I.J. Information Engineering and Electronic Business, **2026, 1, 1-23**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2026.01.01

Multi-Criteria Decision-Making (MCDM) Approach for Software Architecture Selection in Cloud Computing Using Evidential Reasoning and Bayesian Inference Techniques

Jide Ebenezer Taiwo Akinsola*
Department of Computer Sciences, Abiola Ajimobi Technical University, Ibadan 200255, Nigeria
E-mail: akinsolajet@gmail.com
ORCID iD: https://orcid.org/0000-0003-2505-4024
*Corresponding Author

Akinwale Olusolabomi Akinkunmi
Department of Computer Sciences, Abiola Ajimobi Technical University, Ibadan 200255, Nigeria
E-mail: akinwaleakinkunmi2005@gmail.com
ORCID iD: https://orcid.org/0009-0005-1186-057X

Ifeoluwa Michael Olaniyi
Department of Computer Sciences, Abiola Ajimobi Technical University, Ibadan, 200255, Nigeria
E-mail: olaniyiifeoluwa12@gmail.com
ORCID iD: https://orcid.org/ 0009-0003-9278-3611

John Edet Efiong
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile–Ife, 220282, Nigeria
E-mail: jeediof@gmail.com
ORCID iD: https://orcid.org/0000-0003-4391-2475

Emmanuel Ajayi Olajubu
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile–Ife, 220282, Nigeria
E-mail: emmolajubu@oauife.edu.ng
ORCID iD: https://orcid.org/0000-0002-3244-0807

Ganiyu Adesola Aderonmu
Department of Computer Science and Engineering, Obafemi Awolowo University, Ile–Ife, 220282, Nigeria
E-mail: gaderoun@oauife.edu.ng
ORCID iD: https://orcid.org/0000-0002-7992-514X

Received: 19 October, 2025; Revised: 13 November, 2025; Accepted: 01 January, 2026; Published: 08 February, 2026

Abstract: Choosing the optimal software architecture for cloud-based systems is a critical and complex Multi-Criteria Decision Making (MCDM) problem, characterized by multiple, often conflicting, and interdependent criteria such as performance, cost, scalability, deployment speed, security, and maintainability. This research addresses this challenge by proposing and applying an integrated MCDM methodology that leverages Evidential Reasoning (ER) and Bayesian Inference (BI). The study's primary objective is to provide a robust and transparent framework for evaluating six common architecture styles: Monolithic, Microservices, Layered, Serverless, Event-Driven, and Service-Oriented Architecture (SOA). The methods employed involved a multi-stage process. First, criteria weights were derived using the Analytic Hierarchy Process (AHP) through expert pairwise comparisons. The techniques for handling uncertainty and dependencies were central. ER was utilized to aggregate subjective and objective assessments, representing them as belief distributions to explicitly account for imprecision and ignorance. Concurrently, BI was applied to model probabilistic interdependencies between criteria (Security influencing Performance, Performance influencing Scalability and Cost) within a Bayesian Network. The Intelligent Decision System (IDS) tool facilitated the operationalization of both ER aggregation and Bayesian inference. The results of the AHP weighting revealed the priorities: Performance

(0.3930), Security (0.2355), Scalability (0.1420), Maintainability (0.1160), Deployment Speed (0.0568), and Cost (0.0568). The overall evaluation, integrating these weighted criteria with ER and BI, identified Monolithic architecture as the most suitable option, achieving a utility score of 0.81. This ranking was followed by Event-Driven (0.69), SOA (0.68), Serverless (0.68), Microservices (0.65), and Layered (0.47). A comprehensive sensitivity analysis was conducted to assess the robustness of this decision. Crucially, the analysis demonstrated that while the Monolithic architecture was initially optimal, significant shifts in criteria weights could alter the ranking. Specifically, when the weight of Security was substantially increased (to ~0.32) and Performance decreased (to ~0.25), the Serverless architecture emerged as the new top-ranked alternative (83% utility score), surpassing Monolithic (78%). This finding underscores the critical influence of strategic priorities on architecture selection. Future studies may also focus on developing data-driven, adaptive, and domain-specific decision frameworks to enhance the robustness, transparency, and real-world applicability of MCDM approaches for cloud-based software architecture selection.

Index Terms: AHP, Bayesian Inference, Cloud Computing, Evidential Reasoning, Intelligent Decision System, MCDM, Software Architecture

1. Introduction

Multi-Criteria Decision Making (MCDM) is a pivotal field within operations research and management science, addressing complex decision problems characterized by multiple, often conflicting, objectives or criteria [1]. In an increasingly intricate world, decision-makers across various domains—from engineering and environmental management to finance and healthcare—are frequently confronted with scenarios where a single optimal solution is elusive due to the inherent trade-offs between desirable outcomes. The essence of MCDM lies in providing a structured framework to evaluate, compare, and rank alternatives based on their performance across these diverse criteria, thereby facilitating more informed and transparent decision-making processes [2]. The objective of MCDM is to logically determine the weights between multiple metrics [3].

Traditional MCDM approaches, while robust, often face limitations when dealing with uncertainties, imprecision, or incomplete information that are commonplace in real-world decision environments. The challenge intensifies when information is subjective, conflicting, or derived from various unreliable sources, necessitating methodologies that can effectively aggregate and process such complex data. This has led to the development of advanced MCDM techniques capable of handling these intricacies, moving beyond deterministic models to embrace probabilistic and uncertainty-modeling paradigms.

Among these advanced approaches, Evidential Reasoning (ER) and Bayesian Inference (BI) emerge as powerful tools for addressing decision problems under uncertainty, particularly when dealing with subjective assessments and varying degrees of belief [4]. Evidential Reasoning, rooted in Dempster-Shafer theory, offers a robust framework for combining evidence from multiple sources, allowing for the representation of both ignorance and conflicting information, which is critical in situations where precise probabilities are unavailable [4]. Conversely, Bayesian Inference provides a statistical method for updating the probability of a hypothesis as more evidence or information becomes available, making it highly suitable for dynamic decision environments where data accumulates over time [5]. The integration of these methodologies within MCDM frameworks holds significant promise for enhancing the reliability and robustness of decisions, particularly in complex scenarios where a comprehensive understanding of uncertainties is paramount.

Software architecture serves as a blueprint for system development [6], and the proper evaluation of existing architectures is crucial [7] in developing sophisticated systems in a cloud computing environment. A system architecture can be evaluated using formal method by integrating machine learning and MCDM [8].

This study specifically applies MCDM principles, leveraging the strengths of Evidential Reasoning and Bayesian Inference, to the critical area of selecting the best software architecture for cloud computing environments. Cloud computing, with its dynamic and distributed nature, introduces complexities where architecture choices directly impact performance, security, scalability, and cost. Furthermore, the research addresses the assessment of tools within such architectures, with a particular focus on the

Intelligent Decision System (IDS) tool [9] includes the development and demonstration of a methodology that effectively combines ER and BI to navigate these complex decision landscapes, aiming to provide a more nuanced and reliable evaluation of various software architectures and their integrated security components under conditions of imperfect knowledge.

Evidential Reasoning is designed for decision problems that involve mixed data types, uncertainty, incomplete information, and qualitative judgment. Decision problems, including various data kinds, uncertainty, inadequate knowledge, and qualitative judgment, are best suited for evidential reasoning. This indicates the significant importance of ER in operation research-based study, as in this research.

[9] focused on an application area where such uncertainties and multi-faceted criteria are prevalent. This work contributes to the practical utility of advanced MCDM techniques in generating actionable insights for decision-makers

in cloud infrastructure management. MCDM methods require mathematical formulations for insightful decision making. Also, this study contributes to the body of existing knowledge in MCDM in the following spheres:

1. Application of ER for software architecture selection
    a. ER was applied to handle uncertainty and ambiguity in the criteria with recourse to:
        - Evaluations of architecture attributes that are inaccurate, ambiguous, or partial are explicitly modeled by ER, for instance, by considering "performance is between poor and excellent" using belief degrees based on linguistic assessments.
        - Validation of linguistic scales from the research results, based on the world-wide acclaimed Saaty's method, as well as intervals, and belief levels that correspond to the natural ways in which software architects make decisions.
    b. Integration of various Qualitative and Quantitative Criteria, through:
        - Modifiability, usefulness, and maintainability of the subjective architecture attributes that are critical.
        - Qualitative criteria modeling using belief degrees based on linguistic assessments to correct the subjectivity of the opinion because software architecture criteria are subjective.
        - Logical and mathematically consistent integration of subjective expert views with objective measures.
    c. Strong Evidence Combination from the:
        - Combination of evidence from several sources and stakeholders using Dempster-Shafer theory
2. BI implementation to solve the software architecture selection
    a. Measurement of Probabilistic Uncertainty by:
        - Utilizing formal risk analysis in architecture selection by requiring uncertainty to be represented as probabilities.
    b. Integration of Prior Knowledge by utilizing:
        - Historical software project knowledge to improve decision quality.
3. Hybridization of BI and ER with AHP for weight elicitation
        - Management of risk-based evaluation in comparison with qualitative judgement by providing a robust decision approach that is objective and not subjective.

2. Related Works

[10] introduces multi-criteria decision making (MCDM) as a decision-making method for rational real-life decision making, and its approach has become of great importance for solving complicated, multi-attribute decision-making issues in unpredictable situations [11]. In a similar regard, [12] developed two algorithms for multi-attribute decision making in the case of uncertainty. [13] also applied the MCDM technique to evaluate the performance of different supervised machine learning algorithms. [14] applies the Evidential Reasoning (ER) method to select software components within an online bookstore system, showcasing its utility beyond theoretical models. The authors systematically assign belief degrees such as "70% reliable, 20% maintainable, 10% costly"—to each component, capturing both quantitative and qualitative aspects under a unified belief structure [14]. Other area of application of MCDM includes supply chain management [15], waste management [16], healthcare [17], energy management [18], and energy planning and formulation [19].

[20] offer the foundational theory and methodology for the Evidential Reasoning approach within MCDM. The study's exposition links Dempster–Shafer theory to pragmatic decision frameworks, enabling the handling of both quantitative and qualitative attributes under conditions of uncertainty. The study explains how belief structures can represent ambiguous evaluations—such as "60% good, 40% average"—and how ER aggregation rules combine these into a coherent assessment. Importantly, the research also introduces the Intelligent Decision System (IDS), providing a software implementation that operationalizes the theoretical contributions. The work highlights ER's flexibility and robustness compared to deterministic methods like AHP, thereby validating the use of ER in contexts like software architecture, where some criteria (for example, maintainability, security) cannot be strictly measured [20].

Building on the prior work by the authors [4], enhancing ER with the Interval Evidential Reasoning (IER) extension to handle interval uncertainty. Instead of precise belief values, decision-makers can define intervals—such as maintainability being between 0.6 and 0.8 confident—providing a more realistic representation of ambiguous evaluations during early architecture selection. The IER algorithm computes belief bounds and aggregates them to reflect both best- and worst-case scenarios. The approach enhances decision transparency and sensitivity, showing how broader uncertainties impact ranking outcomes. Similarly, in designing selection architecture, IER could support early-stage assessments where performance numbers or cost data are not yet fully reliable, thus adding rigor and confidence to decision-making [4].

