to other Multi-Criteria Decision-Making (MCDM) methods like standard AHP, is its inherent mechanism for ensuring transitive consistency in expert judgments. By requiring only (n-1) pairwise comparisons, CFPR reduces the cognitive burden on experts and minimizes the potential for contradictory preferences. The consistent and stable preference matrices obtained (Tables 11-15) indicate that the derived weights are a reliable representation of expert consensus, thereby providing a more solid foundation for the subsequent machine learning phase compared to ad-hoc or inconsistent weighting schemes.

4.2. Performance of the Hybrid CFPR-LSSVM Model

The core contribution of this study lies in the hybridization of CFPR and LSSVM. The performance metrics on both the COCOMO and NASA datasets validate the effectiveness of this integration.

4.2.1. COCOMO Dataset

On the COCOMO Dataset: The model performance was good, though less stellar than on the NASA dataset, with an MMRE of 28.463% and an RMSE of 0.4705. While the MMRE is slightly above the 25% threshold, it represents a significant improvement over basic methods like RBF-LSSVM alone (MMRE 82%) and is comparable to other hybrid approaches like FAHP-LSSVM (MMRE 57%), as shown in Table 18. The RMSE value is very low and similar to that achieved on the NASA dataset, indicating that the magnitude of the errors is small and the model is stable.

4.2.2. NASA Dataset

On the NASA Dataset: The model achieved exceptional performance, with an MMRE of 1.104% and an RMSE of 0.4593. An MMRE below 25% is widely considered acceptable in SEE literature, and a result near 1% indicates a remarkably high level of accuracy. This suggests that for the NASA dataset, which comprises projects with certain characteristics, the CFPR-weighted LSSVM model can predict effort with minimal deviation from the actual values. The low RMSE further confirms the model's precision and consistency.

The disparity in MMRE performance between the two datasets can be attributed to several factors:

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

4.2.2.1. Dataset Characteristics

The COCOMO81 dataset is known to be more challenging, with higher variance and more non-linear relationships among its attributes. The NASA93 dataset may have a more consistent underlying pattern that is easier for the LSSVM to learn once properly weighted.

4.2.2.2. Impact of CFPR Weighting

The CFPR-derived EAF effectively recalibrates the input features to reflect their true influence on effort. This preprocessing step appears to be particularly beneficial for the NASA dataset, allowing the LSSVM to model the relationships with extreme accuracy. For COCOMO, it brings a substantial improvement over the baseline but does not fully overcome the dataset's inherent complexity.

4.3. Comparative Advantage and Contribution

As illustrated in Tables 18 and 20, the CFPR-LSSVM model consistently outperforms the standalone LSSVM model and shows competitive or superior results compared to other hybrid models like FAHP-LSSVM and LFPP-LSSVM. This comparative analysis highlights two key contributions:

4.3.1. Robustness through Hybridization

The model mitigates the limitations of individual techniques. CFPR alone cannot predict effort, and LSSVM's performance is sensitive to input feature scaling and relevance. By using CFPR to provide theoretically sound and consistent weights, we enhance LSSVM's ability to discern the complex, non-linear relationships between project attributes and effort.

4.3.2. A Framework for Handling Uncertainty

Software project estimation is inherently uncertain. This hybrid approach directly addresses this by using fuzzy logic (via CFPR) to formalize and quantify the uncertainty in expert judgments, and machine learning (via LSSVM) to learn from historical data patterns. The result is a method that is both consistent (thanks to CFPR's mathematical properties) and robust (thanks to LSSVM's predictive power).

4.4. Limitations and Future Work

Despite the promising results, this study has certain limitations. The model's performance is validated on two public datasets; its generalizability should be further tested on a wider array of datasets, including those from modern agile projects and different industrial domains. Furthermore, the CFPR process relies on expert judgment, which, while made more consistent, can still be subject to availability and selection bias. Based on these limitations, future work should focus on:

4.4.1. Developing a Web-Based Tool

As hinted in the manuscript, a practical next step is to implement this hybrid methodology into a user-friendly, web-based estimation system for use by project managers.

4.4.2. Expanding Dataset Validation

Applying and validating the CFPR-LSSVM model on more recent and diverse industrial datasets to strengthen its external validity.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

4.4.3. Exploring Advanced ML Models

Investigating the integration of CFPR weights with other advanced machine learning or deep learning models to see if further accuracy gains can be achieved, especially for more difficult datasets like COCOMO.

