

Article

Automation of Determining Criterion Weights in Decision Support Tasks Based on Regression Analysis Results

Vladimir A. Sudakov ^{1,2} , Aleksandr V. Timoshenko ^{2,*} , Lyubov A. Komarova ²  and Aleksandr S. Zakharov ² ¹ Department of Problems of Mathematical Modeling and High-Performance Computing, Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Moscow 125047, Russia; vasudakov@fa.ru² Faculty of Information Technology and Big Data Analysis, Financial University Under the Government of the Russian Federation, Moscow 125167, Russia; lakomarova@fa.ru (L.A.K.); askakharov@fa.ru (A.S.Z.)

* Correspondence: avtimoshenko@fa.ru

Abstract

Currently, when operating the decision support system of a critical infrastructure facility's situation center, difficulties arise in determining the weights of criteria for selecting the optimal solution. The main source of errors is expert weighting errors, due to the fact that different experts may assign significantly different weights to the same set of criteria. The paper proposes and experimentally investigates an approach to determining the weights of criteria in multi-criteria decision support tasks based on the use of machine learning methods and generative language models. Unlike classical analytical and expert methods, the proposed methodology formalizes the task of assigning weights as a regression task, in which each criterion is described by a set of statistical characteristics of the distribution of its estimates across a set of alternatives. The experiments conducted on the LLM-generated synthetic benchmark showed that the ML approach, in particular the Ridge regression model, achieved the lowest test MSE among the considered regression models and outperformed the analytical and combined baselines in the reported setting. These results suggest that regression-based estimation of synthetic reference criterion weights may be a promising direction for automating weight determination in MCDM tasks. However, since the current empirical evidence is based on synthetic data from one selected domain, the results should not be interpreted as proof of general validity for real expert decision-making settings without further validation on real-world MCDM datasets.

Keywords: multi-criteria decision making; weight coefficients of criteria; automation of weight assignment; machine learning; prioritization of alternatives

MSC: 90B50



Academic Editors: James Liou and
Artūras Kaklauskas

Received: 12 May 2026

Revised: 25 June 2026

Accepted: 29 June 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In decision support tasks, one of the most critical stages is assigning weighting coefficients to criteria, which are used to build objective numerical indicators for the decision. If there is an error in the weighting coefficients, it will not be possible to obtain an objective assessment of alternatives for the critical infrastructure facility (CIF) dispatcher.

Traditional approaches rely on the subjective opinion of experts when determining the importance of criteria [1–3]. This approach often does not allow for an objective assessment of alternatives and does not ensure the correct calculation of the sum of all weighted decision assessments, because different experts may assign significantly different weights to the same set of criteria. Such divergence in expert weighting may be caused not only

by differences in professional experience, but also by cognitive mechanisms that affect decision-making under uncertainty. In particular, habitual decision-making and status quo bias may lead experts to over-rely on familiar operating patterns, previously accepted decisions, or established prioritization schemes, even when the current emergency context requires re-evaluation of criterion importance. This provides an additional behavioral justification for developing data-driven and reproducible procedures for criterion-weight estimation in decision support systems for critical infrastructure dispatchers.

The scientific contribution of this study consists of four aspects. First, the paper formalizes the problem of determining criterion weights in multi-criteria decision-making as a regression problem at the level of criteria, rather than at the level of alternatives. Second, it proposes a feature representation of each criterion based on statistical characteristics of the distribution of expert assessments, including central tendency, dispersion, distribution shape, correlation structure, and entropy-based characteristics. Third, the study integrates machine learning methods with a generative language model-based procedure for obtaining synthetic expert-like assessments and synthetic reference weight samples under controlled conditions. Fourth, it provides a comparative evaluation of regression models and classical weighting approaches for restoring reference criterion weights in decision-support tasks.

2. Literature Review

Determining the weight coefficients of criteria is one of the keys and most difficult stages of multi-criteria decision-making. It is known that the final ratings of alternatives depend on the selected weights of criteria, so simply assigning equal weights to all criteria is usually considered suboptimal. In the literature, methods for assessing the significance of criteria are usually classified into three groups: subjective, objective, and integrated (hybrid) [4–7].

In particular, there may be situations where criteria are often correlated, data may be noisy and incomplete, and priorities depend on the network mode and the criticality of disconnected consumers. For this reason, modern decision support methods (DSM) must be highly resistant to noise, adaptable to different accident scenarios, and reproducible with identical input data. Below is a systematic overview of the principles of each group of methods, as well as their advantages, disadvantages, areas of application, and comparative characteristics.

Subjective methods rely entirely on the opinions and preferences of experts or decision-makers (DMs) when assigning weights to criteria. This approach does not involve working with data. In other words, weights are determined a priori based on judgments, without direct calculation from data. In fact, the responsibility for the significance of the criteria lies entirely with the expert. Various expert assessment techniques are used: direct ranking, when the expert directly ranks the criteria by importance, or the paired comparison method (Saaty's analytical hierarchy process) [4–7], when the expert compares the criteria in pairs according to their relative importance, after which the weights are calculated as components of the eigenvector of the pairwise comparison matrix. These methods are applicable even in the absence of data on alternatives, so they are widely used in strategic and socio-economic tasks where criteria are difficult to formalize quantitatively. Expert methods are flexible and understandable to decision makers, and popular techniques such as the analytic hierarchy process (AHP) provide a transparent and hierarchical evaluation process. The advantage of these methods is that they allow for qualitative factors and expert intuition to be taken into account, but this is also their main disadvantage, as they are dependent on the human factor and are therefore subjective.

Objective methods do not require the involvement of experts in calculating weight coefficients [8–12]. Weights are derived automatically from the initial data, the so-called de-

cision matrix, which contains assessments of all alternatives according to all criteria [13,14]. It is assumed that the information contained in the data reflects the relative importance of the criteria. Typically, such methods measure the degree of variability or dispersion of criterion values: it is believed that a criterion whose values vary significantly between alternatives has a greater ability to influence the choice and therefore should be given greater weight. An important difference between objective approaches is their replicability, where all researchers will obtain the same weights for the same data, eliminating subjective influence. The main representatives are: the entropy method (based on the concept of information entropy), the CRITIC method (Criteria Importance Through Inter-Criteria Correlation), and the Gini method [14] (based on the Gini coefficient, borrowed from economic theory to measure the unevenness of distribution).

