MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1180-1188

Improving Cost Visibility in Agile (Software) Projects Through a Fuzzy AHP–Q-Learning Prioritization Framework

Jitesh R. Neve[*1], Dr. Sohit Agarwal[2]

[1]*Research Scholar, Dept of Computer Engineering and IT, Suresh Gyan Vihar University, India, jiteshneve@gmail.com, ORCID: https://orcid.org/0009-0009-2358-1147*
[2]*Professor and Head, Dept of Computer Engineering and IT, Suresh Gyan Vihar University, India, sohit.agarwal@gmail.com, ORCID: https://orcid.org/0000-0002-1280-7907*

Abstract

The evolution of project management methodologies has significantly impacted software development, particularly due to its adaptable and cyclical approach. Nonetheless, it introduces distinct difficulties in effectively recognizing and prioritizing cost overheads. Conventional cost estimation methods frequently prove inadequate in agile settings because of their rigid assumptions and failure to accommodate uncertainty and evolving conditions. This paper presents an enhanced classification framework aimed at prioritizing Agile Cost Overhead. It integrates the Fuzzy Analytic Hierarchy Process (Fuzzy AHP) with Q-Learning Optimization to effectively tackle existing limitations. The Fuzzy AHP method facilitates a systematic approach to prioritizing cost factors by leveraging expert insights and linguistic preferences, effectively addressing the inherent ambiguity present in agile processes. In addition to this, Q-Learning—an adaptive reinforcement learning method—enhances prioritization by learning from both historical and real-time data, consistently improving decisions through feedback driven by rewards. The hybrid framework is proposed by implementing a Voting Classifier that combines the predictions of Random Forest (RF) and Support Vector Machine (SVM), where the soft voting mechanism is applied for taking the final predictions. It categorizes overhead elements based on their impact (high, medium, and low) and adapts in real-time to project conditions, enhancing cost visibility and facilitating proactive decision-making. This model assists project managers in pinpointing essential cost factors and optimizing resource allocation. Assessment via simulated agile scenarios reveals improved prioritization precision and flexibility in contrast to traditional approaches. The proposed framework effectively connects expert intuition with intelligent learning, providing a scalable, data-driven solution for managing cost overhead in both software and non-software domains in an agile manner.

Keywords: *Enhanced classification framework, agile cost overhead, Fuzzy Analytic Hierarchy Process, Q-Learning*

I. INTRODUCTION

In the evolving field of software engineering, agile approaches have emerged as a prevailing paradigm for iterative and customer-focused product development. Agile methodologies, distinguished by their adaptability, teamwork, and fast delivery intervals, have been extensively embraced across several sectors to address changing business requirements. As businesses expand their agile implementations, they often face a considerable challenge: the precise calculation and prioritization of cost overheads linked to agile initiatives [1], [2]. Cost overheads in agile development may originate from several factors, such as rapid changes in requirements, team reconfigurations, updates to tools, communication delays, and training requirements. If inadequately managed and prioritized, these cost overheads may negatively impact project budgeting, timing, and overall product quality [3], [4]. Conventional cost estimating and prioritizing methods, including expert opinion, historical analysis, and deterministic models, are inadequate in agile organizations. These approaches often presume static inputs and linear connections, which are insufficient to represent the intrinsic uncertainties, interdependencies, and adaptive behaviors characteristic of agile systems. As a result, there is an increasing need for sophisticated, data-driven frameworks capable of managing ambiguity and dynamically prioritizing cost factors in agile projects [5]. This paper offers an Enhanced Classification Framework for prioritizing Agile Cost Overhead, which merges Fuzzy Analytic Hierarchy Process (Fuzzy AHP) with Q-Learning Optimization, a reinforcement learning method. The suggested framework utilizes the advantages of fuzzy multi-criteria decision-making and adaptive learning to provide a resilient, scalable, and intelligent solution for agile cost overhead control. Fuzzy AHP is an enhancement of the conventional Analytic Hierarchy Process (AHP), a recognized decision-making instrument that structures complex issues into a hierarchy of subordinate problems [6]. Fuzzy AHP utilizes fuzzy logic to manage inaccurate and subjective data, often seen in agile contexts where stakeholders provide qualitative evaluations instead of quantitative measures. In this context, Fuzzy AHP is used to create a prioritized hierarchy of agile cost overhead elements grounded on expert judgment and linguistic assessments. It enables decision-makers to articulate their preferences in ambiguous phrases (e.g., "moderately more significant," "strongly less significant"), which are then converted into quantifiable fuzzy weights for each cost criterion.

Notwithstanding its advantages, Fuzzy AHP alone lacks the capacity to evolve in response to input or changing situations inside agile contexts [7], [8]. This is the point at

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

which Q-Learning Optimization improves the model. Q-Learning, a kind of model-free reinforcement learning, allows the system to acquire optimum behaviors via interaction with the environment and the receipt of rewards or penalties. When combined with the Fuzzy AHP outcomes, Q-Learning may enhance the prioritizing of agile cost overheads via ongoing learning and optimization. It assesses many decision pathways, encourages effective tactics, and dynamically adjusts to changes in project needs, team performance, or corporate priorities. This renders the total system robust and astute, ready to adapt alongside the agile lifecycle [9], [10], [11].

This study presents many major contributions. Initially, it presents an innovative hybrid methodology that integrates the interpretative capabilities of Fuzzy AHP with the flexibility of Q-Learning to provide a thorough classification and prioritizing framework. Secondly, the paradigm facilitates the conversion of subjective stakeholder insights into measurable decision-making parameters, thereby reducing uncertainty in cost overhead assessment. Third, with the use of Q-Learning, the model adapts based on past performance data and real-time project input, resulting in enhanced prioritizing methods that are more informed and more successful. The framework facilitates improved categorization by classifying overhead items into several effect levels (e.g., high, medium, low) with optimized fuzzy weights and established decision criteria.