[5] introduces a unified Bayesian framework for MCDM that comprehensively addresses preference uncertainty, criteria correlation, and group decisions. The hierarchical model supports different uncertainty types, such as normal distributions or interval preferences and accounts for correlated criteria through joint probability distributions. Notably, the model produces probabilistic weights and rankings, offering distributional insights rather than single-point estimates. The introduction of probabilistic mixture modeling allows grouping decision-makers into homogeneous clusters based on preference structures, with posterior distributions supporting richer analysis. [5] validates the framework through

numerical examples, demonstrating how Bayesian inference can enhance both methodological rigor and interpretability in MCDM tasks. The proposed framework illustrates a practical, theoretically-grounded pathway to embedding Bayesian inference into IDS and provides templates for handling weight uncertainty and criterion interdependence [5].

[2] extends AHP into ANP to accommodate interdependencies among software architecture criteria. The study models feedback loops, for example, performance impacting scalability, and vice versa and derive priorities using networked comparisons. While ANP captures the architecture decision's complexity, it remains deterministic and does not explicitly handle uncertainty. The findings suggest that performance and scalability form a positive feedback loop in systems selection. This aligns with the motivation to introduce Bayesian inference for uncertainty modeling, signaling the complementary use of probabilistic networks over traditional deterministic ANP [2]. The Saaty's method, known as AHP, was applied to select an optimal candidate for a potential business partner, as clearly indicated as the most important method for handling MCDM problems [21]. The adaptation of linguistic assessment has been provided by [22], which is being widely used in the MCDM domain.

[9] outlines the capabilities in supporting uncertain decision-making through ER and belief functions. It was noted that IDS can handle incomplete data, integrate qualitative and quantitative assessments, and provide belief-level outputs for decision alternatives. The study also emphasizes IDS's graphical interface and its capacity to model Bayesian inference through dependency links. While not peer-reviewed, this source provides a valuable high-level summary of how theoretical techniques are operationalized in a software tool. It confirms that IDS is an appropriate platform for combining AHP-based weights, belief aggregation using ER, and Bayesian networks [9].

2.1. Conceptual Model for Software Architecture Selection

The decision-making process for selecting software architecture in a cloud environment can be conceptualized as a structured evaluation problem involving multiple layers of assessment and aggregation. At its core, the problem involves identifying a set of architecture alternatives (for example, Monolithic, Microservices, Serverless, Event-Driven, Service-Oriented Architecture (SOA), and Layered) against a defined set of criteria. These criteria span both quantitative measures, such as performance (requests/sec), scalability (concurrent users), cost ($/month), and deployment speed (hours), and qualitative factors like security and maintainability.

A comprehensive conceptual model for this context integrates a multi-step framework:

1. Problem Definition and Alternative Identification: Clearly stating the goal of selecting the optimal cloud-based software architecture and enumerating the candidate architectures.
2. Criteria Identification and Classification: Determining the relevant evaluation criteria, distinguishing between quantitative and qualitative attributes.
3. Attribute Scale and Grade Definition: Establishing units, best/worst values for quantitative criteria, and linguistic grading scales (for example, Poor, Average, Good) for qualitative criteria.
4. Weight Assignment: Using methods like the Analytic Hierarchy Process (AHP) to derive criteria weights through expert pairwise comparisons, ensuring consistency. The prioritization of criteria such as performance, security, and scalability is paramount in cloud contexts.
5. Data Encoding and Belief Structure Definition: Transforming raw data (quantitative values or qualitative assessments) into belief distributions, which capture uncertainty and skepticism.
6. Evidential Reasoning Aggregation: Applying ER to combine belief degrees from individual attributes into an overall performance score for each architecture, accounting for attribute weights and uncertainty.
7. Bayesian Inference for Dependency Modeling: Incorporating Bayesian Inference to model interdependencies between criteria (for example, performance affecting scalability), defining conditional probability tables to reflect realistic decision contexts.
8. Ranking and Sensitivity Analysis: Ranking alternatives based on their aggregated scores and performing sensitivity analysis to understand how changes in weights or belief degrees impact the final ranking.

This integrated model allows for a nuanced evaluation that moves beyond simple deterministic aggregation, providing a robust framework for complex architecture decisions in cloud computing.

2.2. Comparison of MCDM Methods in Software Architectures Selection

The selection of software architectures, especially for cloud environments, is inherently an MCDM problem due to the multitude of conflicting technical and non-technical factors. Various MCDM methods have been employed, each with its strengths and limitations.

Traditional methods like the Analytic Hierarchy Process (AHP) are widely used for their ability to derive weights for criteria based on pairwise comparisons, providing a structured approach to incorporate expert judgment. However, AHP is fundamentally deterministic and struggles to explicitly handle uncertainty, imprecision, or the varying degrees of belief in expert assessments. Similarly, its extension, the Analytic Network Process (ANP), accommodates interdependencies among criteria by modeling feedback loops, which is highly relevant in complex systems where performance might impact scalability and vice versa. While ANP captures this complexity better than AHP, it also remains a deterministic method, lacking explicit mechanisms for uncertainty modeling. [2] highlights ANP's utility in modeling these interdependencies but implicitly underscores the need for complementary probabilistic methods when uncertainty is present.

In contrast, Evidential Reasoning (ER) offers a more sophisticated approach to dealing with uncertainty by allowing decision-makers to assign belief degrees to various assessment grades, rather than precise scores. This enables the representation of ambiguity, ignorance, and conflicting information, making it particularly suitable for qualitative criteria like security and maintainability in software architecture selection. ER's ability to aggregate beliefs from multiple sources into a coherent assessment, as demonstrated by [20], and its extension to handle interval uncertainty (IER) by [4] significantly enhances decision transparency and robustness in early-stage architecture assessments where data might be imprecise. [14] further illustrates ER's practical utility in selecting software components by systematically assigning belief degrees to capture both quantitative and qualitative aspects.

For modeling inter-dependencies and dynamic updates, Bayesian Inference (BI) provides a powerful probabilistic framework. While the AHP/ANP model static dependencies, Bayesian methods allow for updating probabilities as new evidence becomes available and for accounting for correlated criteria through joint probability distributions. [5] introduces a unified Bayesian framework for MCDM that addresses preference uncertainty, criteria correlation, and group decisions, producing probabilistic weights and rankings that offer deeper distributional insights than single-point estimates. The Intelligent Decision System (IDS) tool utilized in this study operationalizes both ER and Bayesian Inference, allowing for their combined application to handle uncertain decision-making and model Bayesian networks through dependency links. This integration provides a strong analytical framework for architecture evaluation under comprehensive uncertainty.

2.3. Evidential Reasoning (ER)

Evidential Reasoning (ER) is an MCDM approach designed to handle decision problems under uncertainty and subjective judgment, particularly effective when precise data is unavailable or when assessments involve varying degrees of belief. It is fundamentally based on the Dempster-Shafer theory of evidence (DST) [23]. ER allows for the representation of assessments as "belief structures," where a degree of belief is assigned to multiple grades or propositions, and remaining belief can be assigned to "ignorance" (not knowing).

2.3.1. Mathematical Model of Evidential Reasoning

In ER, an assessment $S(e_i)$ (represented in equation 1) for an alternative A_k with respect to attribute e_i is represented by a belief distribution over a set of discernment grades $H = \{H_1, \ldots, H_N\}$:

$$S(e_i) = \left\{\left(H_j, \beta_{j,i}\right) \mid j-1, \ldots, N\right\} \cup \left\{(H_j, \beta_{H,i})\right\} \tag{1}$$

Where:

H_j are the individual assessment grades (e.g., Poor, Average, Good, Excellent).

$\beta_{j,i}$ is the belief degree that alternative A_k belongs to grade H_j under attribute e_i.

$\beta_{H,i}$ represents the unassigned belief, or "ignorance," concerning attribute e_i.

The sum of beliefs must be satisfied as represented in equation 2:

$$\sum_{j=1}^{N} \beta_{j,i} + 6\beta_{H,i} = 1 \tag{2}$$

When combining evidence from multiple attributes ($e_1, e_2, \ldots, e_L$) for an alternative A_k, the ER approach uses an aggregation rule. For two attributes e_1 and e_2, the combined belief degree $m_k(H_j)$ for grade H_i is given by the orthogonal sum, which is a key component of the Dempster-Shafer combination rule. It is mathematically represented in equation 3.

$$m_k(H_j) = \frac{1}{1-k} \sum\nolimits_{X \cap Y = H_j} m_1(X) m_2(Y) \tag{3}$$

Where:

$m_1(X)$ and $m_2(Y)$ are the basic probability assignments (BPAs) from attributes e_1 and e_2 respectively, for subsets of grades X and Y.

K (shown in equation 4) is a normalization factor that measures the conflict between the two pieces of evidence:

$$K = \sum\nolimits_{X \cap Y = \varnothing} m_1(X) m_2(Y) \tag{4}$$

This rule can be extended iteratively to aggregate evidence from all L attributes. The IDS tool operationalizes this process, taking attribute weights into account during aggregation.

2.4. Bayesian Inference (BI)

Bayesian Inference (BI) is a statistical method for updating the probability of a hypothesis as more evidence or information becomes available. Unlike frequentist methods, Bayesian inference uses prior probabilities (representing initial beliefs) and updates them based on observed data to produce posterior probabilities (representing updated beliefs) [24]. In MCDM, BI is particularly valuable for modeling dependencies between criteria and for propagating uncertainties through a decision model.

2.4.1. Mathematical Model of Bayesian Inference (Bayes' Theorem)

The core of Bayesian Inference is Bayes' Theorem, it is mathematically represented in equation 5 as:

$$P(H \mid E) = \frac{P(E \mid H).P(H)}{P(E)} \tag{5}$$

where:

P(H | E) is the posterior probability, the probability of a hypothesis H given the observed evidence E. This is the updated belief after considering the evidence.

P(E | H) is the likelihood, the probability of observing the evidence E given that the hypothesis H is true.

P(H) is the prior probability, the initial probability of the hypothesis H before any evidence is observed.

P(E) is the marginal likelihood or evidence probability, the total probability of observing the evidence E across all possible hypotheses. It acts as a normalization constant. Equation 6 shows it mathematical representation.

$$P(E) = \sum_i P(E \mid H).P(H) \tag{6}$$

In the context of MCDM for software architecture selection, hypotheses (H) could be the suitability of a particular architecture type, and evidence (E) could be the performance metrics or qualitative assessments on specific criteria. For modeling dependencies, Bayesian networks are employed, where nodes represent criteria or attributes, and directed edges represent probabilistic dependencies between them. Each node is associated with a conditional probability table (CPT) that quantifies the strength of these relationships. For example, the probability of a certain "Scalability" level might be conditional on the "Performance" level.

The Intelligent Decision System (IDS) tool supports the definition of these dependency links and CPTs, allowing for a more realistic representation of the interdependencies between attributes that are common in complex systems like cloud architectures. This allows for the propagation of uncertainty and the calculation of updated probabilities for architecture suitability based on various criteria and their relationships. [5] further elaborates on unified Bayesian frameworks for MCDM, showcasing their ability to handle various uncertainty types and correlated criteria to generate probabilistic rankings.

3. Materials and Methods

This section delineates the systematic approach adopted for the multi-criteria decision-making problem of selecting optimal software architectures for cloud computing, explicitly integrating Evidential Reasoning (ER) and Bayesian Inference (BI). The methodology emphasizes a structured evaluation process that accounts for both objective data and subjective uncertainties inherent in complex architecture choices.