In conclusion, the discussion affirms that the hybrid CFPR-LSSVM approach offers a scientifically rigorous and practically valuable contribution to the field of software effort estimation, providing a more consistent and robust methodology for managing the uncertainty inherent in predicting project effort.

5. CONCLUSION

This study has successfully developed and validated a hybrid software effort estimation (SEE) model by integrating the Consistent Fuzzy Preference Relation (CFPR) method with the Least Squares Support Vector Machine (LSSVM). The primary objective was to enhance prediction accuracy and robustness by systematically handling the uncertainty in expert judgments and effectively modeling the non-linear relationships in project data.

The key findings and contributions of this research are summarized as follows:

1. Effective Hybridization: The proposed CFPR-LSSVM model demonstrates that combining a fuzzy Multi-Criteria Decision-Making (MCDM) technique with a machine learning algorithm creates a synergistic effect. The CFPR method provides a mathematically consistent and reliable framework for determining the weights of cost drivers (the Effort Adjustment Factor), which in turn optimizes the input features for the LSSVM predictive model.
2. Enhanced Accuracy: The experimental results on two standard benchmark datasets confirm the model's effectiveness. On the NASA dataset, the model achieved an exceptionally low MMRE of 1.104% and an RMSE of 0.4593. On the more challenging COCOMO dataset, it delivered an MMRE of 28.463% and an RMSE of 0.4705. These results, particularly when compared to standalone LSSVM and other hybrid models, indicate a significant improvement in estimation accuracy and consistency.
3. Robust Framework for Uncertainty: This research provides a more structured and consistent alternative for handling the inherent uncertainty in software project estimation. The CFPR component ensures that expert preferences are transitive and logically consistent, thereby reducing subjective bias. Meanwhile, the LSSVM component robustly captures the complex, non-linear relationships between project attributes and the required effort.

In conclusion, the CFPR-LSSVM hybrid approach offers a scientifically-grounded and practical methodology for software effort estimation. It bridges the gap between qualitative expert judgment and quantitative data-driven prediction, resulting in a model that is both accurate and reliable. For future work, the implementation of this approach into a practical web-based tool and its validation on a broader range of industrial datasets are recommended to further solidify its utility for both academics and software project managers.

ACKNOWLEDGEMENT

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.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

REFERENCES

[1] P. Sharma and J. Singh, "Systematic literature review on software effort estimation using machine learning approaches," in *Proceedings - 2017 International Conference on Next Generation Computing and Information Systems, ICNGCIS 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 54–57. doi: 10.1109/ICNGCIS.2017.33.

[2] K. Shin Thant and H. Htake Khaung Tin, "Learning The Efficient Estimation Techniques For Successful Software Project Management," *Innovare Journal of Engineering & Technology*, vol. 11, pp. 4–8, Jun. 2023, doi: 10.22159/ijet.2023v11i3.47605.

[3] J. Carpenter, C.-Y. Wu, and N. U. Eisty, "Leveraging Large Language Models for Predicting Cost and Duration in Software Engineering Projects," *ArXiv*, Sep. 2024, [Online]. Available: http://arxiv.org/abs/2409.09617

[4] A. Ali and C. Gravino, "A systematic literature review of software effort prediction using machine learning methods," *Journal of Software: Evolution and Process*, vol. 31, no. 10, Oct. 2019, doi: 10.1002/smr.2211.

[5] M. M. I. Shamim, A. B. bin A. Hamid, T. E. Nyamasvisva, and N. S. Bin Rafi, "Advancement of Artificial Intelligence in Cost Estimation for Project Management Success: A Systematic Review of Machine Learning, Deep Learning, Regression, and Hybrid Models," *Modelling*, vol. 6, no. 2, Jun. 2025, doi: 10.3390/modelling6020035.

[6] X. Zhao, X. Xiong, Z. Mansor, R. Razali, M. Z. Ahmad Nazri, and L. Li, "A data-driven cost estimation model for agile development based on Kolmogorov-Arnold Networks and AdamW optimization," *Journal of King Saud University - Computer and Information Sciences*, vol. 37, no. 5, Jul. 2025, doi: 10.1007/s44443-025-00058-7.

[7] Ritu and P. Bhambri, "Enhancing software development effort estimation with a cloud-based data framework using use case points, fuzzy logic, and machine learning," *Discover Computing*, vol. 28, no. 1, Jul. 2025, doi: 10.1007/s10791-025-09668-1.

[8] A. O. Sousa *et al.*, "Applying Machine Learning to Estimate the Effort and Duration of Individual Tasks in Software Projects," *IEEE Access*, Aug. 2023, doi: 10.1109/ACCESS.2023.3307310.