The suggested framework may be effectively integrated into agile project management systems to facilitate strategic decision-making, cost estimation, and resource distribution. A project manager might use the system to pinpoint which cost drivers—such as sprint delays, backlog grooming sessions, or rework expenses—require prompt intervention, and how these drivers influence total project delivery and budget. The framework may function as a decision-support tool for Agile Project Management Offices (PMOs), assisting them in assessing project health and prioritizing interventions across various agile teams. The ramifications of this study extend beyond software development alone. With the growing use of agile methodologies in non-software sectors like manufacturing, marketing, and construction, the suggested framework provides a comprehensive strategy for managing cost-related risks in any agile-enabled process. The amalgamation of fuzzy logic with reinforcement learning represents a pivotal advancement in intelligent decision support systems that merge human intuition with machine learning. This study introduces an Enhanced Classification Framework for prioritizing Agile Cost Overhead, integrating Fuzzy AHP with Q-Learning Optimization to rectify the deficiencies of conventional cost management methods in agile projects. The framework improves project efficiency by facilitating a sophisticated, adaptable, and data-driven comprehension of cost overheads, hence aiding proactive decision-making and advancing the overarching goal of agile project success. The following portions of this article detail the theoretical underpinnings, methodological implementation, case studies, and evaluation metrics to illustrate the efficacy and scalability of the proposed framework.

II. Related Works

A unique Fuzzy-AHP-based strategy was offered as part of this study [1] with the purpose of systematically discovering, validating, and rating the hidden cost-overhead elements that are related to agile software development. This investigation was carried out with the intention of achieving the aforementioned aim. A method that used a two-stage empirical approach was utilized in this procedure. The first step was conducting a comprehensive systematic literature review (SLR) in order to collect information on potential cost-overheads. For the aim of validation, the second step consisted of a substantial survey that was administered to 154 different freelance agile practitioners. After that, the cost drivers were sorted into their respective categories in line with the commonly used 4P taxonomy, which is composed of People, Process, Project, and Product. After that, the cost drivers were categorized. "Changing/Unclear Requirements," "Client/User Communication," and "Time Zone challenges" were some of the cost variables that were rated when fuzzy-AHP was used to control the inherent uncertainty that was present in pairwise comparisons. These were some of the criteria that were evaluated. Ratings were assigned to each of these factors. In particular, this venture is one of the first in the last five years to completely combine Agile estimating with validated cost identification, theme structuring, and prioritizing while using fuzzy-AHP. This is a significant accomplishment in the field. One might say that this is a noteworthy achievement.

This study [12] emphasizes a number of key contributions, including the use of fuzzy-AHP to express uncertainty in the context of researchgate.net, the empirical validation from agile project managers, and the categorization of cost-overheads by applying the 4Ps. These contributions are highlighted in this paper. This study attracted attention in subsequent research that was based on Agile cost modeling. In addition to reinforcing its premise as a foundation for future frameworks, this study became a foundation for future frameworks.

This study [13] offers evidence that lends credibility to the pattern that has been seen. In addition to highlighting the value of stakeholder engagement, scalability, and validation, the objective of this research was to describe the significance of fuzzy approaches in real-world Agile environments. In order to achieve these objectives, the research project carried out a survey of 32 AI prioritizing approaches that were pertinent to Agile. Fuzzy-AHP was one of the strategies that was being reviewed as part of this study. Subsequent to the establishment of a robust foundation for Fuzzy-AHP in Agile cost prioritization, more research was carried out to broaden the scope of this methodology by including complementary MCDM techniques.

This study [14] was carried out with the purpose of discovering whether or not it is feasible to enhance the robustness of prioritizing by integrating strategies such as fuzzy-AHP, machine learning, and maybe TOPSIS, amongst other methodologies. In light of this discovery, it is clear that academics are doing all in their power to integrate learning algorithms with AHP models to get a more comprehensive understanding. In light of the fact that this new finding has taken place, this is shown.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

A fuzzy hybrid model that combines TOPSIS and Fuzzy-AHP was provided as part of this study [15] for parameter testing. This model was presented when the investigation was being conducted. In order to choose significant testing aspects that are associated with software development, this model was built with the purpose of picking such factors. The work presented here [16] demonstrates how Fuzzy-AHP may be integrated with ranking algorithms to handle domain-specific criteria and uncertainty, paving the way for cost prioritizing systems that are more comprehensive. Despite the fact that it does not expressly address cost overhead, this research does illustrate how Fuzzy-AHP may be implemented into the process.

Additionally, more comprehensive implementations of intuitionistic fuzzy-MCDM ways to identify Agile outsourcing providers during the COVID-19 epidemic illustrate the flexibility and adaptability of fuzzy methods during the course of changing projects throughout the life of the illness. This was also demonstrated by the fact that these implementations were carried out throughout the epidemic [17]. In spite of the fact that they are not cost-specific, they make use of comparable techniques with the intention of addressing uncertainty, causal interdependencies, and prioritization in the face of ambiguity. Even though, they are not designed to be cost-specific, it is observed to be the result.

In order to give the hybrid approach, the purpose of this study [18] was to provide a hybrid method for the categorization and prioritization of Agile software cost-overhead elements. The hybrid approach was the target of this research. Additionally, the fuzzy logic XGBoost algorithm is used in conjunction with this method. Within the context of this specific case, gradient-boosted decision trees were used to learn from the data in order to categorize and rank the components. Fuzzy logic was applied in order to solve linguistic ambiguity in the input variables, such as "poor communication" or "scope creep." This was accomplished in this study [19]. Experiments have shown that the combination of accurate data-driven modeling and fuzzy conceptualization does have the potential to aid in improving cost-overhead detection and ordering. This potential has been demonstrated via the use of experiments. The notion that fuzzy conceptualization is more accurate than classical classifiers is supported by the fact that it is more accurate.