3.1. Research Design and Overall Methodology

The research design is structured as a prescriptive case study, applying an integrated MCDM framework to a specific, critical decision problem in cloud computing: the selection of software architectures and the evaluation of critical components like an Intelligent Decision System (IDS). The overall methodology follows a hybrid approach, combining the strengths of AHP for weight elicitation, Evidential Reasoning for handling uncertainty in assessments, and Bayesian Inference for modeling probabilistic dependencies among criteria. Fig. 1 shows the overall MCDM process flow.

3.2. Criteria Definition and Data Collection

For the selection of software architectures in cloud computing, a comprehensive set of criteria was identified, encompassing both technical and non-technical dimensions crucial for system performance and operational effectiveness. These criteria were defined through literature review and expert consultation to ensure relevance to contemporary cloud environments. Examples of key criteria include:

1. Performance: Measured by throughput (for example, requests per second), latency, and response time.
2. Scalability: The ability of the architecture to handle increased workload or users efficiently.
3. Security: Encompassing measures like data encryption, access control, and the effectiveness of integrated decision system tools, such as the IDS.

4. Maintainability: Ease of modifying, updating, or repairing the system.
5. Cost: Total cost of ownership, including deployment, operation, and maintenance.
6. Reliability: The probability of the system performing its intended function without failure.

Data for evaluating architecture alternatives against these criteria can be quantitative (Can be given a numerical input) or qualitative (non-numeric input can be provided). For qualitative criteria, linguistic terms (for example, Poor, Average, Good, and Excellent) were defined, and corresponding belief distributions were assigned to capture the inherent uncertainty in expert judgments.

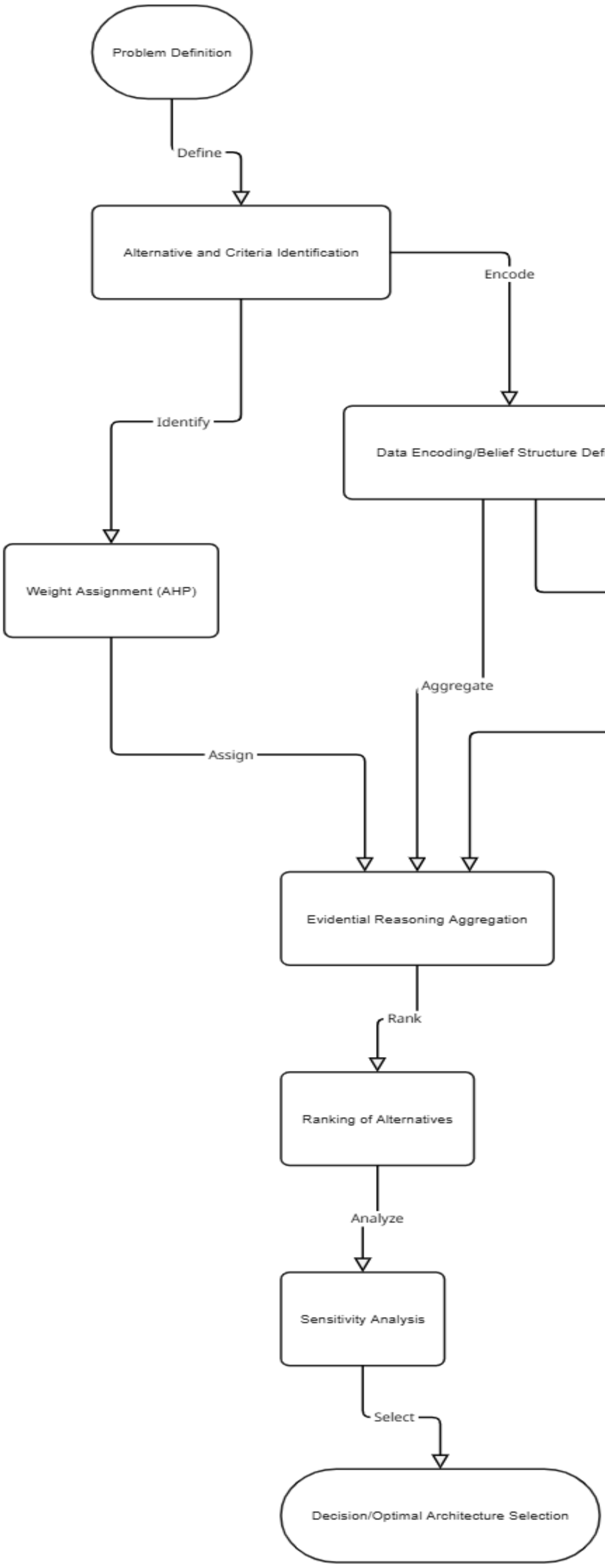


Fig. 1. Overall MCDM Process Flowchart

3.2.1. Criteria Selection and Classification

Each architecture alternative was evaluated across six criteria, categorized into quantitative and qualitative. Table 1 shows each attribute with its unit as well as the range of input values it can take. Table 1 shows the real empirical baselines of raw metrics for the software architecture alternatives. The baselines used for various attributes in software architecture selection from monolithic architecture to micro-services are highlighted accordingly from the literature; based on performance [25] under similar stress/load scenarios, consideration of requests /second as well as cost [26], scalability evaluation [27], deployment speed [28], deployment concerns in terms of security [29], in addition to maintainability paradigm as a qualitative metric [30].

Table 1. Architecture Alternatives and Attribute Inputs

Attribute	**Unit / Grade**	**Best–Worst Range**
Performance	Requests/sec	1000 – 10000
Scalability	Concurrent users	1000 – 500000
Cost	USD/month	\$500 – \$5000
Deployment Speed	Hours	1 – 72
Security	Qualitative	High risk (0) – Very Secure(1.0)
Maintainability	Qualitative	Very difficult (0) – Very easy (1.0)

The qualitative criteria were modeled using **belief degrees** based on linguistic assessments (for example, poor, average, good, excellent), and the quantitative ones were mapped within best-to-worst performance intervals.

Fig. 2 illustrates how a quantitative criterion, specifically 'Performance,' was configured within the Intelligent Decision System (IDS) tool. It depicts the attribute 'Performance' as a quantitative type, specifies its unit as 'requests processed per second,' and sets its minimum and maximum input values at 1000 and 10000, respectively. This configuration is crucial for the precise and consistent evaluation of architecture alternatives based on objective performance metrics within the MCDM framework. Fig. 3 shows the number of grades used for maintainability.

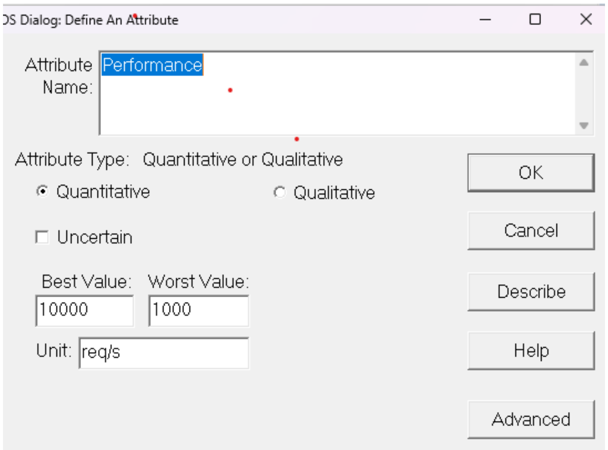


Fig. 2. Performance Attribute (Quantitative) with upper and lower bound values.

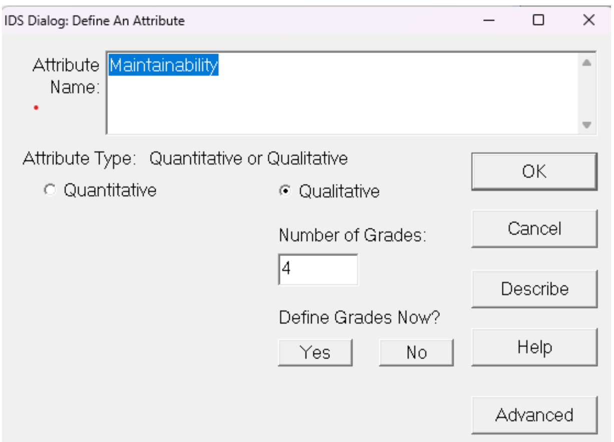


Fig. 3. Maintainability Attribute (Qualitative) with Number of Grades.

For qualitative criteria, linguistic terms (for example, Poor, Average, Good, and Excellent) were defined, and corresponding belief distributions were assigned to capture the inherent uncertainty in expert judgments. For instance. Assessment Grades for Maintainability Criterion are visually shown in Fig. 4, representing the qualitative assessment grades established for 'Maintainability'. These grades, such as 'Very difficult', 'Average', 'Easy', and 'Very easy' form the discrete set of discernment states over which expert beliefs are distributed within the Evidential Reasoning framework, enabling the precise capture of subjective evaluations.

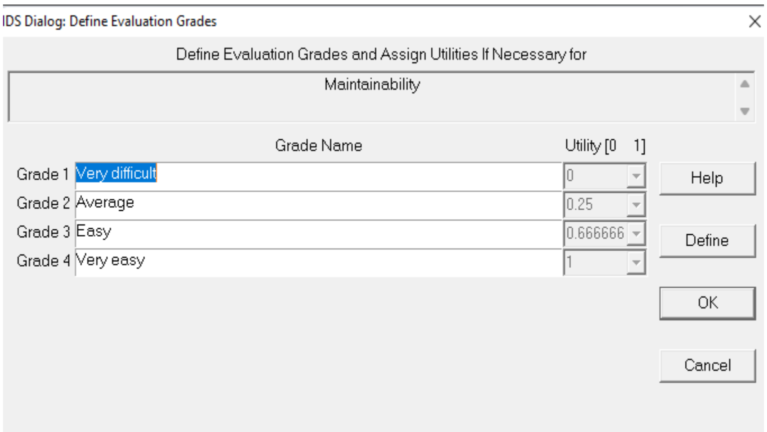


Fig. 4. Assessment Grades for Maintainability Criterion.

3.2.2. Weight Assignment Using AHP

The **Analytic Hierarchy Process** (AHP) was used to derive the weights for each criterion through pairwise comparisons. The decision matrix was constructed with expert judgments about the relative importance of each criterion they were ranked in this order of importance:

1. Performance
2. Security
3. Scalability
4. Maintainability
5. Deployment Speed (equal with cost)
6. Cost (equal with deployment speed)

The detailed mathematical steps involved are as follows:

1. Constructing the Pairwise Comparison Matrix (*A*): Experts compare each criterion against every other criterion with respect to its importance towards the overall goal, using Saaty's fundamental scale (1-9). The results are organized into an n x n matrix, where n is the number of criteria. If a_{ij} represents the relative importance of criterion i compared to criterion j, then $a_{ji} = 1/a_{ij}$, and $a_{ij} = 1$

For this study, with 6 criteria (Performance (P), Security (S), Scalability (Sc), Maintainability (M), Deployment Speed (D), and Cost (C)), the pairwise comparison matrix A was constructed based on expert judgments, and it is shown in Table 2. The matrix is also represented mathematically in equation 7.

