Cluster Computing (2021) 24:1103–1121
https://doi.org/10.1007/s10586-020-03178-z

Cloud services security-driven evaluation for multiple tenants

Sarah Maroc[1] · Jian Biao Zhang[1]

Received: 18 October 2019 / Revised: 20 August 2020 / Accepted: 25 August 2020 / Published online: 9 September 2020

Abstract
Cloud Computing has become a reliable solution for outsourcing business data and operation with its cost-effective and resource-efficient services. A key part of the success of the cloud is the multi-tenancy architecture, where a single instance of a service can be shared between a large number of users, also known as tenants. Service selection for multiple tenants presents a real challenge that has not been properly addressed in the literature so far. Most of the existing cloud services selection approaches are designed for a single-user, and hence are inefficient when applied to the case of a large group of users with different, and often, conflicting requirements. In this paper, we propose a multi-tenant cloud services evaluation framework, whereby service selection is carried out per group of tenants that can belong to different service classes, rather than per a single user. We formulate the cloud services selection for multi-tenants as a complex multi-attribute large-group decision-making (CMALGDM) problem. Skyline method is initially applied to reduce the search space by eliminating the dominated services regardless of tenants' requirements. Tenants are clustered based on their profiles characterized by different personal, service, and environmental features. Each tenant is assigned a weight to reflect its importance in the decision-making. The weight of a tenant is determined locally based on its closeness to the group decision and globally by combining its local weight with its corresponding cluster weight to reflect its total contribution to the overall decision-making. The final ranking of the alternatives is guided by a dynamic consensus process to reach a final solution with the highest level of agreement. The proposed framework supports multiple types of information, including deterministic data, interval numbers, and fuzzy numbers, to realistically represent the heterogeneity and uncertainty of security information.

Keywords Cloud computing · Security evaluation · Multi-tenancy · Decision-making

1 Introduction

1.1 Motivation

Cloud computing is rapidly growing in popularity and has become a reliable solution for outsourcing business data and operations. Gartner, a well-known IT consulting firm, claimed that by 2020, a corporate with 'no-cloud' policy will be a thing of the past, with more than 83% of enterprise workloads will be in the cloud [1]. With this significant increase in cloud adoption and the large number of services with similar functionalities, to evaluate and select the cloud services that best fulfill user's requirements becomes difficult. And is even more challenging in the case of multiple users with different services classes, and often conflicting, requirements.

In practice, the number of cloud users is very large, and multiple users may simultaneously require services with the same functionalities but different requirements. This is referred to as the multi-tenancy architecture [2], which is one of the key parts to the success of the cloud model. It offers the ability to share computing resources (e.g., networks, servers, storage, applications, and services) between multiple users to reduce the operational cost and benefit from the economy of scale. For instance, Salesforce has more than 150,000 customers with a number of multi-tenancy instances to serve thousands of tenants at the same time [3]. A tenant can represent different divisions that belong to the same company or entirely different organizations that serve multiple users with different objectives and governance needs.

✉ Sarah Maroc
ms@emails.bjut.edu.cn

[1] Beijing Key Laboratory of Trusted Computing, Faculty of Information Technology, Beijing University of Technology, Beijing, China

Cloud tenants have their own requirements and risk tolerance levels. For example, one tenant may require fast response time regardless of service cost, while another tenant may be primarily concerned with the security functionalities of the service. On the other hand, public cloud service providers, which serve a large number of users, are less flexible to adapt to a particular user's needs [4]. Indeed, cloud providers offer various services packages to satisfy customized requirements from different users. For example, Salesforce [3] provides Sales CRM services in different packages ranging from essentials targeting small businesses to professional, enterprise, and unlimited editions for large enterprises. Each service class corresponds to a given quality of service (QoS) with several different functionalities. Users with strict requirements can opt for the advanced editions with premium features. However, these users only account for a small part of all users, and the majority of users opt for lightweight services with limited QoS features. Most services in these packages are provided with a standard security mechanism for all tenants, which makes satisfying multiple tenants' security requirements a major challenge for cloud providers.

Moving from services selection for a single user to a large group of users with different requirements adds more complexity to the problem scope. Chen [5] termed this type of decision-making problem as the complex multi-attribute large-group decision-making (CMALGDM) problem, which is characterized by the following four features: (a) a large group (usually more than 20) decision-makers with different importance weights and conflicting requirements; (b) a decision-making environment with variable locations and temporal information; (c) the interdependencies between decision criteria; and (d) the uncertain and fuzzy preferences information of the decision-makers. Existing cloud services evaluation approaches are mostly designed for a single user, thus might not be suitable for complex levels of group decision-making. Additionally, given the dynamic and multi-tenant cloud environment, reducing the complexity of the computation process is of paramount importance to effectively and efficiently respond to large volumes of service requests.

There is a considerable amount of literature on cloud services selection and evaluation. To name a few, Garg et al. [6] proposed an AHP-based framework for ranking cloud services based on the user's QoS requirements. The authors in [7] used the Best-Worst method to prioritize the QoS criteria and the TOPSIS method to rank cloud services alternatives. Ding et al. [8] evaluated the trustworthiness of cloud services and employed a collaborative filtering technique to deal with missing and unavailable data. Hammadi et al [9] proposed a cloud services selection framework based on SLA management consisting of pre-interaction and post-interaction SLA evaluation processes to support users in making informed decisions regarding service suitability and continuity. The authors in [10] addressed services selection problem in federated cloud architecture using grade and joint probability distribution techniques. Sun et al. [11] proposed a fuzzy user-oriented cloud service selection system combining semantic ontologies and MCDM techniques. A recent and extensive literature reviews on cloud services evaluation methods can be found in [12], [13].

Most of the existing research on cloud services evaluation has largely focused on performance-related attributes. Even when security is considered in the evaluation, it is mostly treated as a single attribute that is often assigned a subjective value in a purely qualitative categorization. While there are still no acceptable frameworks for evaluating the security level of cloud services, increased interest in building such frameworks has been witnessed recently. Taha et al. [14] proposed an AHP-based cloud services security-driven approach using the cloud control matrix (CCM) [15] security framework as evaluation criteria. Modic et al. [16] proposed a cloud security assessment technique, called Moving Intervals Process (MIP), aimed at decreasing the time complexity of the assessment algorithm by separating scores of services providers that can exactly fulfill customers' needs from those that are under-provisioning or over-provisioning. Halabi and Bellaiche [17] presented a security self-evaluation methodology for cloud providers. Alabool and Mahmood [18] proposed a framework for ranking and improving IaaS cloud providers by identifying the weaknesses and less performing attributes.

A small number of studies have considered the case of group-based services selection. Wang et al. [19] presented two approaches for cloud multi-tenant service-based systems (SBS) selection. One was aimed for build-time by clustering services according to a precomputed tenants clusters' requirement. The other is for the runtime to replace a faulty service based on the similarities with the corresponding services in the same cluster. He et al. [20] proposed MSSOptimiser to address the services selection problem for multi-tenant SaaS. The approach models the problem as a constraint optimization problem using a greedy algorithm to find near-optimal solutions efficiently and avoid large computation overhead. However, the previous works deal with the problem of services composition as a multi-objective optimization problem, which is different from our approach that aims to find the single best service using multi-attribute decision-making techniques. Also, different tenants were assumed to have the same importance weights, which can lead to erroneous results.

Only a few existing works take into account the concept of users' varied weights. In particular, Liu et al. [21] proposed an approach for cloud service selection under group decision making by integrating both objective and subjective techniques for criteria and decision-makers' weighting. Statistical variance (SV) and simple additive weighting (SAW) were used to account for correlation in performance evaluation data and the decision-makers' preferences, respectively. As for decision-makers' weights, similarity to the group's decision-based method was combined with Delphi AHP to compute DMs' weights. Decision-makers' weights were partially based on their varying knowledge levels, skills, and expertise to reflect their credibility in the assessment of cloud services. However, in our approach, we assume that the values for cloud service performances are acquired directly by the cloud services providers or third-parties, and users are only to give their requirements. Therefore, parameters like knowledge levels, skills, and expertise have little to no influence in the case of services evaluation for multi-tenants. In addition, we adopt a dynamic weight assigning method controlled by the consensus level to yield a more accepted solution by the whole group of tenants.

1.2 Our contributions

From a practical point of view, there are two fundamental issues in supporting cloud services evaluation for multi-tenants. One is how to effectively aggregate the subjective and uncertain service performances and tenants' requirements while considering their varying services classes and features. The other is how to provide the best solutions with a high level of consensus among the tenants. Given the above problems, and based on previous studies, including our former work [22], the major contributions of this paper are summarized as follows.

(1) To improve the efficiency of the proposed solution, we first employ the Skyline method [23]. Skyline method permits to eliminate the dominated services and only select the dominant and pertinent services according to their QoS performances regardless of users' requirements. Thus, it enables to reduce the search space in case of a large number of services while having low complexity.

(2) To ensure the satisfaction of the requirements of tenants belonging to different service classes, we first compare the alternatives with each tenant's requirements based on their associated service class. This measures the similarity between the tenants' requirements and service performances. Evaluation criteria are expressed using different expression domains, including numeric ones, linguistic ones, and interval numbers. That is because, in practice, some security attributes can be expressed in a deterministic way, such as standard compliance applicability, which takes as value the list of the different standards that the cloud service provider complies with. But because of the subjectivity or uncertainty, other attributes are better expressed using fuzzy logic or interval terms.

(3) Tenants are assigned different weights to reflect their importance. Tenants are initially clustered according to their profiles characterized by different personal, service, and environmental features. Still, tenants in the same cluster may have similar yet different requirements and may belong to different service classes. Therefore, weights are assigned locally (relative to the cluster), objectively based on their closeness to the group decision and subjectively given their services classes. The global weight of the tenant is the product of his local weight and the weight of its cluster.

(4) The final selection is carried under the guidance of a consensus control process. That is, in case the conflict level between the tenants is too high, tenants' weights are adjusted through the execution of a systematic procedure, to reduce or minimize the discrepancy between the collective evaluation results and each individual evaluation.

The rest of the paper is organized as follows. Section 2 presents some preliminary knowledge of the fuzzy set theory and TOPSIS technique. Section 3 discusses the proposed framework. Section 4 presents an illustrative scenario for the application of our work. Section 5 presents a comparative analysis. Section 6 concludes the paper.

2 Preliminaries

This section presents the main definitions related to some of the attributes' formats used in the evaluation, in particular: fuzzy numbers and interval numbers. Also, because the proposed framework is based on the TOPSIS method, we briefly introduce the steps of the TOPSIS method.

2.1 Fuzzy numbers

Fuzzy set theory was proposed by Zadeh [24] to represent the membership degree of an object with respect to a specific class. The notion of fuzzy numbers is formally expressed as follows.

Definition 1 Let U be the universe of discourse, a fuzzy subset Ã of U is defined by its membership function $\mu_A(x)$, where: $A = \{\ (x, \mu_A(x)) | x \in U\}$, and $\mu_A(x) : U \to [0, 1]$.