The fact that this hybrid technique bridges the gap between automated, scalable machine learning and expert-based judgment (Fuzzy-AHP) is one of the reasons why it is so promising. There are several reasons why it is so interesting, and this is one of them. As a result of this, it enhances the system's capacity to be interpreted as well as its ability to make accurate predictions. When it comes to constructing fuzzy-AHP weightings based on feedback-driven performance, reinforcement learning, and, more specifically, Q-learning, is a suitable match. Q-learning is especially effective in this regard. This is due to the fact that Q-learning is designed to facilitate learning. This is particularly true for those who are looking for a framework that is very dynamic and can be adapted to meet the specific requirements of their organization. In the process of contrasting Fuzzy-SARSA and Fuzzy-Q-learning for cloud auto-scaling controllers, it was discovered that Fuzzy-Q-learning has the capacity to adjust fuzzy rule weights in an effective manner. This, in turn, leads to a decrease in cloud expenditures while maintaining integrity. Throughout the course of the comparison, this was established. The use of operational data-learning from fuzzy-R L hybrids has been shown to have the potential to be of aid in the optimization of fuzzy parameters. This has been demonstrated via recent research.

Such controllers are the topic of this research project, which intends to construct fuzzy Q-learning controllers for aircraft auto-landing under uncertain situations. The goal of this work [20] is to design such controllers. As a result of their ability to exhibit resilience as well as the capability to handle noise and disturbances, the fuzzy-based controller that they built was able to gain an advantage over the baseline procedures. It has been shown via research that fuzzy-Q-learning can optimize rule-based systems in situations that include high stakes and uncertainty. Even if the setting of the research does not pertain to cost management, this is the conclusion that can be drawn from it.

This study [21] was conducted with the intention of presenting a fuzzy-Q-learning approach that might be used in vehicle networks. In order to discover a solution that is satisfactory in terms of both the cost of a 4G connection and the latency rate, the objective of this work was to collect information about an optimum offloading policy from arxiv.org. Not only were they able to meet their latency requirements, but they were also able to realize cost reductions of between thirty and forty percent. Within the context of this particular case, the need to combine the capability of fuzzy logic to deal with imprecise inputs with the policy optimization of Q-learning is brought to light. The significance of this cannot be overstated in circumstances that include a delicate financial condition.

It is the purpose of this work [22] to provide "Q-EarlySettled-LowCost," a regret-optimal Q-learning strategy that decreases model regret, burn-in cost, and policy switching for single-agent applications. This approach is intended to be provided to fulfill the aim of this study. It is possible that the incorporation of cost-aware Q-learning optimizers into agile systems will be simplified as a result of the approach's major focus on cost efficiency and adaptive learning. This technique does not make use of fuzzy improvement.

In this article [23], the Fuzzy-AHP framework is presented. This framework is an embodiment of the current state of the art in verified cost driver identification and theme structuring. This article provides the structure that is required. Utilizing the 4P taxonomy as the basis for this strategy is the primary objective. As a consequence of this, it provides a solution to the problem of obtaining exact foundations for cost estimation within the setting of fuzzy uncertainty.

Furthermore, it has been discovered that the use of fuzzy-AHP in conjunction with machine learning (XGBoost) results in an increase in the accuracy of classification [24]. Concurrent MCDM hybrids are a manifestation of the trend that is occurring in the field, which is heading in the direction of multi-

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

dimensional prioritization. It is feasible to find ways in which cost weights can be constantly updated in response to input and changes in the environment when cost-aware Q-learning algorithms, such as "Q-EarlySettled-LowCost," are paired with fuzzy Q-learning implementations across domains. This is because it is possible to discover techniques where the cost weights can be adjusted continuously. Other names for this kind of learning include adaptive optimization and similar terms. In addition, learning is accompanied by reinforcement strategies.

III. RESEARCH METHODOLOGY

A. *Architecture of proposed method*

The proposed pipeline process of classification framework for prioritization of agile cost overhead is given in Fig 1.

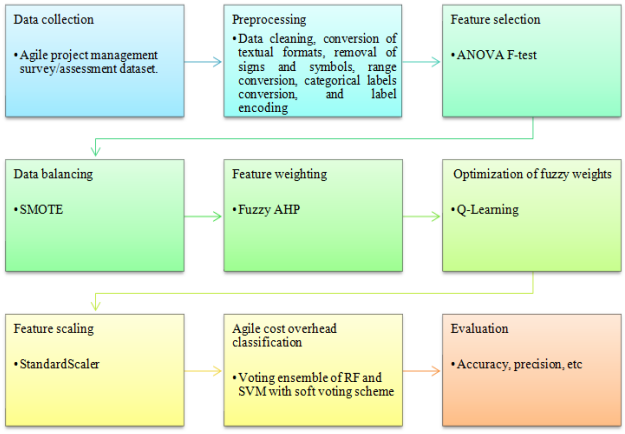


Fig. 1. Proposed architectural pipeline of the classification framework for prioritization of agile cost overhead

B. *Data Collection*

The stages of data collection and pre-processing are absolutely crucial in guaranteeing that the analytical framework that results is accurate and strong. Agile development environments are naturally dynamic, involving constant iterations, changing needs, and human-centric procedures. Building an efficient decision-making and prioritizing system, thus, it depends on obtaining high-quality data that faithfully reflects the several characteristics of cost overhead. Data collection started with the recognition of several sources in agile project environments. These included internal project management tools with structured data on sprint histories, backlog items, and velocity trends. To compile qualitative and quantitative measures of cost overhead, we also investigated communication tools, team reports, sprint retrospectives, and version control repositories. The data collection process concentrated on gathering variables either directly or indirectly, causing increasing Agile project expenses. These data were gathered by means of structured interviews, questionnaires, and retrospectives, subsequently transformed into standardized scales—usually in the form of linguistic variables—low, medium, high.