An integrated approach [15] seeks to combine the advantages of both subjective and objective approaches. In this approach, weights can initially be obtained in one way and then adjusted in another, or the results of two or more methods can be directly aggregated. Combined methods do not have their own unique formula; they borrow features from the original methods that make up the mixture [16–19]. However, their strength lies in more realistic and balanced weights that reflect both expert preferences and data structure. There are many integration options: from simple averaging to complex optimization algorithms [10–12].

Table 1 compares methods and classes in terms of complexity, versatility, and resistance to data noise.

Table 1. Comparison of methods.

Method	Class	Expert Involvement	Complexity	Versatility	Noise Resistance
AHP	Subjective	High	Medium	High	High
BWM (best–worst)	Subjective	High	Low	High	High
Entropy method	Objective	No	Low	Limited	Low
CRITIC	Objective	No	Medium	Limited	Medium
Gini	Objective	No	Medium	Limited	Medium
MEREC	Objective	No	High	Limited	High
CILOS	Objective	No	High	Limited	Medium
IDOCRIW	Integrated	No	Medium	Limited	Medium

Recent studies increasingly consider machine learning as a mechanism for deriving data-driven criterion weights and reducing dependence on direct expert weighting. Gürler et al. [20] used a genetic algorithm to determine criteria weights in MCDM methods and compared the resulting weights with CRITIC, entropy, and equal-weight baselines. Ahmed et al. [21] proposed a learning-based MCDM framework in which machine learning models derive objective weights from patterns in large datasets. Gidiagba et al. [22] integrated non-negative matrix factorization, random forest, and TOPSIS, using random forest to obtain data-driven criterion weights for supplier evaluation.

The integration of generative language models into MCDM is also becoming an active research direction. Wang et al. [23] proposed a general LLM-based framework for solving MCDM tasks and comparing LLM outputs with expert evaluations. Vahidnia [24] investigated multi-agent systems of large language models as weight assigners in spatial MCDM. Memduhoglu et al. [25] evaluated large language models as virtual experts for AHP-based criteria weighting in solar power plant site selection, while Sadeghi et al. [26]

combined AHP and ChatGPT for supplier selection and compared AI-generated judgments with human expert judgments.

A related stream uses LLMs to construct criteria, support pairwise comparisons, or build weighted decision models from unstructured information. Lu et al. [27] combined LLM reasoning with AHP for multi-criteria evaluation of open-ended responses. Wu et al. [28] proposed retrieval-augmented decision-making, where LLMs support the construction of weighted hierarchical decision models with traceable reasoning. Wu et al. [29] developed Doc2AHP, which infers structured AHP models from documents and introduces a multi-agent weighting mechanism with consistency optimization.

The reviewed studies show that both machine learning and generative language models can support criterion weighting, preference elicitation, and the automation of MCDM workflows. However, most existing approaches either directly elicit weights through expert or LLM pairwise comparisons, use machine learning as a component inside a specific ranking pipeline, or focus on constructing decision hierarchies from textual information. Less attention has been paid to the regression-based restoration of criterion weights from statistical characteristics of the distribution of criterion assessments. This gap determines the positioning of the present study.

At the same time, the use of LLM-generated weights should not be interpreted as a direct substitute for independently validated human expert judgments. In the present study, LLM outputs are used as a controlled synthetic benchmark that enables reproducible comparison of weighting methods under fixed generation and preprocessing conditions. This limitation also opens a future research direction connected with reinforcement learning from human feedback and cooperative reward-function optimization: in future real-world implementations, the reward function could be adapted to reflect dispatcher preferences and domain-specific expert feedback rather than relying only on synthetic reference weights.

Another limitation of the current line of weighting methods is that most aggregation procedures treat expert preferences as static. They usually aggregate already-formed opinions or weights, and do not explicitly model how expert opinions evolve, converge, or diverge within a professional community. Recent studies on adaptive opinion dynamics over community networks indicate that consensus formation may depend on community structure, communication constraints, and the difference between private and expressed opinions. Therefore, future extensions of regression-based criterion weighting may incorporate dynamic opinion evolution features, such as consensus stability, inter-community disagreement, and temporal changes in expert preferences, instead of relying only on static statistical characteristics of criterion assessments.

3. Research Gap and Study Objective

Thus, the existing scientific and methodological apparatus for assessing the significance of criteria does not allow for sufficiently accurate, rapid, and objective weighting of each criterion. Currently [2,3], there are studies in which the partial integration of artificial intelligence methods into multi-criteria decision-making opens up opportunities to move from subjective assessments to objectively calculated weights based on data.

This paper considers an approach in which machine learning (ML) is used to automatically determine the weighting coefficients of criteria in order to arrive at an accurate and objective decision when operating a decision support system for critical infrastructure dispatchers. This method will allow for weights to be formed from empirical data on preferences or results, eliminating subjective assumptions and increasing the validity of multi-criteria analysis. The use of ML in multi-criteria analysis has previously faced difficulties due to the lack of sufficient training data. To overcome this difficulty, the paper proposes the use of generative large language models, which made it possible to obtain

the necessary weight samples. The study aims to compare classical analytical methods for determining criterion weights and ML approaches, as well as to test the hypothesis that ML models are capable of predicting criterion weights based on the distribution of its estimates across a set of alternatives, as well as the nature of its statistical relationships with other criteria, and taking into account the structure of expert data.

To clarify the complete methodological logic of the study, Figure 1 presents the overall research framework from the formulation of the multi-criteria decision-making problem to model testing and statistical comparison.

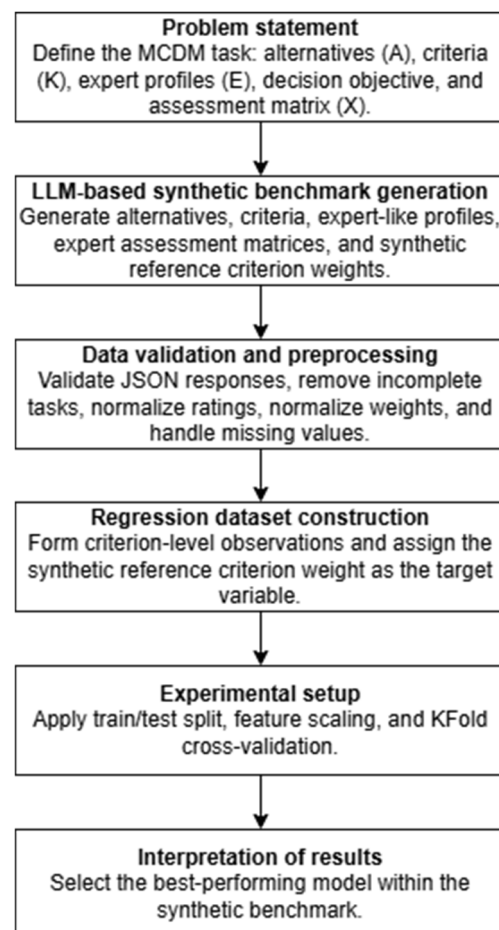


Figure 1. Methodological framework of the study.