[9] 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, [Online]. Available: https://www.jisem-journal.com/

[10] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Comput Intell Neurosci*, vol. 2019, 2019, doi: 10.1155/2019/8367214.

[11] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms," *Journal of Systems and Software*, vol. 137, pp. 184–196, Mar. 2018, doi: 10.1016/j.jss.2017.11.066.

[12] N. Azura Zakaria, A. Ritahani Ismail, A. Yakath Ali, N. Hidayah Mohd Khalid, and N. Zainal Abidin, "Software Project Estimation with Machine Learning," *Int J Adv Comput Sci Appl*, vol. 12, no. 6, pp. 726–734, 2021, doi: 10.14569/IJACSA.2021.0120685.

[13] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Process Lett*, vol. 9, pp. 293–300, Jun. 1998.

[14] A. Jaiswal, J. Raikwal, and P. Raikwal, "A Hybrid Cost Estimation Method for Planning Software Projects Using Fuzzy Logic and Machine Learning," *Journal of Intelligent Systems and Applications in Engineering IJISAE*, vol. 12, no. 1, pp. 696–707, Dec. 2023, [Online]. Available: www.ijisae.org

[15] S. K. Sehra, Y. Singh, B. Navdeep, K. Sukhjit, and S. Sehra, "Software effort estimation using FAHP and weighted kernel LSSVM machine," *Soft comput*, vol. 23, no. 21, pp. 10881–10900, 2019, doi: 10.1007/s00500-018-3639-2.

[16] 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, no. 1, pp. 1–12, Dec. 2022, doi: 10.15294/sji.v10i1.39865.

[17] L. Lavazza, A. Locoro, and R. Meli, "Software Development and Maintenance Effort Estimation Using Function Points and Simpler Functional Measures," *Software*, vol. 3, no. 4, pp. 442–472, Oct. 2024, doi: 10.3390/software3040022.

[18] S. S. Ali, J. Ren, and J. Wu, "Framework to improve software effort estimation accuracy using novel ensemble rule," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 9, Nov. 2024, doi: 10.1016/j.jksuci.2024.102189.

[19] S. K. Sehra, Y. S. Brar, and N. Kaur, "Applying Fuzzy-AHP for software effort estimation in data scarcity," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 45, no. 1, pp. 4–9, Mar. 2017, Accessed: Nov. 30, 2025. [Online]. Available: https://ijettjournal.org/archive/ijett-v45p202

[20] T. Wahyuningrum, A. Azhari, and Suprapto, "An Extended Consistent Fuzzy Preference Relation to Evaluating Website Usability," *Int J Adv Comput Sci Appl*, vol. 10, no. 9, pp. 109–116, 2019, [Online]. Available: www.ijacsa.thesai.org

[21] K. G. Nalbant, "Using an Integrated Consistent Fuzzy Preference Relations and Interval Type-2 Fuzzy Topsis Methodology for Personnel Selection and Promotion," *WSEAS TRANSACTIONS ON COMPUTERS*, vol. 21, pp. 158–164, Jun. 2022, doi: 10.37394/23205.2022.21.20.

[22] H. Nurdiyanto, C. Fauzi, S. Lestari, E. F. Saputra, and M. Mushowir, "Fuzzy Preference Relations-Based AHP for Multi-Criteria Supplier Segmentation," *International Journal of Artificial Intelligence Research*, vol. 8, no. 1, p. 117, Apr. 2024, doi: 10.29099/ijair.v7i1.1.1103.

[23] J. Li, L. li Niu, Q. Chen, Z. xing Wang, and W. Li, "Group decision making method with hesitant fuzzy preference relations based on additive consistency and consensus," *Complex and Intelligent Systems*, vol. 8, no. 3, pp. 2203–2225, Jun. 2022, doi: 10.1007/s40747-021-00585-6.

[24] F. M. A. Alias and L. Abdullah, "Consistent Fuzzy Preference Relation Method and Its Application to Criteria Assessment of Quality of Life Among Population of Setiu Wetlands," *Discovering Mathematics*, vol. 39, no. 2, pp. 52–60, 2017.

[25] F. M. A. Alias, L. Abdullah, X. Gou, H. Liao, and E. Herrera-Viedma, "Consistent fuzzy preference relation with geometric Bonferroni mean: a fused preference method for assessing the quality of life," *Applied Intelligence*, vol. 49, no. 7, pp. 2672–2683, Jan. 2019, doi: 10.1007/s10489-019-01415-6.