Definition 2 The triangular fuzzy membership function, which is broadly used to support fuzzy ranking in MCDM models, is defined as $\check{T} = (a, b, c)$, where $a < b < c$,

$$\mu_{\tilde{T}}(x) = \begin{cases} \dfrac{x-a}{b-a} & if\ a \le x \le b \\ \dfrac{c-x}{c-b} & if\ b \le x \le c \\ 0 & \text{otherwise} \end{cases}$$

Definition 3 Let $\tilde{A} = (a_1, a_2, a_3)$ and $\tilde{B} = (b_1, b_2, b_3)$ be two triangular fuzzy numbers, then:

(1)

$$\tilde{A} + \tilde{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$$

(2)

$$\tilde{A} - \tilde{B} = (a_1 - b_3, a_2 - b_2, a_3 - b_1)$$

(3)

$$\tilde{A} \times \tilde{B} = (a_1 b_1, a_2 b_2, a_3 b_3)$$

(4)

$$\tilde{A} / \tilde{B} = (a_1/b_3, a_2/b_2, a_3/b_1)$$

(5) Euclidean distance:

$$d(\tilde{A}, \tilde{B}) = \sqrt{\frac{1}{3}\left[(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2\right]}$$

2.2 Interval numbers

Under many conditions, it is difficult to exactly quantify an attribute value and is more suitable to represent the degree of certainty by an interval. For example, the meantime of incident recovery attribute can be expressed using exact values like 80 hours or interval numbers like [80, 120]. The basic operations on interval numbers are described below.

Definition 4 Given two nonnegative interval numbers $a = [a^l, a^u], b = [b^l, b^u]$ and a positive real number $\lambda \ge 0$.

(1)

$$a = b \text{ if and only if } a^l = b^l \text{ and } a^u = b^u$$

(2)

$$a + b = [a^l + b^l, a^u + b^u]$$

(3)

$$\lambda a = [\lambda a^l, \lambda a^u]$$

(4) Distance: $d(a,b) = \sqrt{\frac{1}{2}((b^l - a^l)^2 + (b^u - a^u)^2)}$

Minimum function [25]

- If $a \cap b = \emptyset$ and $a^u \le b^l$ then $\min\{a,b\} = a$
- If $a = b$ then $\min\{a,b\} = \{a,b\}$
- If $a^l \le b^l \le b^u \le a^u$ if $b^l - a^l \ge (a^u - b^u)$ then $\min\{a,b\} = a$, else $\min\{a,b\} = b$
- If $a^l < b^l < a^u < b^u$ if $b^l - a^l \ge (b^u - a^u)$ then $\min\{a,b\} = a$, else $\min\{a,b\} = b$.

2.3 TOPSIS technique

TOPSIS (Techniques for Order Preference by Similarity to Ideal Solution) [26] is a ranking technique based on the distance measure of an alternative from the ideal solution. The method accounts for both the closeness distance from the positive ideal solution (PIS) representing the best alternative and the farthest distance from the negative ideal solution (NIS) representing the worst choice. TOPSIS was chosen as it best reflects the risk attitudes of decision-makers and alternatives. The smaller the distance measure from PIS, the higher the alternative preference to profit, whereas the bigger the distance measure from NIS, the higher the alternative preference to avoid risk [27]. This approach is suitable for a security-driven evaluation of cloud services as a risk avoider strategy, which seeks to select the alternatives that best match all tenants' requirements while at the same time avoiding as much risk as possible. The traditional procedure to TOPSIS [26] is as follows.

Step 1 Define the decision matrix. Let X be the decision matrix denoting the performance of each alternative A_i, i = 1, 2, ..., m with respect to criteria C_j, j = 1, 2, ..., n

$$X = \begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_m \end{array} & \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \end{array} \quad (1)$$

Step 2: Normalize the decision matrix. Let $R = (r_{ij})_{m \times n}$ be the normalized decision matrix, where:value in the evaluation matrix

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}, \text{ for benefit attributes} \quad (2)$$

$$r_{ij} = 1 - \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}, \text{ for cost attributes.}$$

Step 3: Compute the weighted normalized decision matrix. Let (w_j) is the weight of the criteria j indicating its relative importance to the decision-maker, and $\sum_{j=1}^{n} w_j = 1$.

$$Y = (y_{ij})_{m \times n} = w_j * r_{ij} \quad (3)$$

Step 4: Determine the positive (PIS) $A^+$and the negative (NIS) A^- ideal solution.

$$A^{+} = \{y_1^{+}, y_2^{+}, \ldots, y_n^{+}\} \text{ and } A^{-} = \{y_1^{-}, y_2^{-}, \ldots, y_n^{-}\},$$
$$y_j^{+} = \max_i (y_{ij}),\ y_j^{-} = \min_i (y_{ij}) \quad (4)$$

Step 5: Calculate the distance from positive and negative ideal solutions.

$$S_i^{+} = \sqrt{\sum\nolimits_j^n \left(y_{ij} - y_j^{+}\right)^2} \quad (5)$$

$$S_i^{-} = \sqrt{\sum\nolimits_j^n \left(y_{ij} - y_j^{-}\right)^2}$$

Step 6: Determine the relative closeness to the ideal solution

$$C_i = \frac{S_i^{-}}{S_i^{-} + S_i^{+}} \quad (6)$$

Step 7: Rank the alternatives according to the closeness indexC_i, the higher the value, the better.

3 The proposed method

We consider cloud services evaluation for a large group of tenants as a complex multi-attribute large-group decision-making (CMALGDM) problem. The proposed framework consists of three main phases. The first phase defines the problem structure in terms of alternatives, evaluation criteria, and tenants' requirements. The second phase is the aggregation phase in which the evaluation matrices, criteria weights, and tenants' weights are computed. To efficiently respond to a large group of tenants' requests in a timely manner, we first employ the Skyline method to reduce the search space by removing the dominated web services regardless of users' requirements. Next, to enhance the accuracy of the aggregation given the large number of users, the k-means clustering algorithm is applied to classify users into more homogeneous groups according to their different features. In each cluster, the similarities between the tenants are maximized, and conflicts are minimized. The tenant's weights are determined based on their local weights relative to their corresponding cluster and that cluster's weight. The third and last phase checks the consensus degree and recommends the final ranking of the alternatives. The consensus process serves as a guide for adjusting the weights of tenants automatically and dynamically to achieve a high level of agreement.

The overall process, shown in Fig. 1, can be summarized as follows:

1. Cluster the tenants using the k-means algorithm based on their profiles, including personal, services, and environmental features. The output is k clusters;
2. Apply Skyline method to reduce the search space by eliminating the dominated services regardless of users' requirements;
3. Compute the evaluation matrices for each tenant to determine the similarities between tenants' requirements and services performances supporting multiple QoS-classes;
4. Normalize the evaluation matrices;
5. Determine the weights of the criteria for each tenant based on AHP as a subjective weighting technique and entropy technique as an objective weighting method;
6. Compute the weighted normalized evaluation matrices;
7. Compute the tenants' weights;
   a. In each cluster, the weight of the tenant is calculated objectively based on its closeness to the cluster decision using the TOPSIS method and subjectively depending on its service class;
   b. Compute each cluster weight based on its closeness to the overall group decision including all other clusters using once again the TOPSIS method;
   c. Obtain the global weight of each tenant by combining its weight with its corresponding cluster weight;
8. Finally, the weights of the tenants are integrated with their evaluation matrices and aggregated to obtain the collective evaluation matrix;
9. The ranking is further guided by the consensus process. If the consensus is above a predefined threshold, the selection process is performed or else readapt the weights of the tenants to converge to a higher level of agreement.

3.1 Problem Definition

Problem definition involves identifying the evaluation target and available alternatives, the tenants' profiles, and the evaluation criteria. The aim is to select the single best service that fulfills tenants' diverse requirements and satisfy their negotiated service level.

In the security evaluation process, criteria specification is a critical step. There is still no standard framework for cloud security evaluation criteria. However, general security standards and some specific cloud security frameworks such as CSA Cloud Control Matrix (CCM) [15], are being leveraged as evaluation criteria for security-based cloud services evaluation. The CSA has additionally developed the CAIQ initiative as a complement to the CCM framework, providing a set of questions that act as requirements to help consumers in assessing the compliance of cloud service providers to the CCM. The answers to the questionnaire are made available in STAR repository [28].

Security requirements and preferences tend to be subjective, imprecise, and uncertain, generally expressed in

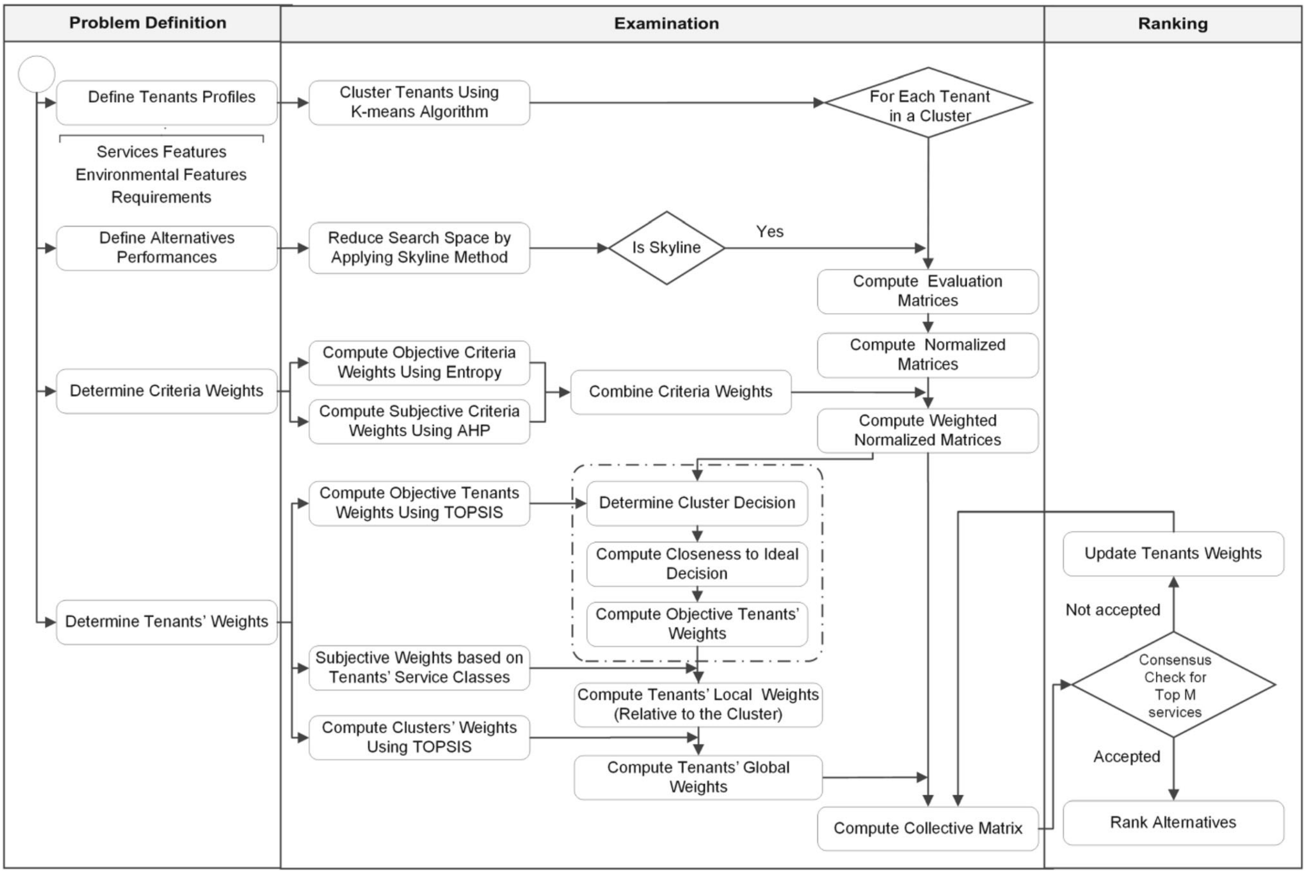
Problem Definition
Examination
Ranking
Define Tenants Profiles
Services Features
Environmental Features
Requirements
Define Alternatives Performances
Determine Criteria Weights
Determine Tenants' Weights
Cluster Tenants Using K-means Algorithm
For Each Tenant in a Cluster
Reduce Search Space by Applying Skyline Method
Is Skyline
Yes
Compute Evaluation Matrices
Compute Normalized Matrices
Compute Objective Criteria Weights Using Entropy
Compute Subjective Criteria Weights Using AHP
Combine Criteria Weights
Compute Weighted Normalized Matrices
Compute Objective Tenants Weights Using TOPSIS
Determine Cluster Decision
Compute Closeness to Ideal Decision
Compute Objective Tenants' Weights
Subjective Weights based on Tenants' Service Classes
Compute Tenants' Local Weights (Relative to the Cluster)
Compute Clusters' Weights Using TOPSIS
Compute Tenants' Global Weights
Compute Collective Matrix
Update Tenants Weights
Not accepted
Consensus Check for Top M services
Accepted
Rank Alternatives