The data is sourced from agile project management survey/assessment dataset. The dataset has a number of samples of 83 and a number of features of 58 (after removing 9 metadata/unwanted attributes). The target variable has four classes such as People, Process, Product, and Project.

Feature categories and agile cost mapping: the experience & project metrics have 4 features, which represent direct or indirect agile cost drivers. Here, experience (in years): human resource cost factor, project revenue type: business/financial dimension, current spillover %: schedule delays and rework cost, and current CSAT rating: customer-impact and service quality cost. Secondly, the pairwise comparison features have 50 features, which is in the format of "a over b", representing perceived importance between agile dimensions. They are mapped to four agile cost categories: category no. Of features represents people, with 18 features for team skills, communication, and capacity-related costs. Process has 17 for Methodology overhead, planning, and coordination cost. The Product has 14 features for Technical complexity, quality, and tooling costs. The Project has 10 features for Schedule risk, compliance, and escalation cost. Finally, the Impact Metrics has 4 features, which are productivity / sprint velocity impact, CSAT/NPS impact → customer retention cost, compliance impact → regulatory/legal costs, and quality/defect impact (when applicable).

C. *Data pre-processing*

After acquiring data, the preprocessing is crucial for preparing the data. Initially, data is cleaned by removing 9 metadata fields (e.g., Timestamp, Email, identifiers) to avoid noise and irrelevant signals. Data normalizing and transformation came next, after cleaning. Especially for algorithms sensitive to scale, like Fuzzy AHP and Q-Learning, variables with different units and scales need to be standardized to enable fair comparison and accurate computation. Quantitative data was standardized using Z-score standardizing and Min-Max scaling, among other normalizing methods. Linguistic terms were translated into fuzzy numbers for qualitative data, especially those derived from survey responses or subjective assessments.

Missing value imputation for numerical data is derived in Eq. (1).

$$x_i = \frac{1}{n}\sum_{j=1}^{n} \quad x_j \tag{1}$$

Normalization factor (x') is given below in Eq. (2).

$$x' = \frac{x - min(x)}{(x) - min(x)} \tag{2}$$

Here, X denotes the feature vector, and x is the original value, x_i denotes a missing value and x_j indicates observed values.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

Z-score is also normalization process, devised in Eq. (3), which is denoted by$z$, μ denotes the mean, and σ indicates standard deviation for the feature.

$$Z = \frac{x-\mu}{\sigma} \quad (3)$$

Next, feature engineering approaches such as conversion of textual formats (e.g., “6 months” to numeric 0.5), removal of signs and symbols like "+" and value standardization, conversion of ranges into their midpoint (“10–20%” to their midpoint (15%)), conversation of categorical labels into numeric ordinal scores (for CSAT rating) and label encoding to convert categories to machine-readable form (especially for project type).

Then, target creation is carried out by using weighted scoring across People, Process, Product, and Project categories, and an additional 1.2 multiplier is applied for the Product class (domain-driven adjustment). Each sample is assigned to the highest-scoring category. Finally, the label encoding is applied and acquired classes such as 0 = People, 1 = Process, 2 = Product, 3 = Project.

Ultimately, the exhaustive and methodical approach to pre-processing and data collecting guaranteed that the input to the improved classification system was both high-quality and context-aware. The foundation was essentially set for exact prioritizing of cost overhead using Fuzzy AHP and reinforcement-driven optimization through Q-Learning by converting raw agile project data into structured, fuzzy-compatible forms.

D. Feature Selection

The approach consists of selecting features from a dataset, which are the most important components to improve the performance of the classification model. It begins with utilizing a statistical method to assess every attribute based on its degree of correlation to the desired category. The top 20 most relevant characteristics are selected by ANOVA F-test.

The final representation of the number of samples is given in Fig 2.

final_dataset

	Experience (in Years)	Current Spillover (Carry forward stories) %	People over Project	Project over Process	Developer's experience over Team Size	Team Prior experience over Personal issues	Client/User Communication over Personal Issues	Technical Complexity over Changing/Unclear Requirements	Technical Complexity over Poor planning	Quality over Changing/Unclear Requirements	...	Task size over Process Maturity	Task size over Scope	Time Zon ove Manageria Skill
0	12.0	2.5	1.0	4.0	1.0	5.0	5.0	5.0	4.0	5.0	...	1.0	2.0	1.
1	7.0	2.5	1.0	5.0	3.0	5.0	5.0	5.0	3.0	5.0	...	1.0	4.0	5.
2	20.0	7.5	1.0	5.0	1.0	1.0	2.0	1.0	1.0	1.0	...	3.0	5.0	1.
3	21.0	12.5	2.0	5.0	1.0	2.0	1.0	3.0	3.0	1.0	...	2.0	2.0	3.
4	19.0	12.5	1.0	1.0	4.0	2.0	3.0	2.0	2.0	3.0	...	2.0	2.0	4.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
78	5.5	7.5	1.0	2.0	4.0	5.0	5.0	3.0	1.0	5.0	...	2.0	5.0	1.
79	12.0	7.5	3.0	1.0	3.0	5.0	5.0	3.0	1.0	5.0	...	2.0	5.0	1.
80	6.5	7.5	1.0	2.0	3.0	5.0	5.0	3.0	1.0	5.0	...	2.0	5.0	1.
81	4.5	7.5	1.0	2.0	3.0	5.0	5.0	2.0	1.0	4.0	...	2.0	5.0	1.
82	0.5	7.5	1.0	2.0	3.0	5.0	5.0	2.0	2.0	5.0	...	2.0	5.0	1.