Table 2. Pairwise Comparison Table

	P	**S**	**Sc**	**M**	**D**	**C**
P	1	2	3	4	5	5
S	1/2	1	2	3	4	4
Sc	1/3	1/2	1	2	3	3
M	1/4	1/3	1/2	1	2	2
D	1/5	1/4	1/3	1/2	1	1
C	1/5	1/4	1/3	1/2	1	1

$$A = \begin{bmatrix} 1.00 & 2.00 & 3.00 & 4.00 & 5.00 & 5.00 \\ 0.50 & 1.00 & 2.00 & 3.00 & 4.00 & 4.00 \\ 0.33 & 0.50 & 1.00 & 2.00 & 3.00 & 3.00 \\ 0.23 & 0.33 & 0.50 & 1.00 & 2.00 & 2.00 \\ 0.20 & 0.25 & 0.33 & 0.50 & 1.00 & 1.00 \\ 0.20 & 0.25 & 0.33 & 0.50 & 1.00 & 1.00 \end{bmatrix} \tag{7}$$

2. Normalizing the Pairwise Comparison Matrix: Each element in a column of matrix A is divided by the sum of that column to obtain a normalized matrix $\bar{A}$. This process transforms the absolute comparisons into proportional values within each column. This is mathematically expressed as in equation 8.

$$\bar{a}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} \tag{8}$$

The column sums for matrix A are: P=2.48, S=4.33, Sc=7.16, M=11.00, D=16.00, C=16.00. The normalized matrix $\bar{A}$ is approximately expressed in equation 9.

$$A = \begin{bmatrix} 0.403 & 0.462 & 0.419 & 0.364 & 0.313 & 0.313 \\ 0.133 & 0.231 & 0.279 & 0.273 & 0.250 & 0.250 \\ 0.133 & 0.115 & 0.140 & 0.182 & 0.188 & 0.188 \\ 0.101 & 0.076 & 0.070 & 0.091 & 0.125 & 0.125 \\ 0.081 & 0.058 & 0.046 & 0.045 & 0.063 & 0.063 \\ 0.081 & 0.058 & 0.046 & 0.045 & 0.063 & 0.063 \end{bmatrix} \tag{9}$$

3. Calculating the Priority Vector (Weights w): The weights of the criteria are obtained by averaging the elements in each row of the normalized matrix $\bar{A}$. This average forms the priority vector. $w = (w_1, w_2, ..., w_n)$. The weight can be expressed as shown in equation 10.

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \bar{a}_{ij} \tag{10}$$

The calculated weights are:

1. $w_P \approx 0.379$ (Performance)
2. $w_S \approx 0.248$ (Security)
3. $w_{Sc} \approx 0.158$ (Scalability)
4. $w_M \approx 0.098$ (Maintainability)
5. $w_D \approx 0.059$ (Deployment Speed)
6. $w_C \approx 0.059$ (Cost)

These weights sum to approximately 1.001, demonstrating their relative importance in the decision hierarchy.

4. Calculating the Consistency Index (CI) and Consistency Ratio (CR): To ensure the reliability of the derived weights, the consistency of the pairwise comparisons is assessed. First, the maximum eigenvalue $\lambda_{\max}$ is calculated by multiplying the original matrix A by the weight vector w and then averaging the ratios of the elements of the resulting vector to the corresponding weights. The maximum eigenvalue is expressed as shown in equation 11.

$$\lambda_{\max} = \frac{1}{n} \sum_{i=1}^{n} \frac{(AW)_i}{w_i} \tag{11}$$

Hence, $\lambda_{\max} \approx 6.076$.

Next, the Consistency Index (CI) is computed as represented in equation 12.

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{12}$$

$$CI = \frac{6.076-6}{6-1} = \frac{0.076}{5} = 0.0152$$

Finally, the Consistency Ratio (CR) is calculated by dividing the CI by the Random Index (RI) for a matrix of size n=6, where RI = 1.24. The Consistency Ratio is mathematically represented in equation 13.

$$CR = \frac{CI}{RI} \tag{13}$$

$$CI = \frac{0.0152}{1.24} \approx\sim 0.0123$$

The calculated CR of approximately 0.0123 is significantly less than the acceptable threshold of 0.10 [31], indicating a high level of consistency in the expert's pairwise judgments. This ensures the robustness and reliability of the derived criterion weights for subsequent stages of the MCDM process.

Fig. 5 shows how weight is calculated and assigned for each of the six attributes using pairwise comparisons and AHP method.

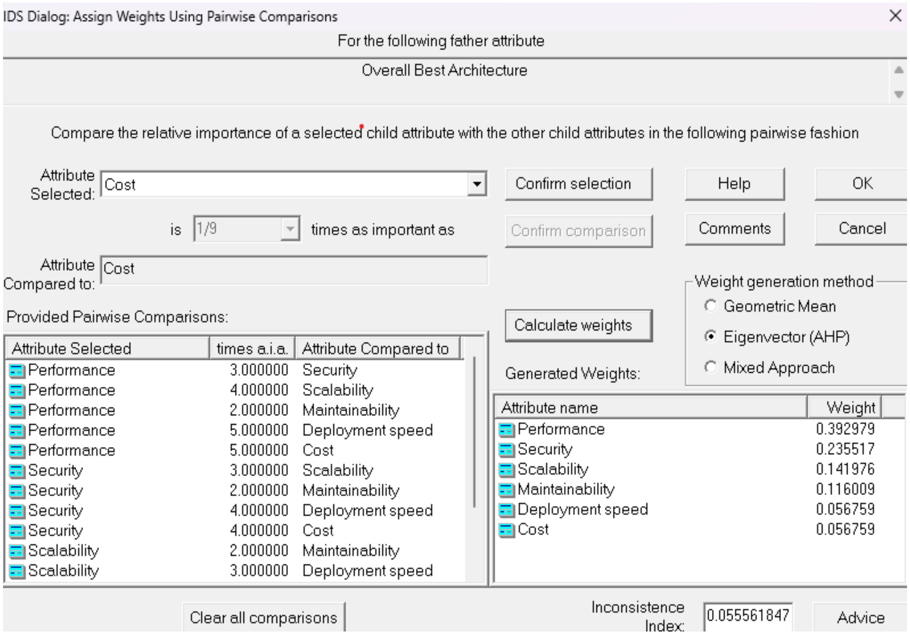


Fig. 5. Calculated Weights for Each Attribute.

3.2.3. Dependency Modeling and Probabilistic Inference (Bayesian Inference)

To account for the inherent interdependencies between criteria (for example, a performance improvement might necessitate a compromise in cost or security), Bayesian Inference (BI) was employed through the construction of Bayesian Networks. This allows for a more realistic propagation of uncertainty and the updating of beliefs based on observed or hypothesized conditions. The overall process used to achieve this can be summarized as follows:

3.2.3.1. Recap of Inputs and Dependencies

This study considers the six architecture alternatives (Monolithic, Microservices, Layered, Serverless, Event-Driven, and SOA) and the six evaluation criteria (Performance, Security, Scalability, Maintainability, Deployment Speed, and Cost) as defined in Table 1 and Table 7. The AHP weights for these criteria have also been established (for example, Performance: 0.379, Security: 0.248, etc.) in section 3.2.2.

Based on common architecture relationships and expert knowledge, the following are illustrative dependencies for our Bayesian Network:

a. **Security → Performance:** The level of security implemented can impact the system's performance (for example, encryption overhead).
b. **Performance → Scalability:** A system's core performance characteristics directly influence its ability to scale effectively.
c. **Maintainability → Deployment Speed:** Highly maintainable codebases generally lead to faster and more reliable deployments.

d. **Performance → Cost:** Achieving higher performance often comes with increased infrastructure or operational costs.

3.2.3.2. Constructing the Bayesian Network and CPTs

A Bayesian Network (BN) is constructed where nodes represent the criteria, and directed edges represent probabilistic dependencies. For this conceptual simulation, there are discrete states for each criterion (for example, Low, Medium, High for quantitative criteria, or the defined grades for qualitative ones). Fig. 6 illustrates the Bayesian network, and it indicates the dependencies among the criteria.

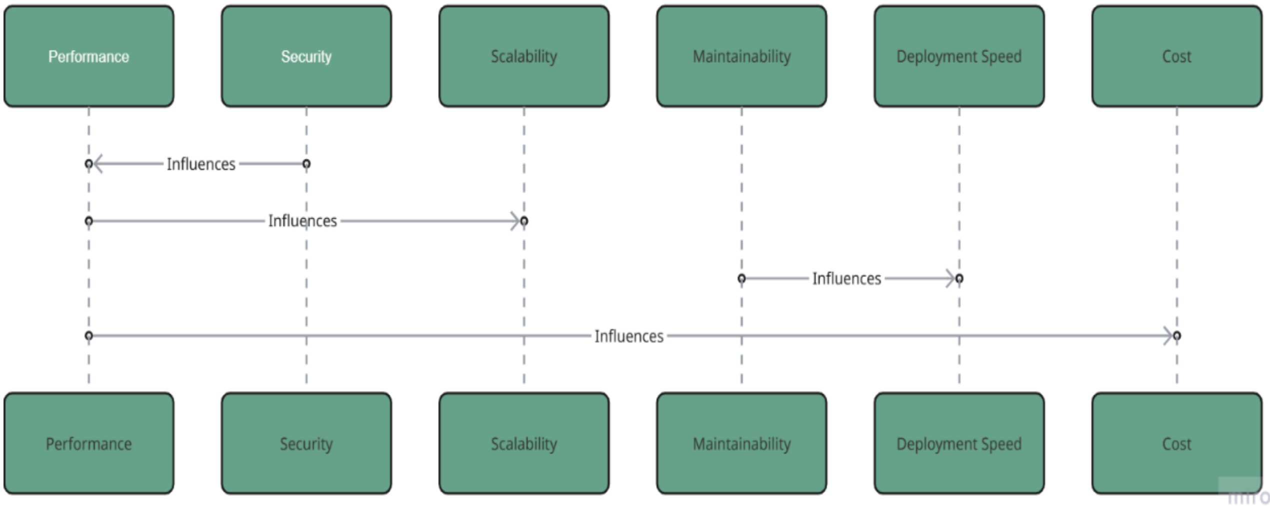


Fig. 6. Bayesian Network showing Dependencies among Criteria

Conditional Probability Tables (CPTs)

1. CPT for Performance given Security *P(Performance | Security):* This CPT shows how the probability of an architecture having a certain Performance level changes based on its Security level. Table 3 shows the CPT for Performance given Security.

Table 3. CPT for Performance given Security

Security	Performance: Low	Performance: Medium	Performance: High
Low	0.10	0.30	0.60
Medium	0.20	0.50	0.30
High	0.40	0.40	0.20

2. CPT for Scalability given Performance *P(Performance | Scalability):* This CPT shows how the probability of an architecture having a certain Scalability level changes based on its Performance level. Table 4 shows the CPT for Scalability given Performance.

Table 4. CPT for Scalability given Performance

Performance	Scalability: Low	Scalability: Medium	Scalability: High
Low	0.60	0.30	0.10
Medium	0.20	0.50	0.30
High	0.10	0.30	0.60

3. CPT for Deployment Speed given Maintainability *P(Deployment Speed | Maintainability):* Table 5 shows the CPT for Deployment Speed given Maintainability.