The framework includes the definition of the decision problem, synthetic benchmark generation, data preprocessing, criterion-level feature extraction, regression dataset construction, model training, baseline method comparison, and final evaluation.

4. Method for Determining Criterion Weights

Let us write down the task of determining the weight coefficients of criteria in a multi-criteria evaluation model.

Let us assume that a set of alternatives is given:

$$A = \{A_1, A_2, \dots, A_M\}, \quad (1)$$

assessed according to a final set of criteria:

$$K = \{K_1, K_2, \dots, K_N\}. \quad (2)$$

Expert assessments are presented in the form of a matrix:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N} \\ x_{2,1} & x_{2,2} & \dots & x_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M,1} & x_{M,2} & \dots & x_{M,N} \end{pmatrix}, \quad (3)$$

where m, n is the value of the m -th alternative according to the n -th criterion.

It is necessary to determine the vector of weight coefficients:

$$w = (w_1, w_2, \dots, w_N). \quad (4)$$

Satisfying the conditions of non-negativity and normalization:

$$w_n \geq 0, \sum_{n=1}^N w_n = 1. \quad (5)$$

This methodology compares a set of complementary statistical characteristics, which significantly distinguishes it from classical methods. The methodology allows for criteria to be represented as a multidimensional feature space and formalizes the task of assigning weight coefficients as a regression analysis task.

For each criterion, a multidimensional feature vector is formed based on the statistical characteristics of its estimates for all alternatives. The features used include indicators of central tendency, variability, distribution shape, correlations with other criteria, and information richness.

For each criterion K_n , the vector of its estimates is considered based on the statistics in Table 2.

Thus, each criterion K_n is assigned a vector of features:

$$z_n = (\mu_n, \sigma_n, x_n^{\min}, x_n^{\max}, R_n, CV_n, \tilde{x}_n, Q_{0.25,n}, Q_{0.75,n}, IQR_n, Skew_n, Kurt_n, \tilde{\rho}_n, \rho_n^{\max}, H_n) \quad (6)$$

In the proposed framework, each criterion-level feature vector is mapped to a criterion weight by a regression model. In the linear case, the following relationship is considered:

$$w_n = z_n^T \beta + \xi_n, \quad (7)$$

where β is the model parameter vector, ξ_n is the random error.

The linear specification is used in this study as an interpretable exploratory benchmark rather than as an assumption that all relationships between criterion-level features and weights are strictly linear. The selected feature vector includes distributional, correlation-based, and entropy-based characteristics, and some of their effects on criterion weights may be non-linear or mutually dependent. Therefore, Ridge regression is treated as a regularized first-order approximation that is suitable for the small synthetic benchmark considered in this paper. More flexible non-linear approaches will be considered in future work when larger real-world datasets become available.

Several regression models were considered in the experimental comparison: Ridge regression, Lasso regression, Random Forest regression, and Gradient Boosting regression. Lasso regression was treated as one of the candidates regularized linear models rather than as the only estimation method. For Lasso, the optimization problem is written as

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{n=1}^N (w_n - z_n^T \beta)^2 + \lambda \sum_{j=1}^P |\beta_j|, \quad (8)$$

where P is the dimension of the feature vector, λ is the regularization parameter.

Table 2. Statistics for generating features in the weight coefficient assignment model.

Key Statistical Characteristics	
Mathematical expectation	$\mu_n = \frac{1}{M} \sum_{m=1}^M x_{m,n}$
Standard deviation	$\sigma_n = \sqrt{\frac{1}{M} \sum_{m=1}^M (x_{m,n} - \mu_n)^2}$
Minimum and maximum	$x_n^{\min} = \min_m x_{m,n}, x_n^{\max} = \max_m x_{m,n}$
Range	$R_n = x_n^{\max} - x_n^{\min}$
Coefficient of variation	$CV_n = \frac{\sigma_n}{ \mu_n + \varepsilon}$
Quantile Characteristics	
Median	$\widetilde{x}_n = \text{median}(x_n)$
First and third quartiles	$Q_{0.25n}, Q_{0.75n}$
Interquartile range	$IQR_n = Q_{0.75n} - Q_{0.25n}$
Characteristics of the Distribution Form	
Coefficient of asymmetry	$Skew_n = \frac{1}{M} \sum_{m=1}^M \left(\frac{x_{m,n} - \mu_n}{\sigma_n} \right)^3$
Excess coefficient	$Kurt_n = \frac{1}{M} \sum_{m=1}^M \left(\frac{x_{m,n} - \mu_n}{\sigma_n} \right)^4 - 3$
Correlation Characteristics	
Average correlation	$\overline{\rho}_n = \frac{1}{N-1} \sum_{k \neq n} \rho_{n,k} $
Maximum correlation	$\rho_n^{\max} = \max_{k \neq n} \rho_{n,k} $
Entropy Characteristic	
Entropy criterion	$H_n = -\frac{1}{\ln M} \sum_{m=1}^M \rho_{m,n} \ln \rho_{m,n}$

The obtained weight values are normalized:

$$\widetilde{w}_n = \frac{\max(w_n, \varepsilon)}{\sum_{k=1}^N \max(w_n, \varepsilon)}, \quad (9)$$

where ε is the numerical threshold for normalization. The method is illustrated in Figure 2.

As a result of applying the proposed methodology, each criterion of the multi-criteria model is assigned a quantitatively justified weight coefficient that reflects its relative importance in solving the multi-criteria optimization problem. The use of statistical characteristics of the distribution of expert assessments allows us to take into account not only the average level of criterion values, but also their variability, asymmetry, interrelationships with other criteria, and information richness.

It is worth noting that training is carried out at the level of criteria, not alternatives, which fundamentally distinguishes this approach from classical MCDM methods.