Fig. 1 Multi-tenant services evaluation framework

natural language rather than exact numbers. To account for this heterogeneity, criteria values are modeled using different representation formats, namely: deterministic values, linguistic assessments, fuzzy numbers, and interval data. Both the ratings of alternatives, as well as the requirements, are assessed using these different types of data. For example, the meantime of incident recovery can be expressed using exact values like 80 hours or interval numbers like [80, 120]. Another example is the attribute of user authentication and identity assurance level, which can be described using a number denoting the level of assurance from a scale of 1 to 4, for instance, or using linguistic terms like poor, medium, or high.

The performance of cloud service providers can differ given the different types of SLAs. Current service providers generally offer up to four SLA levels (i.e., silver, bronze, gold, and platinum) or different service packages (i.e., free, professional, enterprise, unlimited). Each service class corresponds to a given quality of service with several different functionalities. This also denotes the level of security and privacy that can be achieved using the various options provided by the cloud provider.

The evaluation problem can be formally defined as follows: let $A_i(\mathrm{i} = 1, 2, \ldots, \mathrm{m})$ be the set of alternatives, $C_j(\mathrm{j} = 1, 2, \ldots, \mathrm{n})$ the criteria, $S_p(\mathrm{p} = 1, 2, \ldots, \mathrm{Q})$ service SLA classes, $T_k(\mathrm{k} = 1, 2, \ldots, \mathrm{V})$ be the tenants, $G_g(1, 2, \ldots, h)$ the clusters of tenants, $rq_k\left(rq_1^k, rq_2^k, \ldots, rq_n^k\right)$ the tenant's requirement vector as per the criteria, $w_k\left(w_1^k, w_2^k, \ldots, w_n^k\right)$, the criteria weight vector provided by each tenant, where $\sum_{\mathrm{j}=1}^{n} \mathrm{w_j} = 1$, and λ_k is the weight of the tenant T_k where $\sum_{\mathrm{k}=1}^{V} \lambda_k = 1$.

3.2 Examination

3.2.1 Step 1: Cluster the tenants based on their profiles

To increase the level of satisfaction of the tenants, we first cluster them based on their profiles. Tenants profiles include their personal features, service features, or environmental features [29]. Compared with the existing researches, the approach in this paper proposes to integrate not only the tenants' requirements but also other services and environmental influence factors. Combining these features will improve the computation method of user features similarity, thus minimize conflicts.

Tenants' personal features can comprise their industry background, sector, personal requirements (e.g., service function, cost, duration, availability, response time, and regulatory policies), and preferences (e.g. cost more important than response time). Based on these requirements, a tenant can negotiate with the service provider to customize the multi-tenancy service through service level agreement (SLA). Service providers support multiple SLA classes (e.g., silver, bronze, and gold), which depend on how much a customer is willing to pay. Consequently, tenants can also be categorized based on their SLA service classes. Environmental features can be characterized by the tenants' location. For example, when the tenants using the application are geographically distributed, it might be better to cluster them based on their location. Thus, resources can be allocated from a resource pool close to the tenant.

Clustering the tenants will help in determining their weights, which plays an important factor in the final results of the evaluation. The clustering is performed using the K-means algorithm (see Algorithm 1). This algorithm is widely used for clustering because of its computational simplicity. The result is h cluster $(h \geq 2)$ with V_g tenants in each cluster G^g and $\sum_{g=1}^{h} V_g = V$. Tenants are denoted by $T^{gk}(\mathrm{g} = 1, 2, \ldots, \mathrm{h}; \mathrm{k} = 1, 2, \ldots, \mathrm{V})$.For clarity, the cluster index g is omitted in the steps related to computing the evaluation matrices, criteria weights, and weighted normalized evaluation matrices, since these steps do not depend on which cluster the tenant belongs to. The cluster's index will be reintroduced when necessary (step 7).

The selection process can also be performed per group of tenants (i.e., clusters) if deemed unnecessary to select the single best service for all the tenants. In this case, it is a simple group multi-attribute decision-making problem (GDM). However, if it is necessary for all the tenants in different clusters to select the same services, it becomes a complex multi-attribute large-group decision-making problem (CMALGDM). In the latter, tenants of different clusters have an influence on the global decision making, and thus their weights are not only computed with respect to the corresponding cluster but also to the other clusters.

Algorithm 1 K-means clustering algorithm

```
Set $\mu_1, \mu_2, \ldots, \mu_k$ to be randomly selected inputs from $x_1, x_2, \ldots, x_n$
For each cluster $C_j$
Repeat
Assign $x_i$ to the nearest center $\mu_j$ based on
    $\arg min_j \left|\left|x_i - \mu_i\right|\right|^2$
Regenerate cluster centers
$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
Until convergence
End.
```

3.2.2 Step 2: Apply Skyline method to reduce search space

Skyline [23] method is a basic MCDM solution that permits to extract the subclass of dominant services and eliminate the dominated ones regardless of any user's requirements. This is because the optimal solution is necessarily within the dominant services [30]. Skyline algorithm is based on the relation of dominance (see Definition 5), which has very low complexity. This makes it suitable as an initial step, but the number of dominant services can still be important, thus the need for a more accurate MCDM solution to rank the remaining services based on users' profiles.

Definition 5 [23]. Given a set of functionally similar services $S = \{s_1, s_2, \ldots, s_m\}$ and a set of QoS parameter $Q = \{q, q_2, \ldots, q_n\}$, we say that s_i dominates s_j $(s_j \prec s_i)$

If

$$\forall q_k \in Q, \quad k \in \{1, 2, \ldots, n\} \begin{cases} q_k(s_i) \geq q_k(s_j) & \text{for benefit criteria} \\ q_k(s_i) \leq q_k(s_j) & \text{for cost criteria} \end{cases}$$

And

$$\exists q_k \in Q, \quad k \in \{1, 2, \ldots, n\} \begin{cases} q_k(s_i) > q_k(s_j) & \text{for benefit criteria} \\ q_k(s_i) < q_k(s_j) & \text{for cost criteria} \end{cases}$$

That is to say that a dominant service is better or equal to another service for all QoS parameters, and strictly better at least for one QoS parameter. As we consider multiple QoS classes, for a service to be excluded it needs to be dominated by other services in all the predefined services classes.

3.2.3 Step 3 Compute the evaluation matrix for each tenant

Let X^pdefines the performance of the alternatives according to class p. We assume that the alternative performances can be obtained from experts and third parties and hence, are not influenced by tenants' subjective and maybe untrusted opinions.

$$X^p = \begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_m \end{array} & \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix} \end{array} \tag{7}$$

In most approaches, the decision matrix denoting the performance of the alternatives or the preferences of the decision-makers is directly used in the evaluation. However, we use a different approach, whereby we first compare the alternatives performances to each tenant's requirements corresponding to the same SLA class, as shown in Eq. (8). This enables us to support the selection of services for a group of tenants that may belong to different services classes. The result at this stage is an evaluation matrix E^k for each tenant$k$ depicting the satisfaction of tenants' requirements against alternatives performances as per the negotiated SLA class. The tenants' requirements $RQ_k\left(rq_1^k, rq_2^k, \ldots, rq_n^k\right)$are also expressed using different types of data (i.e., deterministic values, linguistic assessments, fuzzy numbers, and interval data).

Note for a class-p, the comparison considers only the performances of services pertaining to class p. For simplicity, instead of including the superscript p in all of the following equations to denote that the calculations are performed on tenants and alternatives pertaining to the same class, we omit the indices with the assumption being still valid.

$$E^k = \begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_m \end{array} & \begin{bmatrix} e_{11}^k & e_{12}^k & \ldots & e_{1n}^k \\ e_{21}^k & e_{22}^k & \ldots & e_{2n}^k \\ \vdots & \vdots & \vdots & \vdots \\ e_{m1}^k & e_{m2}^k & \ldots & e_{mn}^k \end{bmatrix} \end{array}, \; e_{ij}^k = \mathrm{x}_{ij} \oslash rq_j^k \tag{8}$$

The value of $\mathrm{x}_{ij} ø rq_j^k$ is computed as follows.

For deterministic values: Numeric:

$$\mathrm{x}_{ij} ø rq_j^k = \begin{cases} \dfrac{\mathrm{x}_{ij}}{rq_j^k} & if rq_j^k \neq 0 \\ x_{ij} & otherwise \end{cases} \tag{9}$$

Boolean:

$$\mathrm{x}_{ij} ø rq_j^k = \begin{cases} x_{ij} & if rq_j^k = 1 \\ 1 & otherwise \end{cases} \tag{10}$$

Set:

$$\mathrm{x}_{ij} ø rq_j^k = \begin{cases} \dfrac{\left|x_{ij} \cap rq_j^k\right|}{\left|rq_j^k\right|} & if rq_j^k \neq \emptyset \\ \left|\mathrm{x}_{ij}\right| & otherwise \end{cases} \tag{11}$$

For fuzzy triangular numbers:

$$e_{ij}^k = \left(e_{ij}^{kL}, e_{ij}^{kM}, e_{ij}^{kU}\right) = \mathrm{x}_{ij} ø rq_j^k$$

It is calculated as per Eq. (4) in Definition 3.