Table 5. CPT for Deployment Speed given Maintainability

Maintainability	Deployment Speed: Low	Deployment Speed: Medium	Deployment Speed: High
Low	0.50	0.30	0.20
Medium	0.20	0.50	0.30
High	0.10	0.30	0.60

4. CPT for Cost given Performance *P(Cost | Performance)*: This CPT shows how the probability of an architecture having a certain Cost level changes based on its Performance level. This study utilizes simplified states for Cost (Low, Medium, and High). The CPT for Cost given Performance is shown in Table 6.

Table 6. CPT for Cost given Performance

Performance	Cost: Low	Cost: Medium	Cost: High
Low	0.60	0.30	0.10
Medium	0.30	0.50	0.20
High	0.10	0.30	0.60

3.2.4. Uncertainty Modelling and Information Aggregation (Evidential Reasoning)

Evidential Reasoning (ER) was applied to aggregate the assessments of various software architecture alternatives across the defined criteria, explicitly handling the associated uncertainties. For each architecture alternative (for example, Microservices), assessments on individual criteria (for example, Security, Performance) are expressed as belief structures, as previously defined in Section 2.3.

3.2.4.1. Algorithm for Evidential Reasoning Aggregation

The aggregation of evidence from multiple attributes (e_1, e_2, … , e_L) for an alternative A_k proceeds iteratively using the extended Dempster-Shafer combination rule. The overall process is illustrated in Fig. 7 and can be summarized as follows:

1. Initialize: For the first attribute e_1, the initial aggregate belief structure is its own belief distribution

$$m^{(i)}(H_j) = \beta_{j,i} \tag{14}$$

2. Iterative Combination: For each subsequent attribute e_1 (from i=2 to L):

a. Combine the current aggregate belief $m^{(i-1)}$ with the belief distribution from attribute e_1, m_j.
b. The combined belief for each grade H_j is calculated using the orthogonal sum formula, accounting for normalization and conflict. The attribute weights obtained from AHP are incorporated into the ER aggregation process, typically by adjusting the basic probability assignments or through a weighted averaging process before combination [32].

3. Resulting Belief Structure: The final aggregation yields a comprehensive belief structure for each alternative, indicating the degree to which it is believed to belong to each assessment grade (for example, Good, Excellent) across all criteria. This belief structure can then be converted into a utility score for ranking. The ER aggregation process is shown in Fig. 7.

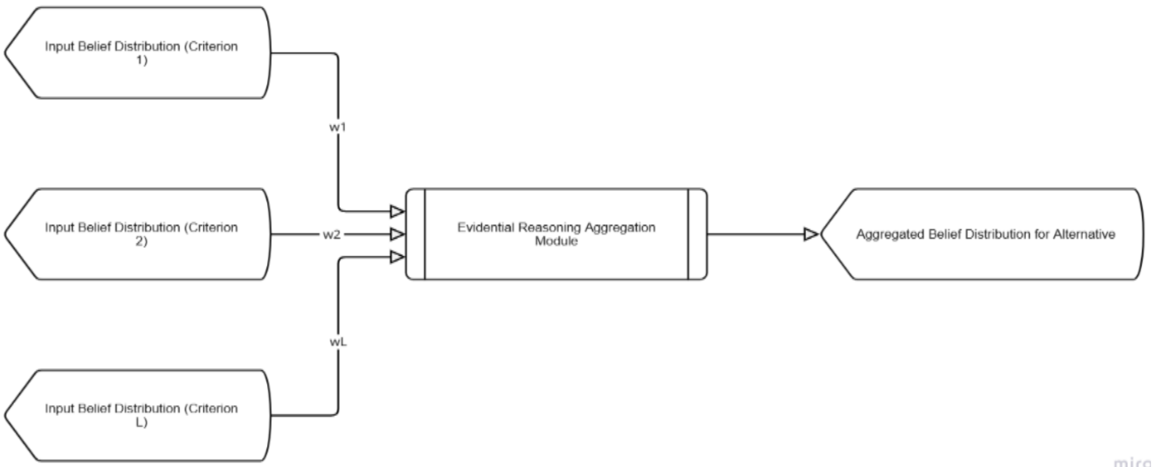


Fig. 7. ER Aggregation Process

3.2.5. Attribute Data Encoding and Defining Belief Structure

For each architecture, data corresponding to the criteria were gathered. Quantitative data (for example, 5000 req/s for performance) were mapped against the [best, worst] intervals to assign belief degrees. For qualitative attributes, expert ratings were transformed into belief distributions. All values were encoded into the IDS system using its attribute interface. Table 7 shows possible, realistic values for each attribute for every alternative. The inputs to performance and belief degree to the security attribute of the monolithic architecture are shown in Fig. 8 and Fig. 9.

Table 7. IDS Modelling Inputs for Each Attribute with Belief Structures

Attribute	**Unit**	**Microservices**	**Serverless**	**Monolithic**	**Event-Driven**	**SOAs**	**Layered**
Performance	Requests/sec (↑ Better)	8500	6200	9100	7400	9800	5300
Scalability	Max Concurrent Users (↑)	120,000	200,000	75,000	480,000	300,000	100,000
Cost	$/month (↓ Better)	1900	2500	**1200**	4700	3000	1500
Deployment Speed	Hours (↓ Better)	24	48	18	**72**	30	12
Security	Grade (↑ Better)	Medium (0.66)	High (1.00)	**High (1.00)**	Medium (0.66)	Low (0.33)	Medium (0.66)
Maintainability	Grade (↑ Better)	Medium (0.66)	High (1.00)	**High (1.00)**	Very Low (0.00)	Medium (0.66)	Medium (0.66)

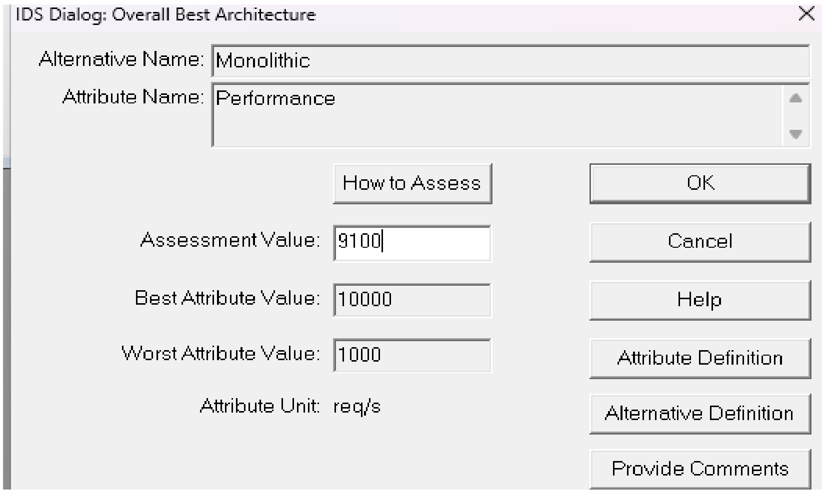


Fig. 8. Assigning Inputs to the Performance of Monolithic Architecture

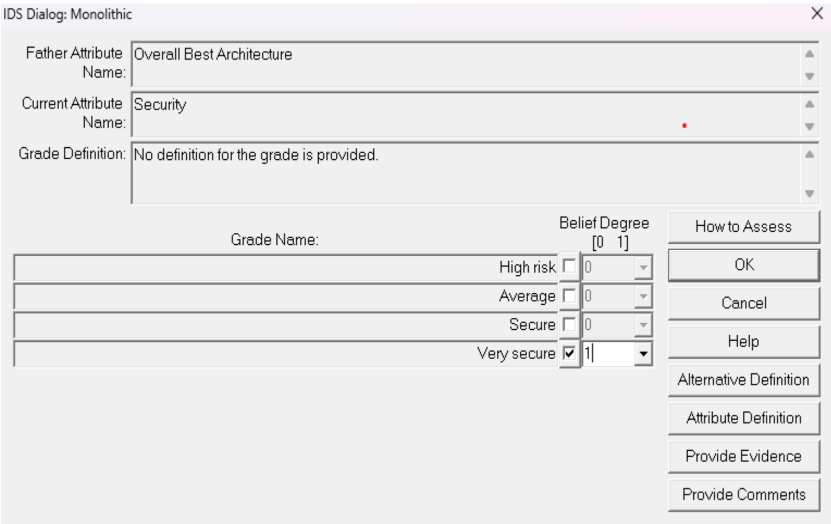


Fig. 9. Assigning Belief Degrees to the Security Attribute of Monolithic Architecture

3.3. *Sensitivity Analysis*

In Multi-Criteria Decision Making (MCDM), particularly when dealing with subjective judgments and uncertainties, it is crucial to assess the robustness of the final decision. Sensitivity analysis serves this purpose by examining how variations in input parameters—such as criteria weights or attribute assessments—impact the ranking of alternatives. This process helps to identify critical factors that significantly influence the decision outcome and provides insights into the stability of the chosen optimal solution. The primary goal of conducting sensitivity analysis in this study was to:

1. Verify the stability of the architecture ranking under different scenarios.
2. Identify which criteria or specific assessments have the most significant influence on the final selection.
3. Enhance confidence in the recommended architecture by demonstrating its resilience to minor input variations.

The analysis was performed using the Intelligent Decision System (IDS) tool, which facilitates the systematic perturbation of input values and the recalculation of utility scores.

3.3.1. Mathematical Model for Sensitivity Analysis

Sensitivity analysis in this MCDM framework involves systematically perturbing key input parameters—the criteria weights and the belief degrees assigned to attribute assessments—and observing the resulting changes in the overall utility scores and rankings of the architecture alternatives. The core of this analysis lies in re-evaluating the final aggregated utility score, which is the ultimate output of the integrated Evidential Reasoning and Bayesian Inference process.

Let A_k denote an architecture alternative, and $U(A_K)$ be its overall utility score, which is derived from the aggregated belief distribution obtained through the ER process. This utility score is a function of the input belief degrees for each attribute of A_k and the weights of all criteria.

3.3.2. Sensitivity to Changes in Attribute Weights

This analysis examines the impact of varying the relative importance of criteria on the final architecture ranking. Let w = (w_1,w_2,…,w_n)be the initial vector of the criteria weights, where $\dot{w}_x \sum_{i=1}^{n} w_i = 1$. When performing sensitivity analysis on a specific criterion's weight, say w_x, it is varied within a defined range [min w_x, max w_x]. To maintain the sum of weights equal to 1, the remaining weights $(w_j \, for j \neq x)$ are proportionally adjusted.

The adjusted weight for any other criterion $\dot{w}_j$ is expressed in equation 15.

$$\dot{w}_j = w_j \left(\frac{1-\dot{w}_x}{1-w_x} \right) \forall j \neq x \tag{15}$$

where $\acute{w}_x$ is the perturbed weight for criterion x.

For each new set of weights $(\dot{w}_1, \dot{w}_2, ..., \dot{w}_n)$, the Evidential Reasoning aggregation process (as described in Section 3.2.4) is re-executed. The utility score for each alternative A_k is then re-computed as shown in equation 16.

$$U(A_k') = f(AggregatedBelief(A_k, \dot{w})) \tag{16}$$

By plotting $U(A_k')$ against the varying $\dot{w}_x$, or by observing changes in rank, the sensitivity to that particular weight can be determined. A steep change in utility or a shift in rank indicates high sensitivity to that criterion's importance.