83 rows × 21 columns

Fig. 2. Final representation of the collected dataset for prioritization of agile cost overhead

Once these salient features are discovered from the original dataset, which simplifies the process and makes it easier, one should focus on the most crucial facts. This explains a more correct and effective paradigm. The selected characteristics are then put in a clean, new dataset with suitable labeling. The selected attributes have consistent and unambiguous names; the column indicating the result or class to be predicted is labeled exactly as "Target_Category." Now ready for use in training and model assessment, this last dataset consists simply of the most significant components coupled with the goal.

E. Managing Class Imbalance with SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) is used to help reduce problems resulting from unequal class distributions. Using their nearest neighbors, SMOTE creates synthetic examples for minority classes, thereby augmenting underrepresented classes and balancing the dataset. Using a k_neighbors value of 2 in this implementation guarantees that every synthetic sample produced closely resembles real samples in the minority class.

Following SMOTE, the target set Y consists of an equal number of samples-51 for each class: 0, 1, 2, and 3; the feature set X comprises 204 samples with 20 features. Synthetic sample analysis (x_{new}x) for SMOTE is given in Eq. (4)

$$x_{new} = x_{minority} + \lambda(x_{nearest} - x_{minority} \quad (4)$$

Term λ is a random number between 0 and 1, $x_{nearest}$xis the closest neighbour of$x_{minority}$. The SMOTE is used to address a heavy imbalance in minority classes. After applying SMOTE, it obtains a balanced 4-class dataset. During classification, this harmony greatly enhances fairness and performance across all levels.

F. Fuzzy AHP-based Weight Assignment

This is the new application of Fuzzy AHP to classify agile cost overhead, along with the integration of Q-learning to refine agile feature weighting dynamically. Here, Fuzzy AHP is applied to incorporate domain knowledge into feature weights. Fuzzy AHP incorporated fuzzy logic to handle uncertainty and vagueness in decision-making by applying triangular fuzzy numbers in pairwise comparisons to better model human judgment. It is beneficial to prioritize features or criteria when clear preferences are difficult to express precisely.

From the feature selection stage, 20 features are acquired and represented as F1 to F20, and a triangular fuzzy number is assigned for each pairwise comparison. Then, it formulates the matrix with diagonal values representing equal importance, whereas upper and lower triangle values represent subjective importance. This fuzzy comparison matrix is further processed by converting a Fuzzy Triangular matrix with triangular fuzzy numbers and their reciprocal values. Next, this is normalized and results in a final pairwise comparison matrix. It ensures that each column’s fuzzy values are scaled relative to the total in that column, preparing it for weight derivation. Finally, feature weights are computed by Fuzzy AHP, where a normalized fuzzy matrix is processed row-wise to compute the average importance of each feature, producing a priority ranking. These acquired weights are useful to select or scale significant

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

features for ML models, along with the reduction of noise and maximization of efficiency and accuracy.

G. Optimization using Q-Learning

Q-learning is used to adjust fuzzy weights dynamically based on feedback, specifically in a reinforcement learning framework. The weights are refined using Q-learning, enabling adaptive optimization based on model feedback. This procedure explains how reinforcement learning, more especially, helps to maximize fuzzy weights. By means of feedback, one hopes to progressively enhance decision-making. The system determines a reward at every stage by matching the present weights with a target pattern. This process enables one to assess the quality of the present behavior. Then, depending on experience kept in the Q-table, the system chooses what to do next, either investigating fresh prospects or applying the most well-known action.

The weights change once an action is chosen. The degree of change or step size = may vary, which would allow more flexible learning. The system changes the Q-values as it learns to represent both the expected future rewards and the instantaneous reward.

The system begins to search less over time and concentrates more on applying what it has discovered to make wise decisions. It also notes weight fluctuations and rewards to track development. The fuzzy weights are eventually refined and optimized depending on the best actions discovered by experience.

- This approach dynamically modulates fuzzy weights for optimization using Q-learning within a reinforcement learning framework. The method learns the best way to change these weights over time using environmental information, hence improving the accuracy of decision-making.

- Since they serve as the optimization objectives, twenty fuzzy weights provide the basic foundation of the process. These weights influence the behavior or performance of the fuzzy logic system and will be adjusted constantly based on learning outcomes.

- Originally empty of any meaning, a Q-table guides the learning. This database keeps Q-values, the expected future reward for each combination of a state and an action. As one develops in knowledge, these ideas evolve to illustrate which actions would be most beneficial in certain situations.

- The system might decide among three possible actions: raising, reducing, or precisely preserving the weights. These activities help with fine-grained weight control during learning. Many crucial features of the Q-learning approach characterize its behavior:

- Through more than 2500 training sessions, the agent interacts with the environment, receives feedback, and develops its awareness of which behaviors provide the best outcomes. Regular improvements and education help to optimize the fuzzy weights for general performance.

- The reward function defines how the agent evaluates its actions. It provides feedback that guides learning.
    - Positive reward → action improves performance
    - Zero reward → no significant change
    - Negative reward → action decreases performance

- The reward structure should reflect the optimization goal of the environment. The state space represents the information available to the agent at each step.

- Q-learning is considered to be converged when Q-values stabilize representing the changes between updates become very small. Policy becomes stable when the agent consistently chooses the same best action. Rewards plateau denotes no significant improvement in episode rewards. Then, the exploration decreases while ε moves toward ε_min, shifting learning to exploitation.