The resulting weight coefficients are resistant to noise in the source data and are formed automatically based on the objective properties of the data, which ensures the reproducibility of results and increases the reliability of the final assessments in conditions of a limited training sample.

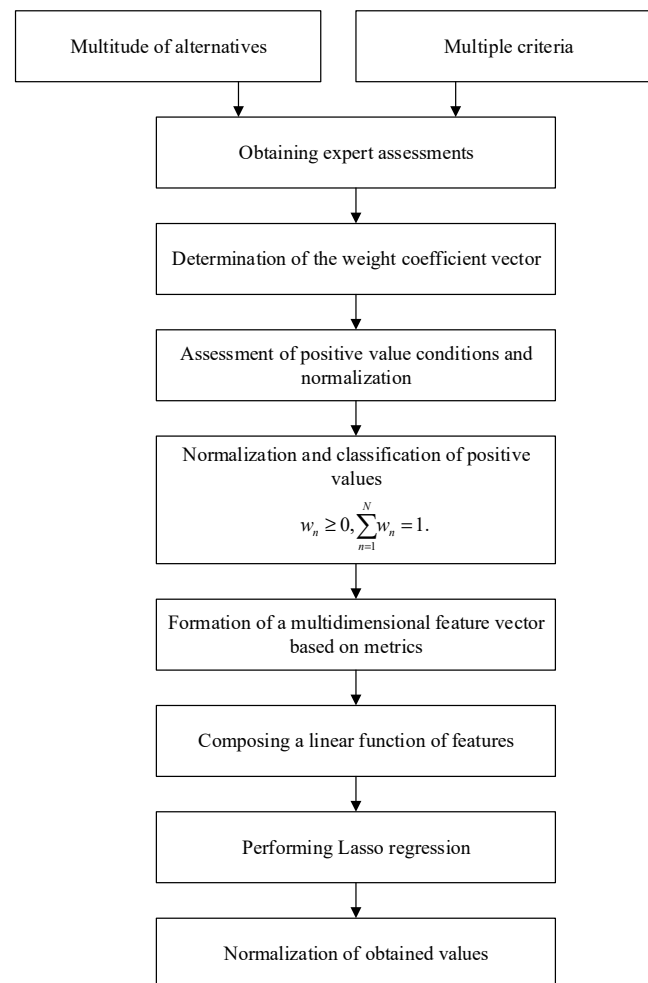


Figure 2. Scheme of the method for determining criterion weights.

To obtain a reproducible experimental dataset, the proposed approach uses an LLM as a controlled generator of synthetic MCDM scenarios, expert-like profiles, alternative assessments, and synthetic reference weight samples. The LLM outputs are not treated as real human expert judgments. Instead, they are used to construct a synthetic benchmark that allows different weighting methods to be compared under fixed generation and preprocessing conditions. Therefore, throughout the experiment, the target weights are interpreted as synthetic reference weights rather than synthetic reference weights obtained from real expert panels.

To configure the LLM-based generator, the input data are formed in a structured way. The generation procedure fixes the domain, decision objective, number of alternatives, number of criteria, number of expert profiles, model parameters, and response format. This reduces parsing errors and improves the reproducibility of the synthetic data-generation process.

5. Data Formation

The first step is to set the generation parameters: M (number of alternatives), N (number of criteria), K (number of experts), as well as the domain and objective text fields. Additionally, the LLM call parameters (model and temperature) are fixed, which ensures the reproducibility of the generation conditions at the configuration level.

Thus, matrices of estimates x_{kij} are formed for each pair (a_i, c_j) for each expert e_k , where $A = \{a_i\}_{i=1}^N$ is the set of alternatives, $C = \{c_j\}_{j=1}^M$ is the set of criteria, $E = \{e_k\}_{k=1}^K$ is the set of experts.

LLM generates lists of alternatives and criteria of strictly defined power:

$$|A| = M, |C| = N \quad (10)$$

To ensure the stability of the results, a JSON response is used. Using a query and keywords (prompt), a set of experts $|E| = K$ is formed, where each expert is described by attributes (identifier, role, experience, competence, reasoning style), which are then used as conditions when obtaining ratings. In practice, this implements controlled heterogeneity of opinions within the group.

Next, for each expert e_k , a query to the model generates an $N \times M$ rating matrix, where the elements x_{kij} lie on a limited scale $[0; 100]$. Generation is performed using the same JSON schema mechanism, which reduces the proportion of erroneous responses and simplifies parsing. Next, the results are brought to a fixed order of alternatives and indicators.

A separate request and keywords for the model form a reference weight vector w based on a list of criteria and the context of the task. The weights are normalized, $\sum_j w_j = 1$, and a textual justification is saved for each weight. The data obtained is used for experiments on weight restoration/reconciliation or for evaluating the quality of ranking.

LLM in this generator is not just a source of text, but a formal participant in the weight assignment procedure, i.e., it models the cognitively and professionally conditioned decisions of a group of experts with a fixed task setting. This is achieved by sequentially assigning the model different functional roles: first, it synthesizes the choice space (alternatives and criteria), then generates a heterogeneous group of expert persons, and finally, on behalf of each such expert, produces numerical evaluations of alternatives for each criterion on a common scale. Accordingly, LLM implements three levels of substantive plausibility validation at once:

- Semantic consistency of entities with the domain and objective context;
Plausible variability of opinions among experts due to differing profiles (role, experience, competence, and style);
- Numerical certainty of assessments sufficient for subsequent aggregation and comparison of group multi-criteria selection methods.

At the prompting level, this is expressed by the fact that the generation of alternatives and indicators is set as a separate stage of data setting.

The generation procedure includes several safeguards aimed at improving procedural consistency. First, the LLM responses are requested in a structured JSON format. Second, generated alternatives, criteria, expert profiles, ratings, and weights are checked for completeness. Third, tasks with empty expert score matrices or missing criterion weights are excluded from the dataset. Fourth, expert scores generated on the $([0, 100])$ scale are converted to the $([0, 1])$ interval before feature extraction. Fifth, expert weight vectors are normalized, and missing values are checked and handled during preprocessing.

These safeguards reduce technical inconsistencies in the synthetic dataset. However, they do not prove equivalence between LLM-generated judgments and real human expert judgments. Since the current study does not include independent real expert panels, the LLM-generated data should be interpreted only as a reproducible synthetic benchmark for comparing methods.