For Linguistic values:

Linguistic terms can be transferred into triangle fuzzy numbers (TFNs) using Table 1.

For Interval values:

$$\mathrm{x}_{ij} = \left[\mathrm{x}_{ij}^l, \mathrm{x}_{ij}^u\right], rq_j^k = \left[\mathrm{rq}_j^{kl}, \mathrm{rq}_j^{ku}\right]$$

Table 1 Linguistic variables and their corresponding TFNs

Linguistic variables	Alternatives performances
The best	(0.8, 0.9, 1)
Better	(0.7, 0.8, 0.9)
Very good	(0.6, 0.7, 0.8)
Good	(0.5,0.6, 0.7)
Normal	(0.4,0.5,0.6)
Bad	(0.3,0.4,0.5)
Very bad	(0.2,0.3,0.4)
Worse	(0.1,0.2,0.3)
The worst	(0,0.1,0.2)

$$\mathrm{e}_{ij}^k = \left[\mathrm{e}_{ij}^{kl}, \mathrm{e}_{ij}^{ku}\right] = \left[\frac{\mathrm{x}_{ij}^l}{\mathrm{rq}_j^{kl}}, \frac{\mathrm{x}_{ij}^u}{\mathrm{rq}_j^{ku}}\right] \tag{12}$$

3.2.4 Step 4. Normalize the evaluation matrix

To ensure comparability of criteria given their different types (i.e., cost and benefit) and dimensions (i.e., time scale, space scale, etc.), we use the following equations [25] to normalize each value in the evaluation matrix.

$$R^k = \begin{array}{c c} & \begin{array}{cccc} C_1 & C_2 & \ldots & C_n \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_m \end{array} & \begin{bmatrix} {r_{11}}^k & {r_{12}}^k & \ldots & {r_{1n}}^k \\ {r_{21}}^k & {r_{22}}^k & \ldots & {r_{2n}}^k \\ \vdots & \vdots & \vdots & \vdots \\ {r_{m1}}^k & {r_{m2}}^k & \ldots & {r_{mn}}^k \end{bmatrix} \end{array}, \text{ where}$$

where

For deterministic values

$$r_{ij}^k = \begin{cases} \dfrac{e_{ij}^k}{\max_j e_{ij}^k} & \text{for benefit attributes} \\ \dfrac{\min_j e_{ij}^k}{e_{ij}^k} & \text{for cost attributes} \end{cases} \tag{13}$$

For fuzzy triangular numbers:

$$r_{ij}^k = \left(r_{ij}^{kL}, r_{ij}^{kM}, r_{ij}^{kU}\right) = \begin{cases} r_{ij}^{kL} = \dfrac{e_{ij}^{kL}}{\max_j e_{ij}^{kU}}, r_{ij}^{kM} = \dfrac{e_{ij}^{kM}}{\max_j e_{ij}^{kU}}, r_{ij}^{kU} = \dfrac{e_{ij}^{kU}}{\max_j e_{ij}^{kU}} & \text{benefit criteria} \\ r_{ij}^{kL} = \dfrac{\min_j e_{ij}^{kL}}{e_{ij}^{kU}}, r_{ij}^{kM} = \dfrac{\min_j e_{ij}^{kL}}{e_{ij}^{kM}}, r_{ij}^{kU} = \dfrac{\min_j e_{ij}^{kL}}{e_{ij}^{kU}} & \text{costcriteria} \end{cases}$$

For Interval values:

$$r_{ij}^k = \left[\mathrm{r}_{ij}^{kl}, \mathrm{r}_{ij}^{ku}\right] = \begin{cases} r_{ij}^{kl} = \dfrac{e_{ij}^{kl}}{max_j e_{ij}^{ku}}, r_{ij}^{ku} = \dfrac{e_{ij}^{ku}}{max_j e_{ij}^{ku}} & benefit criteria \\ r_{ij}^{kl} = \dfrac{min_j e_{ij}^{kl}}{e_{ij}^{ku}}, r_{ij}^{ku} = \dfrac{min_j e_{ij}^{kl}}{e_{ij}^{kl}} & cost criteria \end{cases} \tag{14}$$

3.2.5 Step 5. Determine criteria weights

The weight of criteria w_j^k (where $\sum_{j=1}^{n} w_j^k = 1$) with respect to each tenant are computed using subjective and objective methods to obtain more accurate and less sensitive results to users' preferences or unreasonable criteria prioritization.

3.2.5.1 Step 5.1. Compute subjective criteria weights AHP [31] pairwise comparison approach can be utilized for the assignments of subjective weights $\left(w_j^{sk}\right)$ to the criteria reflecting their degree of importance in view of a particular tenant. That is, each pair of criteria is compared from a scale of 1 (equal) to 9 (extremely important), and the weight of a criterion is obtained from the eigenvector denoting its importance to a particular tenant.

3.2.5.2 Step 5.2. Compute objective criteria weights The entropy method [32] uses the maximum entropy theory proposed by Shannon [33] to provide the objective weighting of evaluation criteria. It determines the criterion's weight based on the information transmitted by that criterion. That is, if a particular criterion has similar values for all the alternatives, then this criterion has little importance in the decision-making. In contrast, the criterion that alternatives are most dissimilar on should have the highest importance weight since it transmits more information and helps to differentiate between the different alternatives. Integrating objective weights with subjective weights helps in adjusting the weights to make them more reliable.

The projected outcomes of a criterion C_j, P_{ij} is defined as:

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \tag{15}$$

The entropy is calculated as follows:

$$ET_j = -\left(\frac{1}{\ln m}\right) \sum_{i=1}^{m} P_{ij} \ln P_{ij} \tag{16}$$

The degree of diversification of the information provided by the criterion j is

$$d_j = 1 - ET_j \tag{17}$$

The entropy weight is then:

$$w_j^o = \frac{d_j}{\sum_{j=1}^{n} dj} \tag{18}$$

3.2.5.3 Step 5.3. Compute the final combined criteria weights $w_j^k = \alpha w_j^o + \beta w_j^{sk}$,where

$$\alpha + \beta = 1. \tag{19}$$

The coefficient $\alpha and \beta$ can be adjusted based on the specific needs of the decision-makers to reflect the influence of subjective and objective weights on the decision-making.

3.2.6 Step 6. Compute the weighted normalized decision matrix

The weighted normalized decision matrix Y^k is computed using each tenant' individual criteria weight vector $\left(w_j^k\right)$ as follows.

$$Y^k = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array} \\ \begin{bmatrix} {y_{11}}^k & {y_{12}}^k & \dots & {y_{1n}}^k \\ {y_{21}}^k & {y_{22}}^k & \dots & {y_{2n}}^k \\ \vdots & \vdots & \vdots & \vdots \\ {y_{m1}}^k & {y_{m2}}^k & \dots & {y_{mn}}^k \end{bmatrix} \end{array}, \text{where}$$

$$Y^k = \left(y_{ij}^k\right)_{m\times n} = \; w_j^k * r_{ij}$$

where

$$Y^k = \left(y_{ij}^k\right)_{m\times n} = w_j^k * r_{ij} \tag{20}$$

3.2.7 Step 7. Determine the weights of each tenant

In the proposed approach, the cloud services selection is performed per group of tenants; thus, we need to consider the ideal decision pertaining to the overall group. Therefore, tenants' weights constitute an important factor that can change the outcome of the decision-making process. Tenants' weights are computed using both subjective and objective methods. Subjective weights can represent, for example, the importance of the tenant based on his SLA class since tenants from different classes can belong to the same cluster (or other factors deemed necessary by the decision-makers), Whereas the objective weight denotes his closeness to the overall group's decision.

The weights of tenants are determined in two stages: locally relative to the cluster and globally by combining tenant's weight with its corresponding cluster weight. The basic steps of the procedure can be summarized as follows: We first determine the local tenant's weight in the cluster based on its closeness to the cluster decision using the TOPSIS method. This weight is combined with subjective weights, as explained earlier. Next, we determine cluster weight with respect to other clusters in an objective way using once again the TOPSIS method. Finally, by combining the local weight of the tenant and its corresponding cluster, we obtain the global weight of the tenant.

Note that the K-means algorithm suffers from the problem of outliers. Assigning weights to the tenants in an objective manner based on their closeness to group decision using the TOPSIS method as proposed in [27], will reduce the influence of outliers since the calculation of the weight maximizes the tenant's weight only if it aligns with the group decision and minimizes it otherwise.

At this step, it is important to differentiate between tenants belonging to different clusters. Therefore, we will reintroduce the superscript g for the cluster index, which was omitted before for clarity reasons. Then, the weighted normalized decision matrix is defined as follows.

$$Y^{gk} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array} \\ \begin{bmatrix} {y_{11}}^{gk} & {y_{12}}^{gk} & \dots & {y_{1n}}^{gk} \\ {y_{21}}^{gk} & {y_{22}}^{gk} & \dots & {y_{2n}}^{gk} \\ \vdots & \vdots & \vdots & \vdots \\ {y_{m1}}^{gk} & {y_{m2}}^{gk} & \dots & {y_{mn}}^{gk} \end{bmatrix} \end{array} \tag{21}$$

3.2.7.1 Step 7.1. Compute tenant's local objective weight

To compute the tenant's local objective weight, we follow the approach proposed in [27] using the TOPSIS technique. The idea is to rank the tenants based on their closeness to the cluster ideal decision. That is the closeness distances from the positive ideal solution (PIS) and the negative ideal solution (NIS).

Obtain the group positive (PIS) and negative ideal solution (NIS)

The group's positive ideal solution (PIS) is obtained by averaging all individual decisions, which will be used as a reference for the cluster's ideal decision.

$$Y^{*g} = \left(y_{ij}^{*g}\right)_{m\times n} = \frac{1}{V_g}\sum_{\mathrm{k}=1}^{V_g} y_{ij}^{gk} \tag{22}$$

The negative ideal solution (NIS) represents the maximum separation from the ideal solution divided into two parts the left negative and the right negative ideal solution. It represents the maximum separation from the group's decision. By employing this method, we would prevent outliers from having a big influence on the decision making of the group.

$$\begin{aligned} Y^{-lg} &= \left(y_{ij}^{-lg}\right)_{m\times n} = \min_{1\le k\le V_g}\left(y_{ij}^{gk}\right) \\ Y^{-rg} &= \left(y_{ij}^{-rg}\right)_{m\times n} = \max_{1\le k\le V_g}\left(y_{ij}^{gk}\right) \end{aligned} \tag{23}$$

Calculate the distance from the positive and negative ideal solutions

$$\begin{aligned} S^{+gk} &= \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(y_{ij}^{gk}, y_{ij}^{*g}\right) \\ S^{-lgk} &= \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(y_{ij}^{gk}, y_{ij}^{-lg}\right) \\ S^{-rgk} &= \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(y_{ij}^{gk}, y_{ij}^{-rg}\right) \end{aligned} \tag{24}$$

Calculate the closeness coefficient

$$Cl^{gk} = \frac{S^{-lgk} + S^{-rgk}}{S^{+gk} + S^{-lgk} + S^{-rgk}} \tag{25}$$

The objective weight of each tenant is determined as follows:

$$\lambda_{gk}^{o} = \frac{Cl^{gk}}{\sum_{\mathrm{k}=1}^{V_g} Cl^{gk}} \tag{26}$$

3.2.7.2 Step 7.2. Compute tenant's combined local weight

Objective weights can be combined with subjective weights λ_{gk}^{s} to represent the importance of the tenant based on his SLA class (or other factors deemed necessary by the decision-makers).

$$\lambda_{\mathrm{gk}} = \mathrm{u}\lambda_{gk}^{o} + \mathrm{v}\lambda_{gk}^{s}, \text{ where } u + v = 1, \tag{27}$$

The coefficient *uandv* can be adjusted based on the specific needs of the decision-makers to reflect the influence of subjective and objective features on the final weight.