- The Q-learning optimization process produces these weights of importance. Particularly, the formula

- $x_{weighted} = x * optimal\_weights$ (6)

H. Feature Scaling

StandardScaler function is used here to normalize the features. By removing their mean and scaling them to unit variance, this method standardizes every feature and guarantees that each one contributes equally during model learning. To determine the scaling parameters, the fit_transform function is run on the training set; the transform is then applied to the test set to guarantee constant transformation. It is used to scale every feature depending on its learnt relevance. This guarantees that during model training, the features that more actively contribute to the predictive outcome have more impact. Consequently, the weighted dataset produced reflects the learning insights gained by reinforcement learning, thereby improving model accuracy. This method improves model stability and reduces the effect of scale variances among features.

I. Classification using an Ensemble Model

The framework for classification starts by designing a Voting Classifier by combining the predictions from the SVM and Random Forest RF classifiers. This ensemble method averages expected class probabilities and chooses the class with the highest average probability as the last output by soft voting. By using the advantages of several models, this approach usually improves prediction dependability. Apart from single classifiers, this hybrid approach gives precise outcomes, exhibiting the superiority over them. It is evaluated and tested in the results section.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental setup

The parameters considered for the experimentation is given in Table I.

TABLE I. SIMULATION PARAMETERS CONISDERED FOR THE EXPERIMENTATION OF THE PROPOSED MODEL

Parameters	Values considered
Simulation platform	Python 3
Software requirements	Anaconda (Jupyter Notebook)
Random seed	Value is assigned as 42 for Q-learning exploration, numpy random operations and SMOTE
Hyperparameters allocation for Q-learning	• alpha = 0.5 # learning rate • gamma = 0.8 # discount factor • epsilon = 0.3 # exploration rate • epsilon_min = 0.01 # minimum exploration rate • epsilon_decay = 0.995 # rate at which epsilon decreases • n_episodes = 2500 # number of training episodes • n_actions = 3 # possible actions: decrease, maintain, increase weights
Hyperparameters allocation for RF	• n_estimators=100, • max_depth=None, • min_samples_split=2, • min_samples_leaf=1, • max_features='sqrt', • bootstrap=True, • random_state=42
Hyperparameters allocation for SVM	• kernel='rbf', • gamma='scale', • degree=3, • probability=False, • random_state=42

B. Train-Test Split

The collected dataset is split between 80-20 following oversampling into training and testing subsets. This guarantees that the model is trained using eighty percent of the data, 163 samples while the remaining twenty percent, 41 samples, are set aside for testing and evaluation. This deliberate division guarantees strong training and helps to evaluate models objectively on unprocessed data.

Model evaluation

The existing classification models are applied to assess their performance on the processed dataset. Every model is trained from the training set and tested using thorough criteria using the test set. The classification models such as hybrid_model (proposed method), Logistic Regression (LR), Decision Tree (DR), and K-Nearest Neighbors Classifier (KNN) are considered for analysis.

C. Performance Measures considered for analysis

The measures like accuracy, precision, recall, and F1-score offer a comprehensive picture of every model's predictive performance.

$$Recall = \frac{TP}{FN+TP} \quad (7)$$

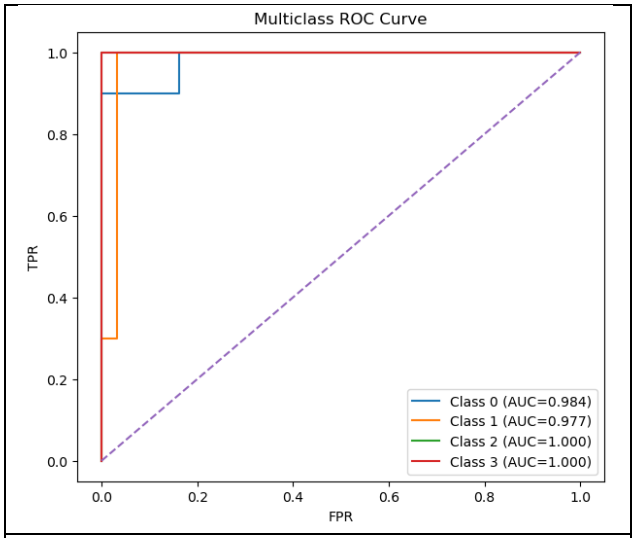
$$Precision = \frac{TP}{FP+TP} \quad (8)$$

$$F1-score = 2\frac{Precision*Recall}{Precision+Recall} \quad (9)$$

Here, TP = True Positives, FP = False Positives.

D. Performance Analysis

Fig 3 demonstrates the performance analysis on the proposed prioritization of agile cost overhead based on fuzzy AHP with Q-learning optimization model. From the analysis, it clearly, shows that the hybrid model beats the other three and boasts in terms of the best accuracy among all the tested models. This better performance shows how well several classifiers taken together create a predictive framework.