6. Results

The initial LLM-generated dataset contained 25 decision-making tasks. During preprocessing, three tasks with empty expert score matrices were excluded. Thus, 22 valid tasks remained after filtering. In the reported experiment, the first domain setting was used; therefore, the model was trained and evaluated on one domain: cloud DWH/BI platform selection. This domain contained 7 expert profiles, 5 alternatives, and 6 criteria.

For each expert and each criterion, a 15-dimensional feature vector was extracted. The features included mean, standard deviation, minimum, maximum, range, coefficient of variation, median, first and third quartiles, interquartile range, skewness, kurtosis, average correlation with other criteria, maximum absolute correlation, and entropy. Since 7 expert profiles and 6 criteria were used, the resulting regression dataset contained 42 criterion-level observations. The target variable was the synthetic reference criterion weight. It was obtained by aggregating LLM-generated expert weights using competence-weighted averaging:

$$w_j = \sum_{e=1}^k \left(\frac{\text{competence}_e}{\sum \text{competence}_e} w_{e,j} \right) \quad (11)$$

The target variable in the regression experiment is the synthetic reference criterion weight. It is obtained by competence-weighted aggregation of the LLM-generated expert weight vectors. The MSE metric is used to evaluate how accurately each method restores these synthetic reference weights. The software complex is implemented in Python using the LangChain and openai libraries and the openai API to generate task experts and alternatives, as well as sklearn for training machine learning models. The implementation is available in the generator repository (https://github.com/komaroval/Expert_dataset_generator (accessed on 20 June 2026)).

The implementation diagram is shown in Figure 3.

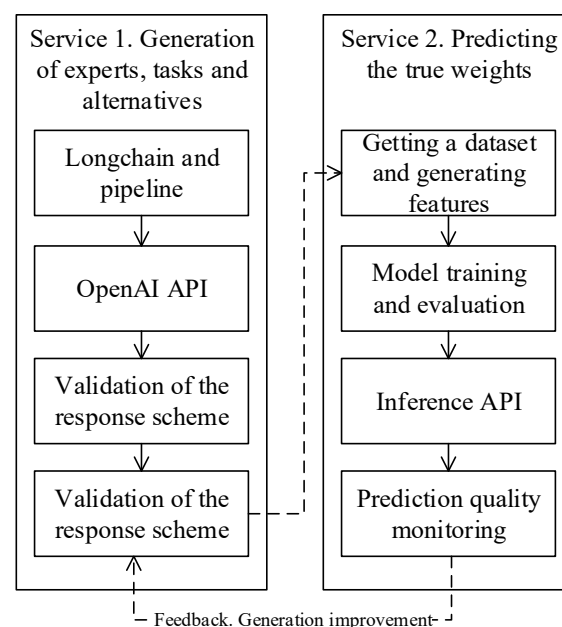


Figure 3. Software package implementation diagram.

The diagram shows the two-service architecture of the Python 3.8 software complex. Service 1 is responsible for generating source data: the orchestrator launches a pipeline implemented on the LangChain library, which forms prompts and calls the OpenAI API to create expert profiles, tasks, and alternatives. The results are validated and normalized (response format is checked), then saved and collected into a single dataset. Service 2 takes

this dataset and performs the ML part: first, the data are converted into features, then in the training and evaluation block, models that restore synthetic reference weights are trained using the scikit-learn library. An inference API can be used for practical applications. The cycle is completed by quality monitoring, the results of which can be used as feedback to improve generation and validation rules in Service 1.

The data for the experiment were divided into training and test subsets using a 70/30 split with `random_state = 42`. Feature scaling was performed using `StandardScaler`, which was fitted only on the training subset and then applied to the test subset. Cross-validation was performed on the training data using 5-fold KFold cross-validation with shuffling.

The hyperparameters were fixed before the reported comparison and were not selected by exhaustive grid search. Ridge regression used $\alpha = 1.0$, Lasso regression used $\alpha = 0.001$ with `max_iter = 5000`, Random Forest used `n_estimators = 50` and `max_depth = 3`, and Gradient Boosting used `n_estimators = 50` and `max_depth = 2`. These values were chosen as conservative baseline settings for an exploratory experiment with a small criterion-level sample, where strong model complexity could lead to overfitting. In particular, Ridge $\alpha = 1.0$ represents a standard L2-regularized baseline, while shallow tree-based models were used to restrict model variance.

The experiments were conducted in a Google Colab notebook environment. According to the available notebook information, the runtime used a CPU environment with 107 GB RAM and 12.7 GB available disk space. The notebook used NumPy, pandas, SciPy, scikit-learn, matplotlib, and seaborn for preprocessing, feature extraction, model training, metric calculation, and visualization. The LLM-generated dataset was loaded from Google Drive.

Four regression models were compared: Ridge regression, Lasso regression, Random Forest regression, and Gradient Boosting regression, as shown in Table 3.

Table 3. Metrics for determining weights using ML models.

Model	MSE_cv	MSE_test
Ridge	0.00780	0.00370
Lasso	0.00819	0.00405
RandomForest	0.00953	0.00384
GradientBoosting	0.01100	0.00457

Based on the results of the experiment, a conclusion is made about the effectiveness of the Ridge model for predicting criterion weights. Next, an experiment was conducted to calculate weights using various objective weighting methods, as shown in Table 4.

Table 4. Average quality of each method according to all experts.

Method	MSE	MAE	MAPE	Spearman	Rank _{MAE}	Rank _{Spearman}
Removal Effects Weighting Method	0.0047	0.0599	88.86	0.9714	0.1143	0.9714
Angular Deviation Weighting Method	0.0074	0.0681	78.05	0.9714	0.1143	0.9429
Gini Coefficient Weighting Method	0.0077	0.0704	86.83	0.9571	0.1714	0.9571
Coefficient of Variation Weighting Method	0.0078	0.0706	87.02	0.9571	0.1714	0.9571
Standard Deviation Weighting Method	0.0082	0.0731	88.18	0.9714	0.1143	0.9714
Entropy Weight Method	0.0471	0.1854	481.03	0.9571	0.1143	0.9714
Correlation-Based Weighting Method	1.1039	0.8427	2106.14	−0.4429	1.8295	−0.4429

The next step was to compare the best model with existing analytical methods: the averaging method and the combined estimation method, as shown in Table 5.

Table 5. Metrics for determining weights using different methods.