[26] T. Wahyuningrum, C. Kartiko, and T. G. Soares, "Evaluating ExcellenceWebometric's CriteriaWeight using Consistent Fuzzy Preference Relation Method," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2021, pp. 117–121. doi: 10.1145/3489088.3489112.

[27] A. F. Isnawati, R. Pandiya, and A. Wahyudin, "Evaluation of MVNO model implementation in remote and border areas using the consistent fuzzy preference relations method," *JURNAL INFOTEL*, vol. 13, no. 4, pp. 223–229, Dec. 2021, doi: 10.20895/infotel.v13i4.691.

[28] R. K. Gora and R. R. Sinha, "A Study of Evaluation Measures for Software Effort Estimation Using Machine Learning," *International Journal of Intelligent Systems and Applications in Engineering IJISAE*, vol. 11, no. 6s, pp. 267–275, May 2023, [Online]. Available: www.ijisae.org

[29] H. Bichri, A. Chergui, and M. Hain, "Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," *International Journal of Advanced Computer Science and Applications (IJCSA)*, vol. 15, no. 2, 2024, doi: 10.14569/IJACSA.2024.0150235.

[30] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *J Big Data*, vol. 9, Feb. 2022, doi: 10.1186/s40537-022-00573-8.

[31] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," in *International Conference On Recent Trends In Advanced Computing ICRTAC*, Elsevier B.V., 2019, pp. 631–641. doi: 10.1016/j.procs.2020.01.057.

[32] B. Mardan and A. Mohsin Abdulazeez, "Credit Card Fraud Detection Based on Machine Learning Classification Algorithm," *Indonesian Journal of Computer Science*, vol. 13, no. 3, pp. 4194–4216, Jun. 2024.

[33] Y. Wang, R. Sagawa, and Y. Yoshiyasu, "Learning Advanced Locomotion for Quadrupedal Robots: A Distributed Multi-Agent Reinforcement Learning Framework with Riemannian Motion Policies," *Robotics*, vol. 13, no. 6, May 2024, doi: 10.3390/robotics13060086.

Vol. 6, No. 6, December, 2025, Page. 5590-5608
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5465

[34] Q. Liu, J. Guo, S. Lin, S. Ma, J. Zhu, and Y. Li, "MASQ: Multi-Agent Reinforcement Learning for Single Quadruped Robot Locomotion," *ArXiv*, Oct. 2024, [Online]. Available: http://arxiv.org/abs/2408.13759

[35] V. Meshram, K. Patil, V. Meshram, D. Hanchate, and S. D. Ramkteke, "Machine learning in agriculture domain: A state-of-art survey," *Artificial Intelligence in the Life Sciences*, vol. 1, Dec. 2021, doi: 10.1016/j.ailsci.2021.100010.

[36] D. E. Clay, S. Brugler, and B. Joshi, "Will artificial intelligence and machine learning change agriculture: A special issue," *Agron J*, vol. 116, no. 3, pp. 791–794, May 2024, doi: 10.1002/agj2.21555.

[37] A. Deekshith, "Advances In Natural Language Processing: A Survey Of Techniques," *International Journal Of Innovations In Engineering Research And Technology (IJIERT)*, vol. 8, no. 3, pp. 2394–3696, Mar. 2021.

[38] Anuj Kumar Dwivedi and Mani Dwivedi, "A Study on The Role of Machine Learning in Natural Language Processing," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 8, no. 4, pp. 192–198, Aug. 2022, doi: 10.32628/cseit228429.

[39] Meharunnisa, M. Saqlain, M. Abid, M. Awais, and Ž. Stević, "Analysis of Software Effort Estimation by Machine Learning Techniques," *International Information and Engineering Technology Association (IIETA)*, vol. 28, no. 6, pp. 1445–1457, Dec. 2023, doi: 10.18280/isi.280602.

[40] Y. H. Chen and R. J. Chao, "Supplier selection using consistent fuzzy preference relations," *Expert Syst Appl*, vol. 39, no. 3, pp. 3233–3240, Feb. 2012, doi: 10.1016/j.eswa.2011.09.010.

[41] 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, [Online]. Available: http://arxiv.org/abs/1403.5753

[42] D. K. Wornyo and X. J. Shen, "Coupled least squares support vector ensemble machines," *Information*, vol. 10, no. 6, Jun. 2019, doi: 10.3390/info10060195.

[43] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics And Models*. California: Benjamin-Cummings Publishing Co., Inc., 1986.