(a)

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

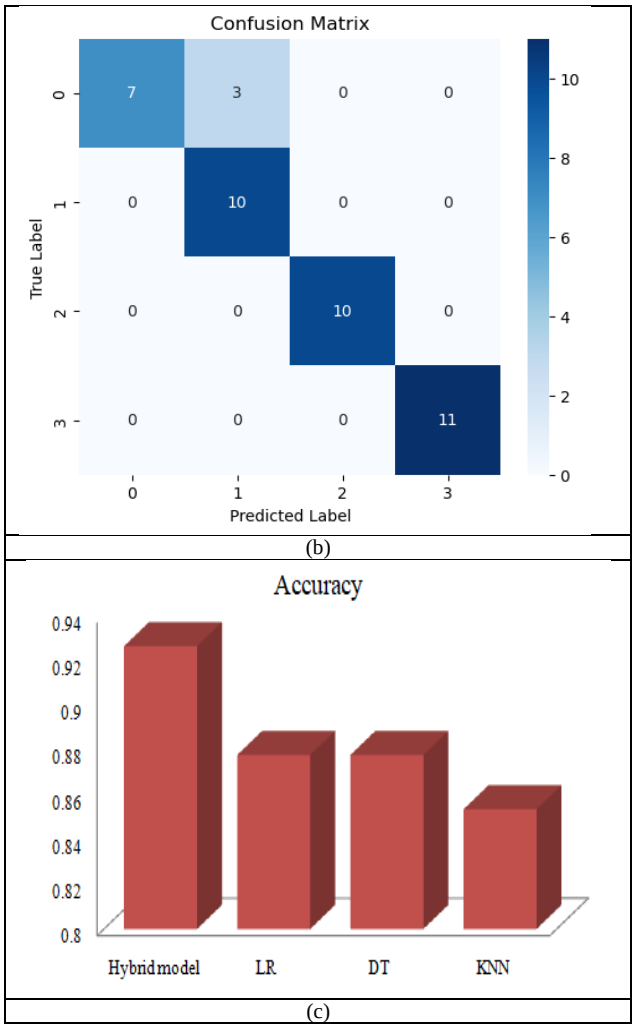


Fig. 3. Performance analysis on the proposed Prioritization of Agile Cost Overhead based on Fuzzy AHP with Q-Learning Optimization with voting classifier in terms of (a) ROC curve, (b) confusion matrix, and (c) accuracy analysis

The hybrid model probably makes use of the strengths of its base learners, via soft voting, whereby the average of predicted probabilities forms the basis of predictions. Among the four methods, KNN classifier observes the lowest accuracy, although the difference is not very great. Its reliance on distance measurements or sensitivity to noise could be the cause of this, which would not be ideal for the feature space even after scaling. The chart shows generally that the hybrid model produces the most accurate predictions, confirming the worth of ensemble learning methods. It also shows the effects of earlier preprocessing techniques, including standard scaling, class balancing via SMOTE, and feature weighting through Q-learning, which, taken together, improve the generalizing capacity of the model.

E. Overall comparative analysis

The Table II and Table III represent the overall performance evaluation on the proposed model by estimating standard performance measures.

TABLE II. OVERALL PERFORMANCE ANALYSIS OF THE PROPOSED ENSEMBLE-BASED PRIORITIZATION OF AGILE COST OVERHEAD

Metric	Mean	Std Dev	95% Confidence Interval
Accuracy	0.926829	0.0420	0.8280 – 0.9016
Precision	0.943715	0.0491	0.8116 – 0.8977
Recall	0.926829	0.0530	0.7996 – 0.8925
F1-score	0.923274	0.0517	0.8004 – 0.8910

TABLE III. OVERALL COMPARATIVE ANALYSIS OF THE PROPOSED ENSEMBLE-BASED PRIORITIZATION OF AGILE COST OVERHEAD BY ESTIMATING WITH TRADITIONAL CLASSIFIERS

Model	Accuracy	Precision	Recall	F1 Score
Hybrid model	0.926829	0.943715	0.926829	0.923274
LR	0.878049	0.89011	0.878049	0.872123
DT	0.878049	0.881991	0.878049	0.876003
KNN	0.853659	0.850333	0.853659	0.845238

The hybrid model demonstrates the highest accuracy from the plot, with a value significantly exceeding 0.92. The hybrid approach stands out as the most effective model for reducing false positives and accurately identifying relevant events. This approach provides a performance edge by potentially combining classifiers in a way that enhances their strengths while mitigating their weaknesses. Although two DT and LR models demonstrate commendable performance, their decreasing accuracy indicates constraints in their ability to generalize effectively across all classes. The KNN exhibits the lowest precision, recorded at a value of 0.85. The reliance on the configuration of the feature space, significant fluctuations in handling minority class distributions, or vulnerability to noise could all contribute to elucidating this decline. Despite the implementation of data scaling and SMOTE balancing, the presence of overlapping class borders or inadequate local patterns may continue to impede its performance.

Overall, the proposed hybrid approach has exhibited higher performance by using precise processing techniques, which has obviously improved positive instance detection. Thereby, this approach clearly demonstrated the superiority of the agile cost overhead prioritization based on Fuzzy AHP with Q-Learning optimization with voting classifier ensuring precise handling of agile applications.

V. CONCLUSION AND FUTURE SCOPE

This research has incorporated Fuzzy AHP and Q-Learning optimization into an upgraded classification system. It has been developed offering a robust framework for the prioritization of agile cost-overhead elements. This methodology has integrated several processing methods including Preprocessing by Data cleaning, conversion of textual formats, removal of signs and symbols, range conversion, categorical labels conversion, and label encoding. Further, the feature selection is conducted by ANOVA F-test whereas data balancing is carried out by SMOTE. Next, the feature weighting is conducted by Fuzzy AHP, where the fuzzy weights are tuned by Q-learning technique. Then, the features are scaled by StandardScaler

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 1173-1179

method. Finally, the Voting ensemble is proposed by integrating RF and SVM with soft voting schemes for Agile cost overhead classification. Finally, the evaluation is carried out by standard performance measures. Through the performance analysis, the hybrid approach acquires 92% of accuracy in precise classification of agile cost overhead. Thereby, it allows practitioners to develop educated cost predictions by carefully verifying and categorizing aspects within the 4Ps structure (People, Process, Product, and Project categories), and then utilizing fuzzy multi-criteria analysis to properly weight the relative significance of these pieces. To conclude, this hybrid method not only improves the accuracy of estimates but also assists agile teams in actively controlling cost overhead, which in turn promotes efficient resource allocation and improved outcomes.