Method	MSE	MAE	MAPE
ML (Ridge)	0.004	0.056	103.64
Averaging	0.013	0.089	259.52
Combined	0.009	0.076	211.85

The averaging method is calculated as the arithmetic mean of the weight vectors obtained by the selected objective weighting methods:

$$w_j^{avg} = \frac{1}{S} \sum_{s=1}^S w_j^{(s)}, \quad (12)$$

where $w_j^{(s)}$ is the weight of criterion j obtained by method s , and S is the number of objective weighting methods included in the comparison.

The combined method was reproduced from the study [30], in which the weight vectors obtained by several objective weighting methods are first transformed into a rank matrix. The agreement between these rankings is then evaluated using Kendall's coefficient of concordance, while statistical significance is checked using Pearson's chi-square criterion. Since the use of all objective methods does not necessarily provide the required level of concordance, the method selects a subset of methods that satisfies the predefined concordance and significance constraints. The final criterion weights are calculated from the normalized sums of ranks within the selected subset, which makes it possible to obtain weights with a specified level of agreement and to increase the objectivity of the resulting multi-criteria decision model.

Based on the results, we can conclude that ML (Ridge) shows better accuracy in weights by 30% compared to averaging estimates and 57% lower compared to the combined approach. This difference is statistically significant (p -value = 0.0016). A graph has also been constructed for the visual comparison of all methods, as shown in Figure 4.

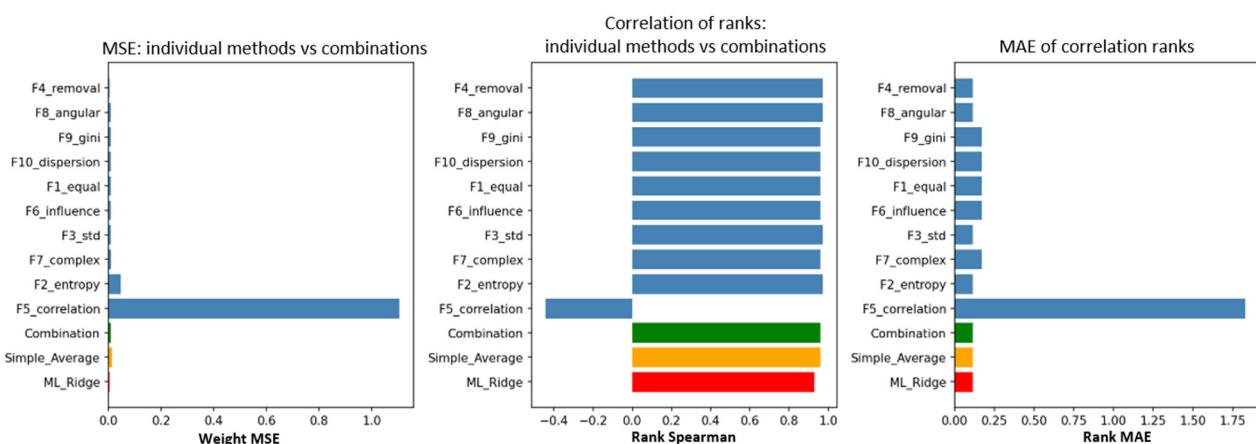


Figure 4. Visual comparison of all methods.

The results (Figure 4, left graph) show that using individual analytical criteria leads to high weight recovery error. The best quality is achieved when using the ML approach, which integrates several statistical characteristics of the criteria simultaneously. Even if some methods give an acceptable correlation, their behavior is unstable (Figure 4, center

graph). The minimum MAE value of the ML approach (Figure 4, right graph) indicates its practical advantage in selecting alternatives, especially in top-k selection tasks.

An additional experiment was conducted to determine the effect of sample size on the accuracy of weight determination by the model. The dependence of MSE weight on sample size is shown in Figure 5.

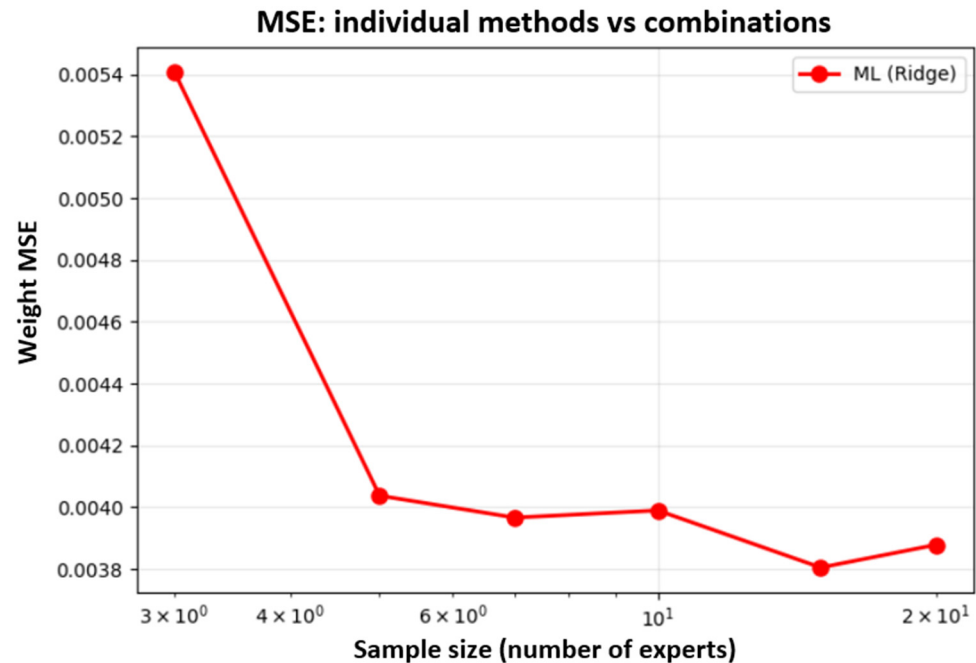


Figure 5. Example of an expert profile.

The graph (Figure 5) demonstrates that the ML approach to determining criterion weights benefits from an increase in data volume. The model becomes significantly more accurate in determining the weights themselves (MSE decreases) and slightly improves its ability to correctly rank alternatives (Rank Spearman increases) as more examples are obtained from experts. Based on the graph, we conclude that, within the considered LLM-generated synthetic benchmark, the regression-based ML approach benefits from an increase in the number of criterion-level observations. However, this result should be interpreted as evidence for the reported synthetic setting only. Additional validation on real-world MCDM datasets is required before making general conclusions about the suitability of ML models for practical expert-based weight assignment.