3.2.7.3 Step 7.3. Determine each cluster's weight

Depending on the current request, if users and tenants of different clusters can select different services, this step can be skipped, and the best services are selected based on the requirements of tenants in each cluster separately. In this case, it is a simple group multi-attribute decision-making problem (GDM). However, if it is deemed necessary for all the tenants in different clusters to select the same services, it becomes a complex multi-attribute large-group decision-making problem (CMALGDM). The difference is that tenants of different clusters have an influence on the global decision making, and thus their weights are not only computed with respect to the corresponding cluster but also to the other clusters.

The cluster weight is also determined based on the distance to the overall group decision using also the TOPSIS method proposed in [27]. The steps are similar to how we previously computed tenant's local objective weight (step 7.1), but while using the overall decision matinspired by the approach proposedrices of each cluster. The decision matrix of a cluster is represented as follows.

$$B^{g} = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array} \\ \begin{bmatrix} {b_{11}}^{g} & {b_{12}}^{g} & \dots & {b_{1n}}^{g} \\ {b_{21}}^{g} & {b_{22}}^{g} & \dots & {b_{2n}}^{g} \\ \vdots & \vdots & \vdots & \vdots \\ {b_{m1}}^{g} & {b_{m2}}^{g} & \dots & {b_{mn}}^{g} \end{bmatrix} \end{array}, \text{ where}$$

where

$$B^g = \left(b_{ij}^g\right)_{m\times n} = \sum_{k=1}^{V_g} y_{ij}^{gk} \tag{28}$$

The group's positive ideal solution considering all the clusters is computed as follows.

$$B^* = \left(b_{ij}^*\right)_{m\times n} = \frac{1}{\mathrm{h}}\sum_{\mathrm{g}=1}^{\mathrm{h}} b_{ij}^g \tag{29}$$

The left negative and right negative ideal solution for all the cluster are:

$$B^{-l} = \left(b_{ij}^{-l}\right)_{m\times n} = \min_{1\le g\le \mathrm{h}}\left(b_{ij}^g\right) \tag{30}$$

$$B^{-r} = \left(b_{ij}^{-r}\right)_{m\times n} = \max_{1\le g\le \mathrm{h}}\left(b_{ij}^g\right)$$

The distances from the positive and negative ideal solution for each cluster are:

$$S^{+g} = \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(b_{ij}^g, b_{ij}^*\right) \tag{31}$$

$$S^{-lg} = \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(b_{ij}^g, b_{ij}^{-l}\right)$$

$$S^{-rg} = \sum_{i=1}^{m}\sum_{j=1}^{n} d\left(b_{ij}^g, b_{ij}^{-r}\right)$$

3.2.8 Calculate the closeness coefficient

$$Cl^g = \frac{S^{-lg} + S^{-rg}}{S^{+g} + S^{-lg} + S^{-rg}} \tag{32}$$

The weight of each cluster is determined as follows:

$$\lambda_g = \frac{Cl^g}{\sum_{\mathrm{g}=1}^{\mathrm{h}} Cl^g} \tag{33}$$

3.2.8.1 Step 7.4. Compute the global weight of each tenant The global tenant weight λ_k is the combination of its weight relative to its cluster and the weight of the cluster itself.

$$\lambda_k = \lambda_{gk} * \lambda_g \tag{34}$$

3.2.9 Step 8. Aggregation of individual evaluation matrices

Now all individual tenant evaluation matrix can be aggregated using each tenant weight to form the group collective evaluation matrix.

$$Y = \sum_{k=1}^{V} \lambda_k Y^{gk} \tag{35}$$

3.3 Ranking

3.3.1 Step 9 Consensus Control

Cloud services evaluation for multiple tenants aims to find the most profitable solution for the whole group of tenants. However, given the different and often conflicting tenants' requirements, the decision-making process may lead to solutions that will not be well accepted by some tenants. Therefore, a consensus reaching process is necessary to monitor the agreement degree and guide the overall decision-making process. Several consensus models have been proposed in the literature, which generally differ in the mechanism adopted to guide the discussion process and the type of consensus measures utilized [34]. The mechanism adopted to guide the discussion refers to the use of a feedback process that provides decision-makers with some advice to modify their preferences. Other methods choose to automatically update the preferences or the weights of the experts to bring them closer to consensus without the need for human intervention [34]. This model is more favorable for our approach enabling the on-demand characteristic of cloud services.

As for the consensus measures, it is mainly calculated based on the distances between experts' preferences or the distances to the collective preference [34]. In this paper, we adopt the latter, i.e., the distance between individual decisions and the collective group decision. If the consensus is at an acceptable level, the final ranking of alternatives is carried out. To have a more granular view on the progress of the consensus process, we measure both the conflict degree and the agreement degree using the weighted sum aggregation operator and the standard of deviation, respectively. The consensus analysis is applied on the top L alternatives. That is because in real-world situations the number of services can be very large, while also having a larger number of tenants. Therefore, there is no need to have a high consensus between all tenants on all available alternatives. The consensus analysis covers the top L alternatives as well as a single selected one.

3.3.1.1 Step 9.1 Measure the degree of consensus Let D_{ki} be the distance between the individual evaluations of each alternative to the overall collective evaluation.

$$D_{k[i]} = \sum_{j=1}^{n} d\left(y_{[i]j}^k, y_{[i]j}\right) \tag{36}$$

where $[i]$ represents the alternative ranked in the *ith* position.

The conflict degree measures the difference between the individual tenants' evaluation matrices Y^k and the group collective evaluation Y on an alternative is

$$Conf_{[i]} = \sum_{k==1}^{V} \lambda_k D_{k[i]} \tag{37}$$

Accordingly, the conflict degree on the top L alternatives can be defined as follows.

$$Conf = \frac{1}{L} \sum_{i=1}^{L} Conf_{[i]} \tag{38}$$

As for the consensus degree, it is calculated based on the deviation between the different tenants' evaluation and the collective evaluation of each alternative.

The standard of deviation is computed as follows.

$$\sigma_i = \sqrt{\frac{1}{V-1}\left(\sum_{k=1}^{V} D_{ki}^2\right)} \tag{39}$$

The consensus degree on each alternative is defined as

$$Con_{\mathrm{i}} = 1 - \sigma_i \tag{40}$$

The overall group consensus on the top L alternatives

$$Cons = \frac{1}{L} \sum_{i=1}^{L} Cons_{[i]} \tag{41}$$

If the consensus degree is acceptable (above a predefined threshold) and the conflict degree is small enough (usually under 0.1), the final ranking and selection of the alternatives are performed. Otherwise, update the weights of the tenants to reach a better consensus level.

3.3.1.2 Step 9.2. Update the weights of the tenants The consensus of the group decision between the tenants is handled in an automatic way such that the tenants' weights are adjusted to reach a high level of agreement without the need for human intervention. Also, it is very time consuming and expensive to have to interact with the tenants to modify their requirements throughout the process, which can take several iterations before finally reaching an acceptable level. On the other hand, the automatic approach doesn't use a feedback mechanism and instead adjusts the weights of the decision-makers. The basic idea is that the weights of the decision-makers with more extreme opinions (i.e., have a larger distance from the group opinion) are reduced to minimize the conflict degree of the group. Algorithm 2 shows the process for weights adjustments inspired by the approach proposed in [35].

Algorithm 2 Tenants weight update

```
1. Measure the consensus on alternative i without the tenant $T_p$
   $Cons_{i\overline{p}} = \sum_{k \in T/\{p\}} \beta_k Y^k$
   Where $\beta_k = \frac{\lambda_k}{\sum_{j \in T/\{p\}} \lambda_k}$
2. Measure the contribution of the pth tenant
   $D_{ip} = Cons_i - Cons_{i\overline{p}}$
   $D_p = \sum_{i=1}^{m} D_{ip}$
   $D_{ip}$ is the contribution of the pth tenant on the ith alternative. $D_p$ is the aggregated contribution of the pth tenant. The higher $D_p$ it is, the stronger the influence that the pth tenant has on the group decision; if $D_p < 0$, it means that the tenant has a negative influence on the consensus reaching process, and thus its weight should be reduced.
4. If the consensus level is smaller than the given threshold, then update the weights of the tenants
   $u_k^{r+1} = \lambda_k^r * (1 + D_k)^\sigma$,   $\lambda_{gk}^{r+1} = \frac{u_k^{r+1}}{\sum_{k=1}^{V} u_k^{r+1}}$
   $\lambda_k^{r+1}$ is the weight of kth expert in the (r+1) iteration.
   Parameter $\sigma$ represents the influence of the tenant's contribution to their weights. The larger the value of $\sigma$, the larger the adjustment degree of the individual weight will be.
5. Recalculate the group solution and consensus level.
```

3.3.2 Step 10. Determine the overall ranking of alternatives

When the consensus level and conflict level are acceptable, we perform the overall assessment of each alternative by measuring its closeness to the ideal solution using the collective evaluation matrix.

The positive ideal solution (PIS) A^+ and negative ideal solution A^- are determined as follows.$A^+ = \left\{y_1^+, y_2^+, \ldots, y_n^+\right\}$ and $A^- = \left\{y_1^-, y_2^-, \ldots, y_n^-\right\}$,

$$y_j^+ = \max_i \left(y_{ij}\right),\ y_j^- = \min_i \left(y_{ij}\right) \tag{42}$$

Then, we calculate the distance of alternatives from PIS

$$S_i^+ = \sum_{j=1}^{n} d\left(y_{ij}, y_j^+\right) \tag{43}$$

$$S_i^- = \sum_{j=1}^{n} d\left(y_{ij}, y_j^-\right)$$

Based on the distances, the relative closeness to the ideal solution is calculated as follows.

$$Cl_i = \frac{S_i^-}{S_i^- + S_i^+} \tag{44}$$

The alternatives are ranked in descending order according to the closeness coefficient Cl_i and the best one is selected.