However, the proposed RF–SVM soft-voting ensemble is limited by its reliance on dataset-specific weighting, static features, and its sensitivity to Agile data variability. Thereby, future work can integrate and test on dynamic weighting strategies, adaptive or meta-learning ensembles, and broader cross-organizational validations with large-scale datasets to enhance robustness and generalizability.

REFERENCES

[1] S. Abusaeed, S. U. Khan, and A. Mashkoor, "A fuzzy AHP-based approach for prioritization of cost overhead factors in agile software development," *Appl. Soft Comput.*, vol. 133, p. 109977, 2023.

[2] M. B. Javanbarg, C. Scawthorn, J. Kiyono, and B. Shahbodaghkhan, "Fuzzy AHP-based multicriteria decision making systems using particle swarm optimization," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 960–966, 2012.

[3] J. Paschen, U. Paschen, E. Pala, and J. Kietzmann, "Artificial Intelligence (AI) and value co-creation in B2B sales: Activities, actors and resources," *Australas. Marketing J.*, vol. 29, no. 3, pp. 243–251, 2020.

[4] M. Huss, D. R. Herber, and J. M. Borky, "Comparing measured agile software development metrics using an agile model-based software engineering approach versus Scrum only," *Software*, vol. 2, no. 3, pp. 310–331, 2023.

[5] J. Y. Lee, H. Burton, and D. Lallemant, "Adaptive decision framework for civil infrastructure exposed to evolving risks," *Procedia Eng.*, vol. 212, pp. 435–442, 2018.

[6] L. Liu and S. R. Ranjithan, "An adaptive optimization technique for dynamic environments," *Eng. Appl. Artif. Intell.*, vol. 23, no. 5, pp. 772–779, 2010.

[7] Y. Liu, C. M. Eckert, and C. Earl, "A review of Fuzzy AHP methods for decision-making with subjective judgements," *Expert Syst. Appl.*, vol. 161, p. 113738, 2020.

[8] H. Shi, Z. Lin, S. Zhang, X. Li, and K.-S. Hwang, "An adaptive decision-making method with Fuzzy Bayesian reinforcement learning for robot soccer," *Inf. Sci.*, vols. 436–437, pp. 268–281, 2018.

[9] D. Wiesmann, "Avoidance of the term agile in software engineering: Necessary and possible," *J. Softw.: Evol. Process*, 2023.

[10] J. A. Martínez-López, F. García, F. Ruiz, and A. Vizcaíno, "Contributions of enterprise architecture to software engineering: A systematic literature review," *J. Softw.: Evol. Process*, 2023.

[11] P. Pandey and R. Litoriya, "Fuzzy AHP-based identification model for efficient application development," J. Intell. Fuzzy Syst., vol. 38, no. 3, 2020. doi: 10.3233/JIFS-190508.

[12] M. Ali, M. Rehman, and S. Hussain, "A hybrid Fuzzy AHP–TOPSIS method for decision-making in software project management," *Appl. Soft Comput.*, vol. 123, p. 108994, 2022.

[13] A. Khan, M. Javed, and M. Saeed, "AI-driven prioritization techniques in Agile: A comprehensive survey," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 3, pp. 101–111, 2024.

[14] A. Rehman and N. Ahmed, "Fuzzy AHP–TOPSIS integration for decision support in Agile environments," *SSRN Electron. J.*, 2022.

[15] M. Borhan, R. Farooq, and S. Iqbal, "Story prioritization in Agile-Scrum using fuzzy logic techniques," *J. Syst. Softw.*, vol. 200, p. 111423, 2024.

[16] I. Gondal, U. Tariq, and M. Khan, "Fuzzy-based prioritization of requirement elicitation techniques in Agile development," *Int. J. Inf. Technol. Decis. Making*, vol. 24, no. 2, pp. 367–385, 2025.

[17] S. Duleba, S. Moslem, and Á. Török, "Aggregation techniques in intuitionistic fuzzy AHP: An analysis for public policy evaluation," *Decis. Support Syst.*, vol. 140, p. 113429, 2021.

[18] Y. Zhang, J. Li, and X. Zhao, "Enhancing lean-agile supply chain resilience using fuzzy AHP-based methods," *J. Manuf. Syst.*, vol. 67, pp. 312–324, 2023.

[19] T. Falana, M. Ogunniyi, and K. Musa, "Prioritization of privacy risks using fuzzy AHP and game theory," *Comput. Secur.*, vol. 133, p. 103155, 2024.

[20] H. Arabnejad, C. Pahl, P. Jamshidi, and A. Metzger, "Comparison of fuzzy Q-learning and fuzzy SARSA for adaptive cloud resource scaling," *J. Syst. Softw.*, vol. 134, pp. 203–216, 2017.

[21] H. Zahmatkesh, A. Habibzadeh, and M. Khezri, "A fuzzy Q-learning-based controller for automatic aircraft landing," *Aerosp. Sci. Technol.*, vol. 139, p. 107212, 2023.

[22] T. M. Nguyen, H. Q. Bui, and T. T. Vo, "Cost-saving strategies in vehicular networks using fuzzy Q-learning optimization," *Veh. Commun.*, vol. 45, p. 101031, 2024.

[23] R. Peralta, N. Singh, and A. Kumar, "Incident response prioritization using hybrid fuzzy Q-learning and NLP models," *Expert Syst. Appl.*, vol. 238, p. 121573, 2025.

[24] H. Chang, S. Wang, and Y. Liu, "Fuzzy AHP-based evaluation of agility criteria in global production networks," *Comput. Ind. Eng.*, vol. 168, p. 108028, 2022.