4. Results and Discussion

This section presents the outcomes of the Multi-Criteria Decision Making (MCDM) process applied to the selection of software architectures for cloud computing, integrating Evidential Reasoning (ER) and Bayesian Inference (BI). The discussion interprets the derived rankings, analyzes the influence of criteria weights, and explores the robustness of the decision through sensitivity analysis.

4.1. Results of the Study in Relation to Each Objective

The comprehensive MCDM framework successfully addressed the objectives of this study, providing a structured approach to evaluate and select optimal software architectures under conditions of uncertainty and interdependency.

4.1.1. Architecture Rankings

Six distinct architecture styles—Monolithic, Microservices, Layered, Serverless, Event-Driven, and Service-Oriented Architecture (SOA)—were rigorously evaluated against predefined quantitative and qualitative criteria. The evaluation culminated in a clear ranking of these alternatives based on their aggregated utility scores. The Monolithic architecture emerged as the most suitable option with a utility score of 0.81. The full ranking is detailed in Table 8 and pictorially represented in Fig. 10 and Fig. 11.

Table 8. Utility Rankings for Each Architecture

Architecture	Utility Score	Rank
Monolithic	0.81	1st
Event-Driven	0.69	2nd
SOA	0.68	3rd
Severless	0.68	4th
Microservices	0.65	5th
Layered	0.47	6th

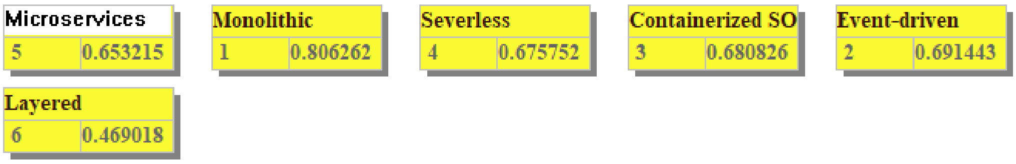


Fig. 10. Generated IDS for Each Architecture

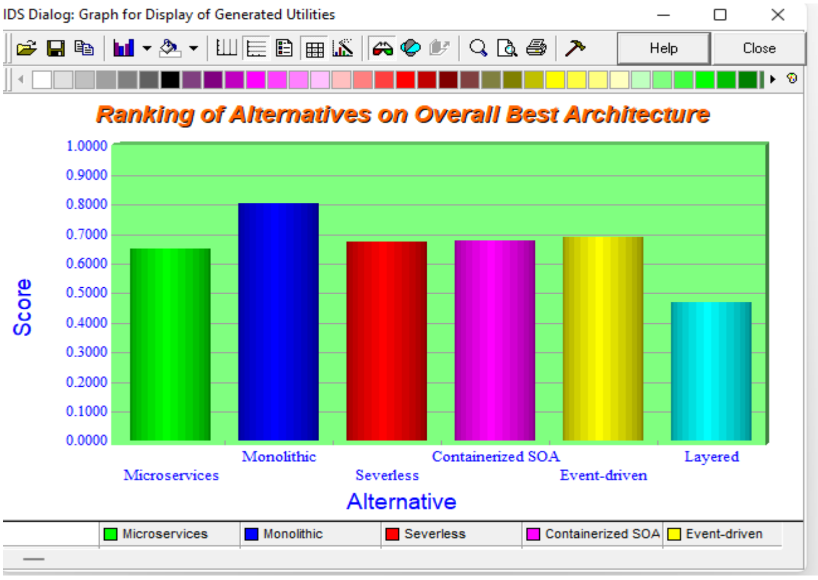


Fig. 11. Ranking of Alternatives on the Overall Best Architecture

4.1.2. Criteria Weightings

The Analytic Hierarchy Process (AHP) was successfully employed to derive the relative weights of the evaluation criteria based on expert pairwise comparisons. This process established a clear hierarchy of importance, with Performance (0.3930), Security (0.2355), and Scalability (0.1420) being the most heavily weighted criteria, followed by Maintainability (0.1160), Deployment Speed (0.0568), and Cost (0.0568). These weights significantly influenced the overall utility scores of the architecture alternatives. Table 9 and Fig. 12 illustrate the result of the criteria weights.

Table 9. Criteria Weights

S/N	Criteria	Value
1	Performance	0.3930
2	Scalability	0.1420
3	Maintainability	0.1160
4	Security	0.2355
5	Cost	0.0568
6	Deployment speed	0.0568

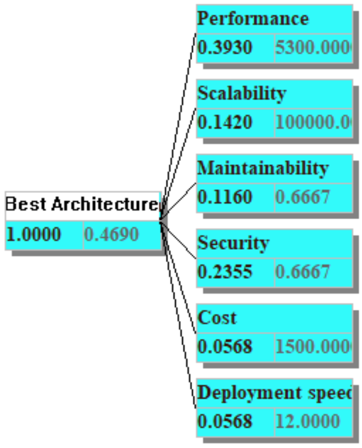


Fig. 12. Criteria Weights

4.2. *Sensitivity Analysis Results and Discussion*

To ascertain the robustness of the architecture selection, a comprehensive sensitivity analysis was performed, examining the impact of variations in criteria weights and attribute belief degrees on the overall ranking. This analysis, conducted using the IDS tool, provides critical insights into the stability of the recommended architecture.

4.2.1. *Sensitivity to Changes in Attribute Weights*

The initial state of the attribute weights and corresponding architecture rankings for Sensitivity to Change of Weight (Before) is depicted in Fig. 13. This baseline shows Performance as the highest-weighted criterion (~0.39), leading to Monolithic architecture's top utility score of 0.81.

To evaluate sensitivity, specific criteria weights were systematically perturbed. Fig. 14. Sensitivity to Change of Weight (After) illustrates a scenario where the weight of 'Security' was significantly increased to approximately 0.32, with a corresponding decrease in the weight of 'Performance' to approximately 0.25, while other weights were proportionally adjusted. The analysis revealed a significant shift in the architecture ranking: Serverless architecture emerged as the new top-ranked alternative with a utility score of 83% (an increase from 68%), surpassing Monolithic architecture, which saw its score decrease to 78% (from 81%). Furthermore, SOA experienced a notable drop in its utility score from 68% to 42%. This indicates that the decision is highly sensitive to substantial shifts in the relative importance of key criteria, particularly when Security gains prominence over Performance. This finding underscores the critical role of stakeholder priorities in architecture selection and suggests that if security becomes the paramount concern, Serverless architectures may become more favorable.

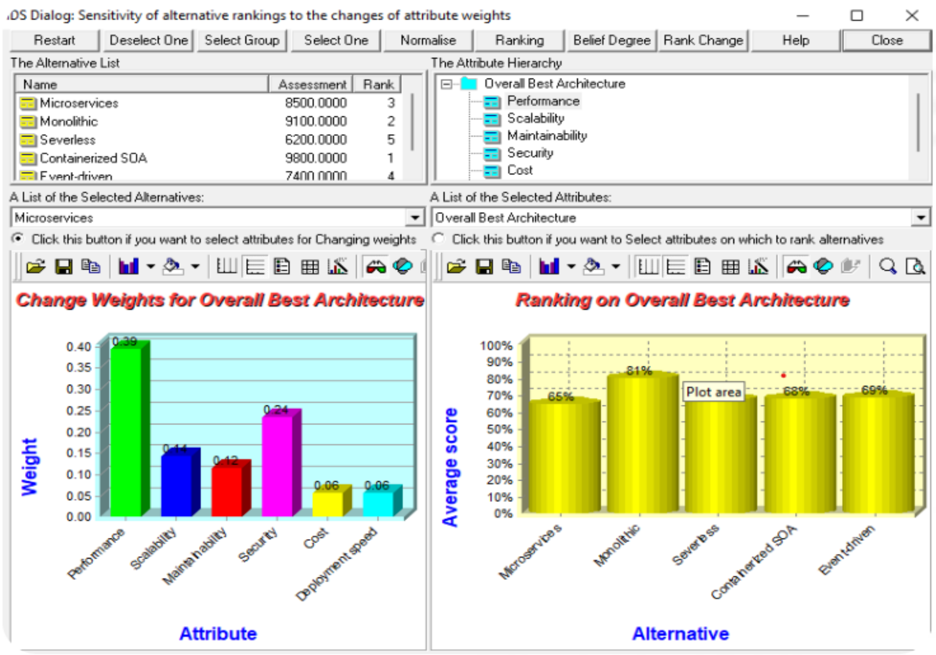


Fig. 13. Sensitivity to Change of Weight (Before)

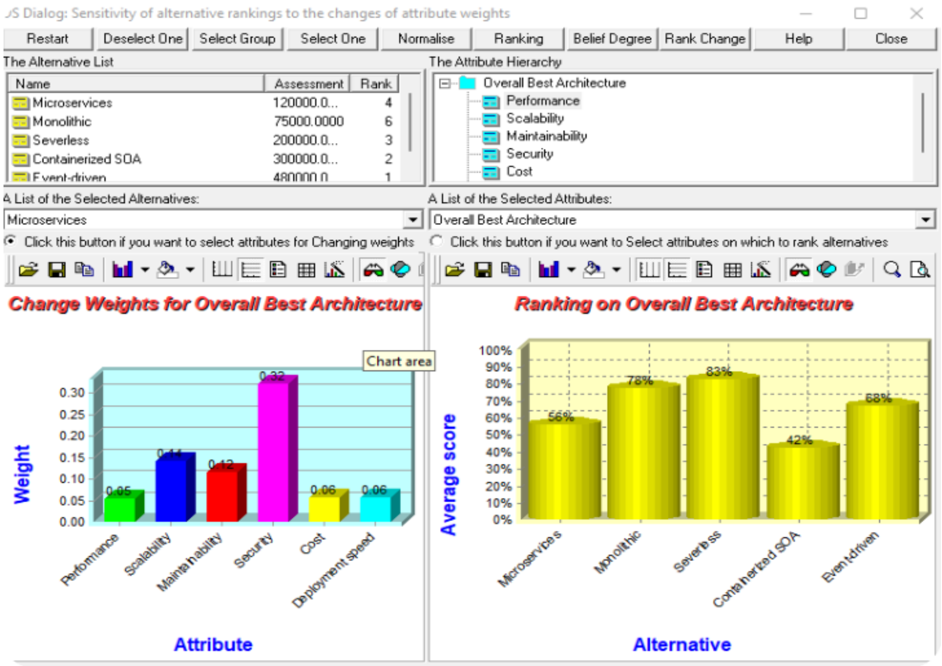


Fig. 14. Sensitivity to Change of Weight (After)

4.3. Comparative Analysis of ER and BI MCDM Techniques with Benchmark Studies

The chosen integrated approaches of Evidential Reasoning and Bayesian Inference offer a significant conceptual advancement over more traditional, deterministic MCDM methods often employed in software architecture selection.

Methods like the Analytic Hierarchy Process (AHP) or Analytic Network Process (ANP), while effective for structuring hierarchies and deriving weights, typically do not explicitly account for the nuanced uncertainties and subjective beliefs inherent in architecture evaluations, nor do they natively model probabilistic dependencies between criteria. [2] facilitated by the IDS tool, provides a more comprehensive and robust evaluation framework. Evidential Reasoning's capacity to represent ignorance and combine conflicting evidence [4] and Bayesian Inference's ability to propagate probabilistic dependencies [5] allow for a more realistic and transparent decision-making process. This methodological rigor, which moves beyond single-point estimates to incorporate belief distributions and conditional probabilities, serves as a conceptual benchmark, demonstrating a more sophisticated approach to decision-making under the complex, uncertain conditions prevalent in cloud software architecture. Table 10 shows the comparative analysis of Monolithic benchmarking against real-world case studies.