4 Illustrative example

While the number of cloud tenants, as well as services, tend to be large in the real-word, for simplicity we consider the case of evaluation of six possible cloud services alternatives for five tenants according to five security criteria namely: authentication level (C1), level of uptime (C2), logs retention period (C3), third party authentication support (C4), and certifications and compliances (C5). These criteria are expressed using different format types including real numbers, boolean, interval numbers, and fuzzy numbers. The descriptions and format type of the criteria are listed in Table 2. For the linguistic terms, their corresponding fuzzy numbers are depicted in Table 1 (Section 3). We assume that each service offers two QoS classes, gold and silver. Table 3 outlines the performances of the alternatives according to the services classes. Table 4 presents the requirements of tenants, having the first tenant in gold class and the remaining in the silver class.

Following the steps of the proposed framework, we cluster the tenants based on their profiles (step 1). The characteristics considered in the clustering are tenants' requirements and their service class. The results of applying the k-means algorithm for two clusters (h = 2) are clusters G1 = {T1, T3, T5}, and G2 = {T2, T4}.

We apply the Skyline method to reduce the number of services to retain only the dominant ones, as explained in step 2 of the framework. A1, A2, A3, A4, and A5 are not dominated by any other services. Note that the service A5 is dominated by A2, A3, and A4 in the gold package but is not dominated by any other service in the silver package and thus, is a Skyline service. The alternative A6 is dominated by the services A1, A2, A3, and A4 in all service classes. Therefore, A6 is excluded. The Skyline services (i.e., the dominant services) that are considered for the ranking are A1, A2, A3, A4, A5.

Following the steps (3) to (6) as proposed in the framework, we obtain the integrated criteria weights by combining objective criteria weights using the entropy model and subjective criteria weights using the AHP technique, presented in Table 5. We suppose that the objective weights and subjective weights have equal importance (α = β = 0.5) in computing the integrated criteria weights. Table 6 shows the weighted normalized evaluation matrix for each tenant.

Next, to aggregate the individual evaluation matrices, we need to compute the weights of the tenants, as discussed in step 7. We first calculate the local tenant's weight (relative to its cluster) based on the closeness of the tenant's individual evaluation from its corresponding cluster ideal solution. Table 7 shows the distances of each tenant from its corresponding cluster's positive ideal solution (PIS) shown in column $S*$, and separation from the negative ideal solution (NIS) shown in column S^{-r} (max) and S^{-l} (min), as well as the closeness coefficient Cl, the local weight λ^l, clusters weights, and the combined global weight of each tenant. Table 8 presents the aggregated collective decision matrix of the group of tenants as per step 8.

At the final stage, consensus analysis is performed to check the degree of agreement among the tenants on the alternatives as discussed in step 9. Assume that the threshold of conflict level is at 0.1, and the consensus level is at 0.75. Table 8 depicts the conflict level and consensus level over each of the alternatives. By using Eqs. (38) and (41) we obtain the overall group conflict level (Conf = 0.19) and consensus level (Cons = 0.78) on all the alternatives (L = 5), respectively. Since the conflict level above the predefined threshold (> 0.1), we need to update the weights of tenants as defined in step (9.2) with setting $\sigma = 2$. At each iteration, tenants' weights are calculated based on their contribution to the overall group consensus and based on which the collective evaluation matrix,

Table 2 Criteria used for cloud services evaluation, their description, and format type

Evaluation criteria	Description of criteria	Format type
Authentication level (C1)	The assurance level of the authentication mechanism used for accessing the service	Triangular fuzzy numbers
Level of uptime (C2)	The period of time during which the service was available	Interval
Logs retention period (C3)	The period of time during which logs are available for analysis	Number
Third party authentication support (C4)	The availability of a third-party authentication support	Boolean
Certifications and compliances (C5)	The list of the standards and certifications that the service complies with	Set

Table 3 Performance of alternatives with respect to the criteria and service classes

	C1	C2	C3	C4	C5
Gold					
A1	High (0.5, 0.7, 0.9)	[99.9–100]	356	Yes	{HIPAA, ISO 27001, PCI}
A2	Medium (0.3, 0.5, 0.7)	[99.95–100]	365	No	{HIPAA, ISO 27001, SOC, PCI}
A3	Very high (0.7, 0.9, 1)	[99.99–100]	356	No	{HIPAA, ISO 27001, SOC, PCI}
A4	Very high (0.7, 0.9, 1)	[99.95–100]	365	Yes	{HIPAA, ISO 27001, SOC}
A5	Medium (0.3, 0.5, 0.7)	[99.95–100]	356	No	{ISO 27001}
A6	Medium (0.3, 0.5, 0.7)	[99.9–100]	356	No	{HIPAA, ISO 27001}
Silver					
A1	High (0.5, 0.7, 0.9)	[99.9–100]	230	Yes	{HIPAA, ISO 27001, PCI}
A2	Medium (0.3, 0.5, 0.7)	[99.95–100]	180	No	{HIPAA, ISO 27001, SOC, PCI}
A3	High (0.5, 0.7, 0.9)	[99.95–100]	230	Yes	{HIPAA, ISO 27001, SOC, PCI}
A4	High (0.5, 0.7, 0.9)	[99.9–100]	356	Yes	{HIPAA, ISO 27001, SOC}
A5	Medium (0.3, 0.5, 0.7)	[99.95–100]	356	No	{ISO 27001}
A6	Medium (0.3, 0.5, 0.7)	[99.9–100]	180	No	{HIPAA, ISO 27001}

Table 4 Tenants services classes and requirements with respect to the criteria

	C1	C2	C3	C4	C5
Gold					
T1	Very high (0.7, 0.9, 1)	[99.95–100]	356	Yes	{HIPAA, ISO 27001, SOC, PCI}
Silver					
T2	High (0.5, 0.7, 0.9)	[99.9–100]	230	No	{HIPAA, ISO 27001, PCI}
T3	Medium (0.3, 0.5, 0.7)	[99.95–100]	180	Yes	{ISO 27001}
T4	Very high (0.7, 0.9, 1)	[99.9–100]	230	No	{HIPAA, ISO 27001}
T5	Medium (0.3, 0.5, 0.7)	[99.9–100]	180	Yes	{HIPAA, ISO 27001}

Table 5 Subjective, objective, and integrated weights of the criteria in view of each tenant

	C1	C2	C3	C4	C5
Objective criteria weights	0.031	0	0.028	0.841	0.101
T1					
Subjective	0.23	0.15	0.16	0.18	0.28
Integrated	0.13	0.08	0.09	0.51	0.19
T2					
Subjective	0.17	0.17	0.15	0.24	0.26
Integrated	0.10	0.09	0.09	0.54	0.18
T3					
Subjective	0.36	0.14	0.17	0.12	0.22
Integrated	0.19	0.07	0.10	0.48	0.16
T4					
Subjective	0.28	0.15	0.18	0.16	0.23
Integrated	0.16	0.08	0.10	0.50	0.16
T5					
Subjective	0.24	0.17	0.17	0.15	0.26
Integrated	0.13	0.09	0.10	0.50	0.18

consensus level, and conflict level are recalculated. This procedure terminates after the 11th iteration, where the overall conflict level is under the threshold (0.09). Figure 2 shows the results of the iterative process depicting the overall conflict level and consensus level along with the collective evaluation of each alternative (Table 9). It shows that the conflict level is decreasing throughout the weight updating process dropping from 0.19 To 0.09. The final accepted overall collective evaluation of alternatives and ranking are presented in Table 10, having A4 as the best alternative. Note that the ranking of the best three alternatives from the group is always the same (A4 > A1 > A3) regardless of the changes in tenants' weights in this case.

Figure 3 depicts the changes in tenants' weights. It indicates that the fifth tenant's weight increases while others decrease with the decline of the group conflict level.

5 Comparative analysis

To investigate the effectiveness of the proposed approach for cloud services selection for multiple tenants, we make (1) a quantitative comparison with other established multi-

Table 6 Weighted normalized evaluation matrices

	C1	C2	C3	C4	C5
T1					
A1	(0.05,0.07,0.12)	[0.07583,0.07595]	0.09	0.51	0.14
A2	(0.03,0.05,0.09)	[0.07587,0.07595]	0.09	0.00	0.19
A3	(0.06,0.09,0.13)	[0.07590,0.07595]	0.09	0.00	0.19
A4	(0.06,0.09,0.13)	[0.07587,0.07595]	0.09	0.51	0.14
A5	(0.03,0.05,0.09)	[0.07587,0.07595]	0.09	0.00	0.05
T2					
A1	(0.03,0.06,0.10)	[0.08548,0.08565]	0.06	0.54	0.18
A2	(0.02,0.04,0.08)	[0.08552,0.08565]	0.05	0.54	0.18
A3	(0.03,0.06,0.10)	[0.08552,0.08565]	0.06	0.54	0.18
A4	(0.03,0.06,0.10)	[0.08548,0.08565]	0.09	0.54	0.12
A5	(0.02,0.04,0.08)	[0.08552,0.08565]	0.09	0.54	0.06
T3					
A1	(0.05,0.09,0.19)	[0.06859,0.06870]	0.06	0.48	0.16
A2	(0.03,0.06,0.15)	[0.06863,0.06870]	0.05	0.00	0.16
A3	(0.05,0.09,0.19)	[0.06863,0.06870]	0.06	0.48	0.16
A4	(0.05,0.09,0.19)	[0.06859,0.06870]	0.10	0.48	0.16
A5	(0.03,0.06,0.15)	[0.06863,0.06870]	0.10	0.00	0.16
T4					
A1	(0.06,0.10,0.16)	[0.07579,0.07595]	0.07	0.50	0.16
A2	(0.04,0.07,0.12)	[0.07583,0.07595]	0.05	0.50	0.16
A3	(0.06,0.10,0.16)	[0.07583,0.07595]	0.07	0.50	0.16
A4	(0.06,0.10,0.16)	[0.07579,0.07595]	0.10	0.50	0.16
A5	(0.04,0.07,0.12)	[0.07583,0.07595]	0.10	0.50	0.08
T5					
A1	(0.03,0.06,0.13)	[0.08548,0.08565]	0.06	0.50	0.18
A2	(0.02,0.04,0.10)	[0.08552,0.08565]	0.05	0.00	0.18
A3	(0.03,0.06,0.13)	[0.08552,0.08565]	0.06	0.50	0.18
A4	(0.03,0.06,0.13)	[0.08548,0.08565]	0.10	0.50	0.18
A5	(0.02,0.04,0.10)	[0.08552,0.08565]	0.10	0.00	0.09

criteria decision-making (MCDM) methods based on rank conformance analysis and conflict level analysis, and (2) a qualitative comparison with the state-of-the-art group-based cloud services evaluation methods.