7. Limitations

The study has several limitations that should be taken into account when interpreting the results. First, the empirical evaluation is based on an LLM-generated synthetic dataset. The generated expert profiles, assessments, and criterion-weight samples are used as a reproducible benchmark for comparing methods, but they are not equivalent to independently collected judgments from real human experts. Therefore, the results should not be interpreted as direct evidence that the proposed model will generalize to real expert decision-making settings.

Second, the current study does not include validation on a real-world MCDM dataset with independently obtained expert weights. Such validation would require selecting or collecting an applied decision-making case, reconstructing its decision matrix, obtaining reliable expert weights, extracting criterion-level features, and comparing predicted weights with independent reference weights. This remains an important direction for future work.

Third, the main regression experiment was conducted on one selected domain: cloud DWH/BI platform selection. This domain contained 7 expert profiles, 5 alternatives, and 6 criteria, which produced 42 criterion-level observations. Although this setting is sufficient for an exploratory comparison of methods, it remains limited in terms of domain diversity and sample size. Future experiments should include multiple domains, larger numbers of alternatives and criteria, and out-of-domain validation.

Fourth, the train/test split was performed at the criterion-feature observation level. This means that the reported evaluation measures the model's ability to restore synthetic reference weights within the selected domain and data-generation setting. It does not represent a strict out-of-domain test where the model is trained on one set of decision problems and evaluated on entirely different domains.

Fifth, several features used in the model are scale-dependent, including the mean, standard deviation, minimum, maximum, range, and interquartile range. In the current experiment, expert ratings were generated on the $([0, 100])$ scale and then converted to the $([0, 1])$ interval. Robustness to other measurement scales, such as Likert-scale ratings or unbounded utility values, was not tested. Future work should compare alternative normalization strategies, including min-max normalization, z-score standardization, rank transformation, and scale-invariant feature engineering.

Finally, the reported p -value should be interpreted as exploratory. The paired t -test was conducted only for the comparison between ML Ridge and the optimized combined method over 7 expert profiles, and no correction for multiple comparisons was applied. Therefore, the statistical result supports the reported experimental comparison but should not be treated as confirmatory evidence of universal superiority.

8. Conclusions

The paper proposes and experimentally investigates an approach to determining criterion weights in multi-criteria decision support tasks based on regression analysis and LLM-generated synthetic data. Unlike classical analytical and expert methods, the proposed methodology formalizes the task of assigning weights as a criterion-level regression problem, in which each criterion is represented by a multidimensional vector of statistical characteristics calculated from the distribution of its assessments across alternatives.

The reported experiment shows that, within the considered synthetic benchmark, the Ridge regression model achieved the lowest test MSE among the compared ML models and provided lower MSE and MAE than the averaging and optimized combined baselines. The obtained results suggest that regularized regression models, especially Ridge regression in the reported synthetic benchmark, may be useful for estimating synthetic reference criterion weights from statistical characteristics of criterion assessments.

However, the current study should be interpreted as an exploratory benchmark rather than as evidence of general operational validity. Further research should validate the approach on real-world MCDM datasets with independently obtained expert weights, evaluate robustness across multiple domains and rating scales, and investigate adaptive hyperparameter optimization, CoRLHF-based preference alignment, non-linear fuzzy-topological regression models, and dynamic opinion evolution features.

Author Contributions: Conceptualization, V.A.S. and A.V.T.; methodology, V.A.S. and A.V.T.; software, L.A.K.; validation, A.S.Z.; formal analysis, A.S.Z.; investigation, V.A.S. and A.V.T.; resources, V.A.S. and A.V.T.; data curation, V.A.S.; writing—original draft preparation, A.S.Z.; writing—review and editing, L.A.K.; visualization, V.A.S.; supervision, V.A.S.; project administration, A.V.T.; funding acquisition, A.S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used gpt-4o-mini for the purposes of task generation and random experts. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

$A = \{A_1, A_2, \dots, A_M\}$	Set of alternatives
$K = \{K_1, K_2, \dots, K_N\}$	Set of criteria
$E = \{E_1, E_2, \dots, E_N\}$	Set of expert profiles
X	Decision matrix containing assessments of alternatives according to criteria
$w = (w_1, w_2, \dots, w_N)$	Vector of criterion weights
zn	Feature vector describing criterion Kn
P	Number of features in the criterion-level feature vector
β	Vector of regression coefficients
λ	Regularization parameter
ε	Small numerical threshold used for normalization and division stability
MCDM	Multi-criteria decision making
LLM	Large language model
ML	Machine learning
CIF	Critical infrastructure facility
DSM	Decision support method
AHP	Analytic hierarchy process
MSE	Mean squared error
MAE	Mean absolute error
MAPE	Mean absolute percentage error
Spearman	Spearman rank correlation coefficient

References

1. Nguyen, V.T.T.; Vo, N.T.M.; Truong, V.C.; Nguyen, V.-T. *Multi-Criteria Decision-Making and Optimum Design with Machine Learning: A Practical Guide*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2024. [\[CrossRef\]](#)
2. Jia, W. Research on Intelligent Financial Decision Support System Based on Data Mining. In Proceedings of the 2024 5th International Conference on Computer Science and Management Technology, Xiamen, China, 18–20 October 2024; pp. 173–178. [\[CrossRef\]](#)
3. Jia, T.; Wang, C.; Tian, Z.; Wang, B.; Tian, F. Design of Digital and Intelligent Financial Decision Support System Based on Artificial Intelligence. *Comput. Intell. Neurosci.* **2022**, 2022, 1962937. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Michalski, M.A.D.C.; Murad, C.A.; Kashiwagi, F.N.; De Souza, G.F.M.; Da Silva, H.J.B.; Côrtes, H.M. A Multi-Criteria Framework for Selecting Machine Learning Techniques for Industrial Fault Prognosis. *IEEE Access* **2025**, *13*, 154508–154544. [\[CrossRef\]](#)
5. Cheng, W.; Ahmad, H.; Gao, L.; Xing, J.; Nie, Z.; Chen, X.; Xu, Z.; Zhang, R. Diagnostics and Prognostics in Power Plants: A Systematic Review. *Reliab. Eng. Syst. Saf.* **2025**, *255*, 110663. [\[CrossRef\]](#)
6. Pratiwi, D.H.K.; Sujoko. Systematic Review of Leadership Types Impacting Decision Support System Effectiveness Across Life Cycle Phases. *J. Ris. Ekon. Manaj. (REKOMEN)* **2025**, *8*, 619–630. [\[CrossRef\]](#)
7. Achimugu, D.; Ukatu, C.; Anaege, A.E. The Ethics of AI in Financial Planning: Bias, Transparency, and the Role of Human Judgment. *Int. J. Res. Sci. Innov.* **2025**, *12*, 1–15. [\[CrossRef\]](#)
8. Gaddapuri, N.S. Cognitive Regulatory Engines: Integrating Explainable AI (XAI) in Real-Time Compliance Platforms for Financial Institutions. *QIT Press-Int. J. Comput. Sci. Eng. Res. Dev.* **2025**, *5*, 1–15. [\[CrossRef\]](#)
9. Khaliq, Z.; Farooq, S.U.; Khan, D.A. Artificial Intelligence in Software Testing: Impact, Problems, Challenges and Prospect. *arXiv* **2022**. [\[CrossRef\]](#)
10. Jankova, Z.; Dostal, P. Type-2 fuzzy expert system Ap-proach for decision-making of financial assets and investing under different uncertainty. *Math. Probl. Eng.* **2021**, 2021, 3839071.