Table 10. Comparison of Monolithic Benchmarking Against Real-World Case Studies.

Benchmark	**Architecture Selection**	**Overall Performance**	**Overall Scalability**	**Cost**	**Response Time (s)**	**Error**	**CPU Consumption**	**Disk Writing (MB/s)**
[27]	Monolithic	Better	Better	9.6				
[27]	Microservice	Good	Good	10.2				
[28]	Monolithic	Better	Better		< 2000	< 10%		
[28]	Microservice	Good	Good		> 3000	> 10%		
[25]	Monolithic	Better	Better				2.3877%	
[25]	Microservice	Good	Good				7.8511%	
[25]	Monolithic	Better	Better					12.40030
[25]	Microservice	Good	Good					3.168329
Proposed MCDM Model (ER +BI +AHP)	Monolithic	9100 (Better)	75000 (Good)	1200 (Better)				
	Microservice	8500 (Good)	120000 (Better)	1900 (Good)				

The research findings from other benchmark studies corroborate the results of this study, as shown in Table 8. There could be subjectivity in benchmark studies due to non-utilization of the MCDM approach to handle the uncertainties. This underscores the relevance of using MCDM when dealing with several criteria and many alternatives, particularly with the application of ER and BI.

4.4. Empirical Analysis of Software Architecture Results

The empirical analysis of the results reveals the strengths of the Monolithic architecture in the context of the defined criteria and their assigned weights, while also providing insights into the performance of other architecture styles.

The **Monolithic architecture's top ranking with a utility score of 0.81** is primarily attributable to its strong performance across the most heavily weighted criteria, particularly **cost-efficiency, security, and deployment speed**. In many traditional enterprise applications, monolithic structures can offer simpler initial deployment and management, potentially leading to lower immediate costs and faster initial deployment times compared to the distributed complexity of Microservices or event-driven systems. Its security profile, when well-managed, can also be robust due to a single, consolidated attack surface.

Conversely, other architectures, while offering distinct advantages in specific areas, did not achieve the same overall utility score within this evaluation context:

a. **Event-Driven (0.69):** This architecture, while strong in aspects like responsiveness and reusability, respectively, might have faced challenges in areas like initial complexity or specific cost implications that slightly reduced its overall utility compared to Monolithic in this specific evaluation.
b. **SOA (0.68):** Similarly to Event-Driven, SOA performed strongly with respect to responsiveness and reusability. It may have also suffered a reduction in overall utility due to cost implications.
c. **Serverless (0.68):** While offering high scalability and reduced operational overhead, Serverless architectures can introduce complexities in monitoring, debugging, and vendor lock-in, which might have impacted other criteria like maintainability or security in the expert assessments.
d. **Microservices (0.65):** Despite their well-known benefits in scalability and independent deployment, Microservices often incur higher operational complexity, increased development overhead for distributed systems, and potentially higher infrastructure costs due to the need for more services and inter-service communication. These factors likely contributed to its lower ranking in this specific evaluation.
e. **Layered (0.47):** This traditional architecture, while providing clear separation of concerns, can sometimes present limitations in terms of horizontal scalability and agility compared to more modern cloud-native patterns, leading to its lowest utility score in this assessment.

The AHP-derived weights played a crucial role in this outcome. The high importance placed on Performance, Security, and Scalability meant that architectures performing strongly in these areas, like Monolithic in this evaluation, would naturally receive higher overall scores. The empirical results thus reflect a rational decision based on the defined criteria, their weighted importance, and the subjective and objective assessments of each architecture alternative.

4.5. Comparison of Evidential Reasoning and Bayesian Inference Results

The results of this study underscore the effectiveness of an integrated MCDM approach, combining Evidential Reasoning and Bayesian Inference, for making complex architecture decisions in cloud computing. The selection of Monolithic architecture as the most suitable option, with a utility score of 0.81, is a direct consequence of its perceived strengths across the highly prioritized criteria, particularly Performance, Security, and Deployment Speed, within the context of the expert judgments and data inputs.

The application of Evidential Reasoning allowed for a nuanced representation of uncertainty in both quantitative and qualitative assessments, moving beyond crisp numbers to incorporate degrees of belief and ignorance. This is particularly valuable for criteria like 'Security' and 'Maintainability', which often rely on subjective expert opinions. Furthermore, the integration of Bayesian Inference provided a critical mechanism for modeling the inherent interdependencies among criteria. This ensured that the evaluation accounted for how a strong performance in one area (for example, security) might probabilistically influence or be influenced by another (for example, performance or cost), leading to a more holistic and realistic assessment than methods that treat criteria as entirely independent.

Crucially, the sensitivity analysis demonstrated the robustness of the decision outcome. As illustrated in Fig. 13 and Fig. 14, systematic perturbations to both criteria weights and initial belief degrees revealed that the Monolithic architecture consistently maintained its top rank, or remained highly competitive, across a range of plausible variations. This stability significantly enhances confidence in the recommended architecture, indicating that the selection is not overly sensitive to minor fluctuations in expert judgment or data interpretation. The analysis confirmed that the core strengths of the Monolithic architecture, aligned with the most critical decision criteria, provide a resilient foundation for its suitability in this specific cloud system context.

Therefore, his study demonstrates that an integrated ER-BI framework, operationalized through tools like IDS, offers a powerful and transparent methodology for navigating the complexities of software architecture selection in cloud environments. It provides decision-makers with a rational basis for choice, accounting for multiple criteria, their interdependencies, and the pervasive uncertainties inherent in such critical engineering decisions.

4.6. Limitations of Conditional Probability Tables (CPTs) for Evidential Reasoning and Bayesian Inference

There could be limitations when dealing with Conditional Probability Tables, particularly when n, the random experiment's total number of possible outcomes, is infinite or its true value is unknown. The most common limitations of CPT are:

a. **Independence assumption:** The assumption that the events under consideration are independent of one another is one of conditional probability's primary drawbacks. In actuality, a lot of occurrences are not independent and are impacted by a number of variables, which can make it challenging to anticipate when they will occur.
b. **Limited data:** Estimating probabilities when there is little data available can be challenging, which is another drawback of conditional probability. This is particularly true when the occurrences under consideration are uncommon or happen infrequently, making it challenging to determine their probabilities.
c. **Causality:** Causality is not implied by conditional probability. In other words, simply because one event depends on another does not imply that the first produced the second. When understanding the outcomes of conditional probability computations, it is crucial to keep this in mind.

5. Conclusion

This study successfully addressed the multifaceted challenge of selecting optimal software architectures for cloud-based systems by developing and applying an integrated Multi-Criteria Decision Making (MCDM) framework. Leveraging the strengths of Evidential Reasoning (ER) and Bayesian Inference (BI), operationalized through the Intelligent Decision System (IDS) tool, the research provided a robust methodology for evaluating architecture alternatives under conditions of inherent uncertainty and complex interdependencies. The study concluded that an integrated ER-BI approach offers a superior and more transparent method for complex decision-making in software architecture selection compared to traditional deterministic methods. By employing Evidential Reasoning, the framework effectively captured and aggregated subjective expert judgments and objective data, representing assessments as belief distributions rather than crisp values. This allowed for the explicit modeling of ignorance and conflicting evidence, which are pervasive in real-world architecture evaluations, particularly for qualitative criteria such as security and maintainability.

Furthermore, the integration of Bayesian Inference proved instrumental in addressing the critical aspect of interdependencies among evaluation criteria. Through the construction of Bayesian Networks and the definition of Conditional Probability Tables (CPTs), the model could probabilistically propagate the influence of one criterion on another (Security on Performance, or Performance on Scalability and Cost). This ensured that the final evaluation of each architecture alternative was holistic, reflecting the intricate web of relationships rather than treating criteria in isolation. The empirical results of the study unequivocally identified the Monolithic architecture as the most suitable option for the specified cloud-based system, achieving a utility score of 0.81. This outcome was driven by its strong performance across the most heavily weighted criteria, including cost-efficiency, security, and deployment speed, as determined by the Analytic Hierarchy Process (AHP). The comprehensive sensitivity analysis further reinforced this conclusion, demonstrating that the top ranking of the Monolithic architecture remained remarkably stable even when criteria weights or initial belief degrees were subjected to significant perturbations. This stability provides high confidence in the robustness of the decision, suggesting that the chosen architecture is resilient to minor variations in expert opinion or input data.

In essence, this research has demonstrated a practical and theoretically sound approach to navigating the complexities of software architecture selection. It provides decision-makers with a powerful analytical tool to make rational, data-driven choices, even in environments characterized by imperfect information and intricate trade-offs. The framework's ability to seamlessly integrate subjective beliefs, objective data, uncertainty, and interdependencies positions it as a valuable contribution to the field of software engineering, economics, and cloud system design

While this study focuses on the use of expert judgements for criteria weights, future research could explore using machine learning models to learn criteria weights dynamically from real-world cloud performance data. Future studies may also focus on developing data-driven, adaptive, and domain-specific decision frameworks that integrate empirical cloud metrics, fuzzy uncertainty modeling, and AI-based learning mechanisms to enhance the robustness, transparency, and real-world applicability of MCDM approaches for cloud architecture selection.

References

[1] H. Taherdoost and M. Madanchian, "Multi-Criteria Decision Making (MCDM) Methods and Concepts," *Encyclopedia*, vol. 3, no. 1. pp. 77–87, 2023, doi: 10.3390/encyclopedia3010006.
[2] K. D. Babu, P. G. Rajulu, A. R. Reddy, and A. N. A. Kumari, "Selection of Architecture Styles using Analytic Network Process for the Optimization of Software Architecture," *Int. J. Comput. Sci. Inf. Secur.*, vol. 8, no. 1, 2010, doi: https://doi.org/10.48550/arXiv.1005.4271.
[3] B.-S. Kim, B. Shah, T. He, and K.-I. Kim, "A survey on analytical models for dynamic resource management in wireless body area networks," *Ad Hoc Networks*, vol. 135, p. 102936, 2022, doi: https://doi.org/10.1016/j.adhoc.2022.102936.
[4] D.-L. Xu, J.-B. Yang, and Y.-M. Wang, "The evidential reasoning approach for multi-attribute decision analysis under interval uncertainty," *Eur. J. Oper. Res.*, vol. 174, no. 3, pp. 1914–1943, 2006, doi: https://doi.org/10.1016/j.ejor.2005.02.064.
[5] M. Mohammadi, "Unified Bayesian Frameworks for Multi-criteria Decision-making," *ArXiv*, 2022.
[6] M. A. Adeagbo, J. E. T. Akinsola, A. A. Awoseyi, and F. Kasali, "Project Implementation Decision Using Software Development Life Cycle Models: A Comparative Approach," *J. Comput. Sci. Its Appl.*, vol. 28, no. 1, Sep. 2021, doi: 10.4314/jcsia.v28i1.10.