5.1 Comparison with other MCDM methods

5.1.1 Rank conformance analysis

To evaluate the performance of an MCDM method, we can evaluate the conformity of its ranking results with other MCDM methods. To this end, we compare the results obtained from the proposed method with other well-established MCDM methods, namely: Analytical Hierarchy Process (AHP) [31], Weighted Sum Approach (WSA) [21], Grey Relational Analysis (GRA) method [36], and the traditional TOPSIS method [26]. We consider the same data input used in the previous example, from service performances (Table 3), tenants' requirements (Table 4), to criteria weights (Table 5). We obtain the ranks of services using the above-mentioned methods under similar constraints. Figure 4 depicts the comparison results.

We can observe a close degree of similarity between the ranks of the proposed method and the other MCDM methods. As shown in Fig. 4, 60% (i.e. 3/5) of considered methods show full consensus on the top two alternatives (A4 and A3). Whereas, 80% (i.e. 4/5) of the considered methods show agreement on the third-best alternative (A1). There are some differences in results between the proposed method and the other MCDM methods. This is because the calculation of the ideal reference indexes in the proposed method takes into account the requirements of the tenants and their weights, as opposed to other methods that did not consider these factors. In the real world, it is not always reasonable to assume that all tenants have the same weight and importance level. As discussed before, tenants have different requirements and can belong to different service classes. Still, all methods including the proposed method show agreement on the top three alternatives. From the results, it can be concluded that the performance of the proposed method is up to par with other MCDM methods.

5.1.2 Tenants consensus level on alternatives evaluation

In this part, we assess the consensus level between the tenants on alternatives ratings using different MCDM methods (AHP, WSA, GRA, traditional TOPSIS). For this purpose, we consider the same datasets from the previous example. We obtain the ratings of services using different

Table 7 The distance of tenant's individual evaluation from (PIS)/ (NIS), the closeness coefficient, and the local and global tenants' weights

Cluster	Tenant	S*	S^{-l}	S^{-r}	CI	λ local	Cluster weight	λ global
G1	T1	0.6290	0.3569	0.9408	0.6735	0.3344	0.4908	0.1641
	T3	0.4861	0.8393	0.3920	0.7169	0.3559		0.2529
	T5	0.3769	0.2390	0.3859	0.6238	0.3097		0.1747
G2	T2	0.3030	0.3402	0.3021	0.6794	0.4966	0.5092	0.2563
	T4	0.2740	0.3021	0.3040	0.6887	0.5034		0.1520

Table 8 Collective decision matrix of the group of tenants

	C1	C2	C3	C4	C5
A1	(0.04396, 0.07553, 0.13932)	[0.07846, 0.078761]	0.06794	0.50716	0.16747
A2	(0.02637, 0.05395, 0.10836)	[0.07850, 0.07861]	0.05676	0.26423	0.17539
A3	(0.04692, 0.07882, 0.14143)	[0.07851, 0.07861]	0.06794	0.42353	0.17539
A4	(0.04692, 0.07882, 0.14143)	[0.07847, 0.07861]	0.09741	0.50716	0.15209
A5	(0.02637, 0.05395, 0.10836)	[0.07850, 0.07861]	0.09704	0.26423	0.08597

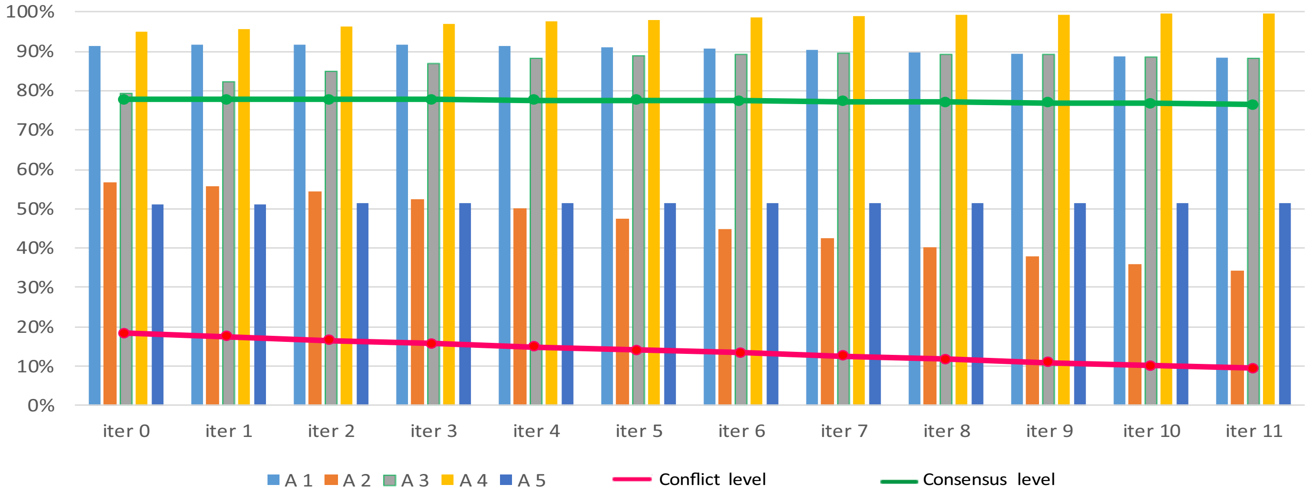

Fig. 2 Group conflict level, consensus level, and the overall collective evaluation of alternatives over the tenants' weights updating process

Table 9 The standard of deviation, consensus, and final ranking of each alternative at the initial iteration

Iteration 0	Standard of deviation σ	Consensus	Conflict level	Alternative rating	Ranking
A1	0.0738	0.92	0.06	0.9146	2
A2	0.3429	0.66	0.30	0.5674	4
A3	0.2708	0.73	0.18	0.7931	3
A4	0.0771	0.92	0.07	0.9490	1
A5	0.3554	0.64	0.31	0.5093	5

Table 10 The standard of deviation, consensus, and final ranking of each alternative at the 11th iteration

Iteration 11	Standard of deviation σ	Consensus	Conflict level	Alternative rating	Ranking
A1	0.0785	0.92	0.04	0.8835	2
A2	0.3487	0.65	0.18	0.3414	5
A3	0.2884	0.71	0.04	0.8823	3

MCDM methods, as well as the proposed method under similar constraints. We assess the consensus level between the tenants on alternatives ratings based on the conflict degree. The conflict degree is calculated based on the distance between the individual tenants' evaluation matrices and the group collective evaluation on each alternative (see Eqs. (36–38)). The smaller the conflict degree the better it is. Fig. 5 depicts the conflict level between tenants on each alternative and the total conflict level on all alternatives.

We can observe from the results a high total conflict degree using other MCDM approaches, namely (0.19) using AHP, (0.22) using WSA, (0.2) using GRA, and (0.39) using traditional TOPSIS method. This is because the evaluation of alternatives using these methods does not consider tenants' requirements or their different weights. The total conflict level is lower using the proposed method, (0.18) before applying the dynamic consensus (at iteration 0) and is at the lowest level (0.1) after applying the dynamic consensus process, compared to other methods. Also, as shown in Fig. 5, the conflict level is the lowest

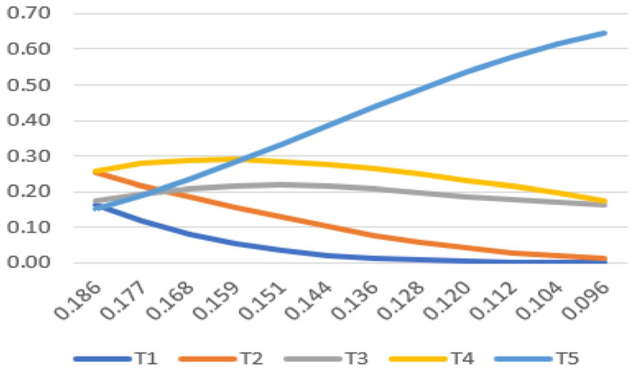

Fig. 3 Group conflict level with the changing of tenants' weights

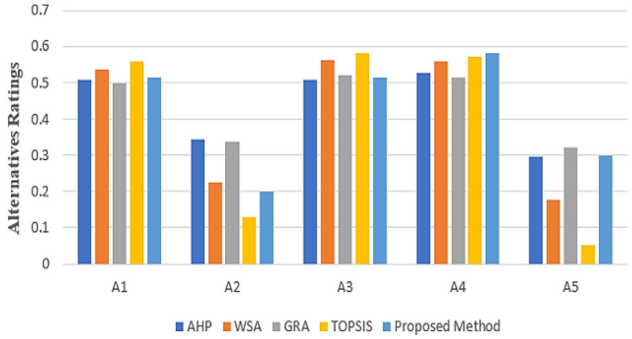

Fig. 4 Comparison of the proposed method with other MCDM methods on cloud services ratings

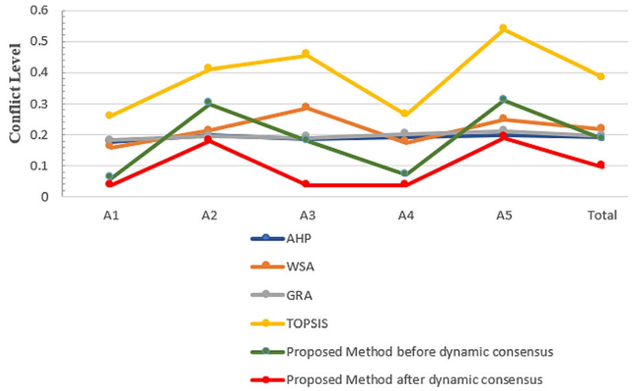

Fig. 5 Conflict level analysis between tenants on alternatives using different methods

using the proposed method, not only in total but also on each alternative. Thus, we can observe an improvement in the consensus degree between tenants on services evaluation using the proposed method. From the results, we can conclude that the proposed method performs significantly better than the individual methods. Further, the results are more reliable and trustworthy as they are based on the consensus of improved results because of the consideration of tenants' requirements and their weights in the evaluation, in addition to performing a dynamic consensus process to reach a final solution with the highest level of agreement.

5.2 Comparison with related work

The proposed framework addresses cloud services selection for a large group of tenants sharing service instances with different, and often conflicting, requirements. A number of studies have considered the case of group-based services selection. Wang et al. [36] proposed a multi-user web service selection method using the Kuhn-Munkres algorithm to select the global optimal solution for multiple users. Wu et al. [37] proposed a time-aware recommendation algorithm for runtime service selection. The Long Short-Term Memory (LSTM) model was used to learn and predict preferences and features before recommending services for users. He et al. [20] addressed the services selection problem for multi-tenant SaaS as a constraint optimization problem using a greedy algorithm. The previous works deal with the problem of services composition as a multi-objective optimization problem, which is different from our work, which aims to find the single best cloud service for multiple tenants.

Among the works considering the problem of single cloud service selection for multiple users, Yadav and Goraya [38] proposed a two-way ranking method (TRCSM). Cloud services were evaluated based on their offered QoS attributes value, while cloud users were evaluated based on three behavioral attributes: service transaction, turnover, and duration. The authors in [39] also proposed a bidirectional cloud services selection framework (MECSM), which evaluates both the cloud provider and consumer in parallel during service mapping. AHP method was used to rank the service providers and the RFM (Recency, Frequency, and Monetary) model was used to rank the consumers. Wang et al. [19] proposed two approaches for cloud multi-tenant service-based systems (SBS) selection. One was aimed for build-time by clustering services according to a precomputed tenants clusters' requirement. The other is for the runtime to replace a faulty service based on the similarities with the corresponding services in the same cluster. The authors aimed at improving the efficiency of services selection for multiple tenants by first clustering the tenants based on their requirements, and then clustering cloud services based on the representative requirements of tenants in each cluster.