11. Zhao, Y. Decision Support System for Economic Management of Large Enterprises Based on Artificial Intelligence. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9453580. [\[CrossRef\]](#)
12. Yasentsev, D.; Shevgunov, T.; Efimov, E.; Tatarskiy, B. Using Ground-Based Passive Reflectors for Improving UAV Landing. *Drones* **2021**, *5*, 137. [\[CrossRef\]](#)
13. Černevičienė, J.; Kabašinskas, A. Explainable Artificial Intelligence (XAI) in Finance: A Systematic Literature Review. *Artif. Intell. Rev.* **2024**, *57*, 216. [\[CrossRef\]](#)
14. Alam, A.; Ahamad, M.K. K-Means Hybridization with Enhanced Firefly Algorithm for High-Dimension Automatic Clustering. *J. Adv. Res. Appl. Sci. Eng. Technol.* **2023**, *33*, 137–153. [\[CrossRef\]](#)
15. Canco, I.; Kruja, D.; Iancu, T. AHP, a Reliable Method for Quality Decision Making: A Case Study in Business. *Sustainability* **2021**, *13*, 13932. [\[CrossRef\]](#)
16. Dahooie, J.H.; Kashan, A.H.; Naeini, Z.S.; Vanaki, A.S.; Zavadskas, E.K.; Turskis, Z. A Hybrid Multi-Criteria-Decision-Making Aggregation Method and Geographic Information System for Selecting Optimal Solar Power Plants in Iran. *Energies* **2022**, *15*, 2801. [\[CrossRef\]](#)
17. Ayan, B.; Abacioğlu, S.; Babilio, M.P. A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision-Making. *Information* **2023**, *14*, 285. [\[CrossRef\]](#)
18. Yang, G.; Yang, J.-B.; Xu, D.; Khoveyni, M. A Three-Stage Hybrid Approach for Weight Assignment in MADM. *Omega* **2017**, *71*, 93–105. [\[CrossRef\]](#)
19. Chang, K.-H. Integrating Subjective–Objective Weights Consideration and a Combined Compromise Solution Method for Handling Supplier Selection Issues. *Systems* **2023**, *11*, 74. [\[CrossRef\]](#)
20. Gürler, H.E.; Özçalıcı, M.; Pamucar, D. Determining criteria weights with genetic algorithms for multi-criteria decision making methods: The case of logistics performance index rankings of European Union countries. *Socio-Econ. Plan. Sci.* **2024**, *91*, 101758. [\[CrossRef\]](#)
21. Ahmed, M.; Dogru, A.; Zhang, C.; Meng, C. Learning-Based Multi-Criteria Decision Model for Site Selection Problems. *arXiv* **2025**, arXiv:2504.04055. [\[CrossRef\]](#)
22. Gidiagba, O.J.; Tartibu, L.; Okwu, M. Integrating Machine Learning with Multi-Criteria Decision-Making Models for Sustainable Supplier Selection in Dynamic Supply Chains. *Logistics* **2025**, *9*, 152. [\[CrossRef\]](#)
23. Wang, H.; Zhang, F.; Mu, C. One for All: A General Framework of LLMs-based Multi-Criteria Decision Making on Human Expert Level. *arXiv* **2025**, arXiv:2502.15778. [\[CrossRef\]](#)
24. Vahidnia, M.H. Multi-agent systems of large language models as weight assigners: An approach to collaborative weighting in spatial multi-criteria decision-making. *Geomatica* **2025**, *77*, 100071. [\[CrossRef\]](#)
25. Memduhoglu, A.; Fulman, N.; Polat, N.; Atas, T. Large language models as virtual experts? Evaluating AHP-based criteria weighting performance for solar power plant site selection. *Expert Syst. Appl.* **2026**, *299*, 130171. [\[CrossRef\]](#)
26. Sadeghi, N.; Dehghani, M.; Kabadayi, N. The AI-driven Decision-Making (AIDM) Framework: Integrating AHP and ChatGPT for Supplier Selection. *Ann. Oper. Res.* **2026**, *359*, 2593–2615. [\[CrossRef\]](#)
27. Lu, X.; Li, J.; Takeuchi, K.; Kashima, H. AHP-Powered LLM Reasoning for Multi-Criteria Evaluation of Open-Ended Responses. *arXiv* **2024**, arXiv:2410.01246. [\[CrossRef\]](#)
28. Wu, H.; Xia, J.; Zhang, H.; Chen, W. Retrieval Augmented Decision-Making: A Requirements Engineering Approach to Industrial Decision Support. *arXiv* **2025**, arXiv:2505.18483. [\[CrossRef\]](#)
29. Wu, H.; Zhou, S.; Zhang, H.; Chen, W. Doc2AHP: Inferring Structured Multi-Criteria Decision Models via Semantic Trees with LLMs. *arXiv* **2026**, arXiv:2601.16479. [\[CrossRef\]](#)
30. Solovjev, D.S. The objectification method of the weight coefficients for decision-making in multicriteria problems. *Sci. Tech. J. Inf. Technol. Mech. Opt.* **2023**, *23*, 161–168. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.