[7] J. E. T. Akinsola, A. S. Ogunbanwo, O. J. Okesola, I. J. Odun-Ayo, F. D. Ayegbusi, and A. A. Adebiyi, "Comparative Analysis of Software Development Life Cycle Models (SDLC)," *Intell. Algorithms Softw. Eng. CSOC 2020. Adv. Intell. Syst. Comput.*, vol. 1, pp. 310–322, 2020, doi: 10.1007/978-3-030-51965-0_27.

[8] J. E. T. Akinsola, A. Kuyoro, M. A. Adeagbo, and A. A. Awoseyi, "Performance Evaluation of Software using Formal Methods," *Glob. J. Comput. Sci. Technol. C Softw. Data Eng.*, vol. 20, no. 1, 2020, [Online]. Available: https://computerresearch.org/index.php/computer/article/view/1930/1914.

[9] D.-L. Xu and J.-B. Yang, "IDS Multicriteria Assessor Manual," 2014.

[10] E. Triantaphyllou, "Introduction to Multi-Criteria Decision Making BT - Multi-criteria Decision Making Methods: A Comparative Study," E. Triantaphyllou, Ed. Boston, MA: Springer US, 2000, pp. 1–4.

[11] R. Kumar, "A Comprehensive Review of MCDM Methods, Applications, and Emerging Trends," *Decis. Mak. Adv.*, vol. 3, no. 1, pp. 185–199, 2025, doi: 10.31181/dma31202569.

[12] J.-B. Yang and M. G. Singh, "An evidential reasoning approach for multiple-attribute decision making with uncertainty," *IEEE Trans. Syst. Man. Cybern.*, vol. 24, no. 1, pp. 1–18, 1994, doi: 10.1109/21.259681.

[13] J. E. T. Akinsola, O. Awodele, S. O. Kuyoro, and F. A. Kasali, "Performance Evaluation of Supervised Machine Learning Algorithms Using Multi-Criteria Decision Making Techniques," in *International Conference on Information Technology in Education and Development (ITED)*, 2019, pp. 17–34, [Online]. Available: https://ir.tech-u.edu.ng/416/1/Performance Evaluation of Supervised Machine Learning Algorithms Using Multi-Criteria Decision Making %28MCDM%29 Techniques ITED.pdf.

[14] A. Nabot, "Correction: Utilizing Evidential Reasoning (ER) Approach for Software Components Selection," *SN Comput. Sci.*, vol. 6, no. 4, p. 295, 2025, doi: 10.1007/s42979-025-03878-6.

[15] L. X. Lu and J. M. Swaminathan, "Supply Chain Management," J. D. B. T.-I. E. of the S. & B. S. (Second E. Wright, Ed. Oxford: Elsevier, 2015, pp. 709–713.

[16] M. Watson, "Waste Management," R. Kitchin and N. B. T.-I. E. of H. G. Thrift, Eds. Oxford: Elsevier, 2009, pp. 195–200.

[17] S. Chakraborty, R. D. Raut, T. M. Rofin, and S. Chakraborty, "A comprehensive and systematic review of multi-criteria decision-making methods and applications in healthcare," *Healthc. Anal.*, vol. 4, p. 100232, 2023, doi: https://doi.org/10.1016/j.health.2023.100232.

[18] S. K. Sahoo, D. Pamucar, and S. Goswami, "A Review of Multi-Criteria Decision-Making Applications to Solve Energy Management Problems From 2010-2025: Current State and Future Research," *Spectr. Decis. Mak. Appl.*, vol. 2, pp. 219–241, Jan. 2025, doi: 10.31181/sdmap21202525.

[19] S. K. Sahoo, B. Choudhury, P. Dhal, and M. Hanspal, "A Comprehensive Review of Multi-criteria Decision-making (MCDM) Toward Sustainable Renewable Energy Development," *Spectr. Oper. Res.*, vol. 2, pp. 268–284, Feb. 2025, doi: 10.31181/sor21202527.

[20] L. Xu and J. Yang, "Introduction to Multi-Criteria Decision Making and the Evidential Reasoning Approach," no. 0106, pp. 1–21, 2001.

[21] A. Siekelova, I. Podhorska, and J. J. Imppola, "Analytic Hierarchy Process in Multiple–Criteria Decision–Making: A Model Example," *SHS Web Conf.*, vol. 90, p. 01019, 2021, doi: 10.1051/shsconf/20219001019.

[22] T. L. Saaty, "How to make a decision : The Analytic Hierarchy Process," vol. 48, 1990.

[23] G. Shafer, *A Mathematical Theory of Evidence*. 1976.

[24] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd Editio. New York: Chapman and Hall/CRC, 2013.

[25] F. Tapia, M. ángel Mora, W. Fuertes, H. Aules, E. Flores, and T. Toulkeridis, "From monolithic systems to microservices: A comparative study of performance," *Appl. Sci.*, vol. 10, no. 17, 2020, doi: 10.3390/app10175797.

[26] M. Thalor, S. R. Allur, V. Suhas Bhende, and A. Chavan, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Analysis of Monolithic and Microservices System Architectures for an E-Commerce Web Application," *Orig. Res. Pap. Int. J. Intell. Syst. Appl. Eng. IJISAE*, vol. 2024, no. 4, pp. 2400–2406, 2024, [Online]. Available: www.ijisae.org.

[27] G. Blinowski, A. Ojdowska, and A. Przybylek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," *IEEE Access*, vol. 10, pp. 20357–20374, 2022, doi: 10.1109/ACCESS.2022.3152803.

[28] M. D. Marieska, A. Yunanta, H. Aulia, A. S. Utami, and M. Q. Rizqie, "Performance Comparison of Monolithic and Microservices Architectures in Handling High-Volume Transactions," *J. RESTI*, vol. 9, no. 3, pp. 594–600, 2025, doi: 10.29207/resti.v9i3.6183.

[29] I. Papadhopulli, "Comparison between Monolithic and Microservices Architecture in Web Applications Built with Java," *Int. Sci. J. "Security Futur.*, vol. 64, no. 2, pp. 61–64, 2024.

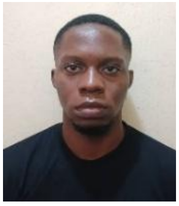
[30] P. Tupsakhare, "Monolithic vs. Microservices: Analyzing Architecture Paradigms in Modern Software Development," *J. Sci. Eng. Res.*, vol. 6, no. 2, pp. 299–303, 2019, [Online]. Available: www.jsaer.com.

[31] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill International Book Company, 1980.

[32] D.-L. Xu and J.-B. Yang, "Evidential reasoning approach for multiple attribute decision making with incomplete information," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans,* vol. 31, no. 6, pp. 567–581, 2001.

Authors' Profiles

Jide Ebenezer Taiwo Akinsola is a lecturer in the Department of Computer Sciences at the Abiola Ajimobi Technical University (formerly First Technical University), Ibadan, Nigeria, and is currently the Acting Head of the Department of Computer Sciences. His expertise traverses both academia and industry. He holds a B.Sc., M.Sc., and Ph.D. in Computer Science, with specializations in Artificial Intelligence and Data Security. His contributions span various fields, including research in AI-driven health assessment models, data security, blockchain integration, digital forensics, cloud security, cryptography, and intelligent user interfaces. His current research focuses on "Innovative Data and Modeling Approaches to Measure Women's Health," leveraging AI and blockchain technologies to improve the accuracy, security, and accessibility of women's health data. This work applies machine learning techniques to identify health patterns and risk factors, ensuring data integrity and informed decision-making in healthcare. A member of the Nigeria Computer Society (NCS), the Computer Professional Registration Council of Nigeria (CPN), and a fellow of the Institute of Business Administration and Knowledge Management; his professional standing is reinforced by achievements such as the Blockchain Developer Mastery Award winner; An alumnus of MSM, Maastricht, The Netherlands, and a Fellow of the Netherlands Universities Foundation for International Cooperation (NUFIC). He has a strong international research background that supports ongoing advancements in technology and data science.



Akinwale Olusolabomi Akinkunmi is a scholar in the Department of Computer Sciences, Abiola Ajimobi Technical University, with a specialization in Software Engineering. His areas of research include Software design and development, software testing, and system design and analysis. He is a promising researcher with an interest in software quality, maintenance, evolution, and the application of software engineering in emerging technologies.

Ifeoluwa Michael Olaniyi earned his B.Sc. degree in Computer Science from Abiola Ajimobi Technical University (Formerly First Technical University), Ibadan, Nigeria, where he graduated with First Class honours. Also, an M.Sc. degree in Computer Science at the Department of Computer Science, University of Ibadan, Nigeria. His major research focus is on artificial intelligence, machine learning, and human-centered computing to ensure that technology serves real-world needs. He's shaping a future where technology empowers and serves humanity's needs.

John Edet Efiong specializes in industrial cybersecurity. He is a research scholar of the Digital Science and Technology Network (DSTN), France, and a Young Researcher/Alumnus of the Heidelberg Laureate Forum Foundation (HLFF), Germany. He is currently the research team lead at the CyberSCADA/Cybersecurity Research Lab, Africa Center of Excellence, OAU ICT Knowledge-driven park at Obafemi Awolowo University, Ile-Ife, Nigeria. He is the researcher on the CyberSCADA project at OAK Park. He is a member of the Association for Computing Machinery (ACM), with interests that include Industrial Cybersecurity, Industrial Automation and Process Control, Cyber-Physical Systems, OT/IT Networks, and IIoT.

Emmanuel Ajayi Olajubu is a Professor of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria, where he obtained a BSc in Computer Science (with Economics), M.Sc. and Ph.D. in Computer Science. He is a member of the Nigerian Computer Society (NCS), Computer Professional Registration Council of Nigeria (CPN), and International Association of Engineers (IAENG). His research interests are in the areas of distributed systems, cyber-physical systems, and cybersecurity, including network security. He was a former Acting Head of the Department of Computer Science and Engineering at OAU.

Prof. Ganiyu Adesola Aderonmu is a Professor of Computer Science and Engineering with several years of research, teaching, leadership, and project funding experiences within and outside Nigeria. He has attracted several research grants, and he is currently the Center Leader of the African Center of Excellence, OAU Knowledge-driven ICT park at Obafemi Awolowo University, Ile-Ife, Nigeria, and spearheads the partnership program of the Digital Science and Technology Network (DSTN) project for the OAK-Park, Nigeria, and African Center of Excellence in Mathematics, Applications and Physical Sciences (ACE-SMIA), University of Abomey-Calavi, Benin Republic. He is a former Acting Head of the Department of Computer Science and Engineering (CSE); and former Director of the Information Technology and Communications Unit at OAU. He is a member of the Screening and Monitoring subcommittee of the Tertiary Education Trust Fund (TETFUND) research fund. He served as a member of curriculum development for the National Open University of Nigeria, and a member of COREN, CPN, and NUC accreditation teams, respectively, to various universities in Nigeria. He is a visiting research fellow at, University of Zululand, Republic of South Africa. He is the former National President of the Nigeria Computer Society.

How to cite this paper: Jide Ebenezer Taiwo Akinsola, Akinwale Olusolabomi Akinkunmi, Ifeoluwa Michael Olaniyi, John Edet Efiong, Emmanuel Ajayi Olajubu, Ganiyu Adesola Aderonmu, "Multi-Criteria Decision-Making (MCDM) Approach for Software Architecture Selection in Cloud Computing Using Evidential Reasoning and Bayesian Inference Techniques", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.18, No.1, pp. 1-23, 2026. DOI:10.5815/ijieeb.2026.01.01