In contrast to the previous works, our study is motivated by service selection considering security-related requirements. Ensuring the satisfaction of the security requirements of the tenants is of utmost importance. To achieve this, we compare cloud service alternatives with each tenant's requirements based on their associated service class. This measures the similarity and matching degree between the tenants' requirements and service performances. But in the same time, to improve the efficiency of the approach, we first applied the Skyline method [23] to reduce the search space by removing the dominated web services. Tenants were then clustered based on their features. Another important issue not addressed in the above-mentioned works is tenants' weights. Different tenants were assumed to have the same importance weights and belong to the same service classes. However, in practice tenants are characterized by different personal, service, and environmental features. Even by clustering tenants, tenants in the same cluster may still have similar yet different requirements and may belong to different service classes. Therefore, assigning weights to tenants is important in order to reflect their respective contributions in the decision-making.

Only a few existing works take into account the concept of users' varied weights. In particular, Liu et al. [21] proposed an approach for cloud service selection while considering decision-makers' weights. Decision-makers' weights were based on their similarity to the group's decision and their reliability considering different factors such as knowledge levels, skills, and expertise. However, in our approach, we assume that cloud service performance values are obtained directly by the cloud services providers or third-parties, and users are only to give their requirements. Thus, parameters like knowledge levels, skills, and expertise have little to no influence in our case. Instead, tenants were characterized by different personal, service, and environmental features.

Each of the proposed methods has its advantages and disadvantages. Overall, compared with the existing researches, the approach in this paper focuses on two issues. One is how to effectively aggregate the subjective and uncertain service performances and tenants' requirements while considering their varying services classes and features. The other is how to provide the best solutions with a high level of consensus among the tenants. To improve the efficiency of the proposed solution and reduce computation overhead, the Skyline method [23] is first applied to eliminate the dominated services and only select the dominant and pertinent services. Tenants are clustered based on different personal, service, and environmental features. Weights are then assigned to tenants locally based on their closeness to the cluster decision and their services classes, and globally considering their corresponding cluster's weight. The final ranking of alternatives is carried under the guidance of a consensus process, which adjusts tenants' weights dynamically to yield a more accepted solution by the whole group of tenants.

6 Conclusion

In this paper, we presented a cloud services evaluation framework for multi-tenants supporting different QoS classes. The proposed framework considers three essential aspects: the different and heterogeneous preferences and requirements of the tenants and their respected importance level, the subjective and fuzzy nature associated with the security evaluation process, and the consensus degree of the decision-making results. Future research will explore the support of dynamic evaluation environment in terms of variable services alternatives and cloud tenants, as new tenants may come and existing tenants may leave, also new services may be introduced, and existing services in use may fail. Thus, the framework needs to be flexible and scalable.

References

1. J. Bennett, "Why a no-cloud policy will become extinct," Gartner Available at https://www.gartner.com/smarterwithgartner/cloud-computing-predicts/
2. Badger, L., Patt-corner, R., Voas, J.: NIST cloud computing synopsis and recommendations. Nist Spec. Publ. **800**(146), 81 (2012)
3. Salesforce. [online]. https://.salesforce.com
4. Hawedi, M., Talhi, C., Boucheneb, H.: Multi-tenant intrusion detection system for public cloud (MTIDS), vol. 74. Springer, New York (2018)
5. Chen, X.H.: Complex large-group decision-making methods and application. Press, Beijing, Sci (2009). **in Chinese**
6. Garg, S.K., Versteeg, S., Buyya, R.: A framework for ranking of cloud computing services. Futur. Gener. Comput. Syst. **29**(4), 1012–1023 (2013)
7. Kumar, R.R., Kumari, B., Kumar, C.: CCS-OSSR : A framework based on hybrid MCDM for optimal service selection and ranking of cloud computing services". Cluster Comput. **3**, 1–17 (2020)
8. Ding, S., Yang, S., Zhang, Y., Liang, C., Xia, C.C.C.: Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. Knowledge-Based Syst. **56**, 216–225 (2014)
9. Hammadi, A., Hussain, O.K., Dillon, T., Hussain, F.K.: A framework for SLA management in cloud computing for informed decision making. Cluster Comput. **16**(4), 961–977 (2013)
10. Sundara, M.A.S., Avudaiappan, P.T.: Priority-based prediction mechanism for ranking providers in federated cloud architecture. Cluster Comput. **22**(s4), 9815–9823 (2019)
11. Sun, L., Ma, J., Zhang, Y., Dong, H., Hussain, F.K.: Cloud-FuSeR: fuzzy ontology and MCDM based cloud service selection. Futur. Gener. Comput. Syst. **57**, 42–55 (2016)
12. Sun, L., Dong, H., Hussain, O.K., Hussain, F.K., Chang, E.: Cloud service selection: state-of-the-art and future research directions. J. Netw. Comput. Appl. **45**(October), 134–150 (2014)
13. Alabool, H., Kamil, A., Arshad, N., Alarabiat, D.: Cloud service evaluation method-based multi-criteria decision-making: a systematic literature review. J. Syst. Softw. **139**, 161–188 (2018)
14. Taha, A., Trapero, R., Luna, J., Suri, N.: AHP-based quantitative approach for assessing and comparing cloud security. In: Proc. -

2014 IEEE 13th Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2014, pp. 284–291 (2015)

15. Cloud Controls Matrix. [Online]. https://cloudsecurityalliance.org/group/cloud-controls-matrix/
16. Modic, J., Trapero, R., Taha, A., Luna, J., Stopar, M., Suri, N.: Novel efficient techniques for real-time cloud security assessment. Comput. Secur. **62**, 1–18 (2016)
17. Halabi, T., Bellaiche, M.: Towards quantification and evaluation of security of Cloud Service Providers. J. Inf. Secur. Appl. **33**, 55–65 (2017)
18. Mohammad, H., Ahmad, A., Bin, K.: A novel evaluation framework for improving trust level of Infrastructure as a Service. Cluster Comput. **19**(1), 389–410 (2016)
19. Wang, Y., He, Q., Zhang, X., Ye, D., Yang, Y.: Efficient QoS-aware service recommendation for multi-tenant service-based systems in cloud. IEEE Trans. Serv. Comput. **1374**, 1–14 (2017)
20. He, Q., Han, J., Yang, Y., Grundy, J., Jin, H.: QoS-driven service selection for multi-tenant SaaS. In: Proc. - 2012 IEEE 5th Int. Conf. Cloud Comput. CLOUD 2012, pp. 566–573 (2012).
21. Liu, S., Chan, F.T.S., Ran, W.: Decision making for the selection of cloud vendor: an improved approach under group decision-making with integrated weights and objective/subjective attributes. Expert Syst. Appl. **55**(2016), 37–47 (2016)
22. Maroc, S., Zhang, J.B.: Cloud services security evaluation for multi-tenants. In: 2019 IEEE Int. Conf. Sig. Proces, Com. and Comp. (ICSPCC) (2019).
23. Skoutas, D., Sacharidis, D., Simitsis, A., Kantere, V., Sellis, T.K.: Top-k dominant web services under multi-criteria matching. In: EDBT, ACM Inte. Conf. Procd., vol. 360, pp. 898–909 (2009).
24. Zadeh, L.A.: Fuzzy sets. Inf. Contr. **8**, 338–353 (1965)
25. Aghajani-Bazzazi, A., Osanloo, M., Karimi, B.: Deriving preference order of open pit mines equipment through MADM methods: application of modified VIKOR method. Expert Syst. Appl. **38**(3), 2550–2556 (2011)
26. Yoon, K., Hwang, C.L.: TOPSIS (Technique for order preference by similarity to ideal solution)-A multiple attribute decision making (1980)
27. Yue, Z.: A method for group decision-making based on determining weights of decision-makers using TOPSIS. Appl. Math. Model. **35**(4), 1926–1936 (2011)
28. STAR.: Security trust assurance and risk [online]. https://cloudsecurityalliance.org/.
29. Ma, H., Hu, Z., Yang, L., Song, T.: User feature-aware trustworthiness measurement of cloud services via evidence synthesis for potential users. J. Vis. Lang. Comput. **25**(6), 791–799 (2014)
30. Alrifai, M., Skoutas, D., Risse, T.: Selecting Skyline Services for QoS-Based Web Service Composition, pp. 11–20. ACM, New York (2010)
31. Saaty, T.L.: Decision Making with Dependence and Feedback: The Analytic Network Process, 2nd edn. RWS Publications, Pittsburgh (2001)
32. Xu, X.H., Zhang, L.Y., Wan, Q.F.: A variation coefficient similarity measure and its application in emergency group decision-making. Syst. Eng. Proc. **5**, 119–124 (2012)
33. Shannon, E.: The mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
34. Palomares, I., Estrella, F.J., Martínez, L., Herrera, F.: Consensus under a fuzzy context : taxonomy, analysis framework AFRYCA and experimental case of study. Francisco J. Estrella **20**, 252–271 (2014)
35. Ben-Arieh, D., Chen, Z.: Linguistic-labels aggregation and consensus measure for autocratic decision making using group recommendations. IEEE Trans. Syst. Man, Cybern. Part A Syst Hum. **36**(3), 558–568 (2006)
36. Wang, S., Hsu, C., Liang, Z.: Multi-user web service selection based on multi-QoS prediction. Inf Syst Front. **16**, 143–152 (2014)
37. Wu, X., Fan, Y., Zhang, J. Lin, H., Zhang, J.: QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation. In: Proc. - 2019 IEEE Int. Conf. Serv. Comput. SCC 2019 - Part 2019 IEEE World Congr. Serv., pp. 202–209 (2019).
38. Yadav, N., Goraya, M.S.: Two-way ranking based service mapping in cloud environment. Futur. Gener. Comput. Syst. **81**, 53–66 (2018)
39. Major, N., Goraya, S., Singh, D.: Satisfaction aware QoS-based bidirectional service mapping in cloud environment. Cluster Comput. **7** (2021).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sarah Maroc is a Ph.D. candidate in the College of Computer Science and Technology at Beijing University of Technology, China. She received the Master degree in computer science from the Oran University of Sciences and Technology, Algeria, in 2014. Her research interests include cloud computing, service computing, network, and information security.

Jian Biao Zhang received the Ph.D. degree in computer science from the Northwestern Polytechnic University of Xian, China. He engaged in Postdoctoral research in computer network and security at Beihang University. He is currently a Professor and the vice director of the Department of Information Security at Beijing University of Technology, China. He has participated in many national research projects and published several papers in international journals and conferences. His research interests include trusted computing, information security, network security, and computer network. He is a member of the IEEE.