IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 10 June 2022, accepted 26 June 2022, date of publication 7 July 2022, date of current version 18 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3189172

RESEARCH ARTICLE

An Efficient Hybrid Ranking Method for Cloud Computing Services Based on User Requirements

ZONGBAO MA[1], MOHAMMAD HOSSEIN NEJAT[2], HAMED VAHDAT-NEJAD[3], BEHNAM BARZEGAR[4], AND SAEED FATEHI[5]

[1]School of Information Engineering, Xi'an University, Xi'an 710062, China
[2]Department of Computer Engineering, Islamic Azad University, Sari 4818875666, Iran
[3]Department of Computer Engineering, University of Birjand, Birjand 9717434765, Iran
[4]Department of Computer Engineering, Islamic Azad University, Babol 47471-37381, Iran
[5]Department of Computer Engineering, University College of Allameh Helli chalus, chalus 4661966854, Iran

Corresponding author: Behnam Barzegar (barzegar.behnam@yahoo.com)

ABSTRACT An increase in the number of cloud services makes service selection a challenging issue for cloud users. It is important to determine the best service that can fulfill user requirements. To this end, this paper proposes a hybrid multiple-attribute decision-making (MADM) model. The proposed method considers service measurement index cloud (SMICloud) structure for qualitative attributes of cloud services as well as user requirements based on fuzzy values to consider vague user requirements. Analytical hierarchy process (AHP) and fuzzy logic are used to rank cloud services. Furthermore, a fuzzy Delphi filtering method is proposed to decrease the execution time of ranking cloud services. In experiments, different aspects such as accuracy, execution time, scalability, and sensitivity analysis are investigated. The results confirm that the proposed method outperforms available methods in terms of execution time and scalability. Furthermore, the experiments show that the proposed method has achieved an accuracy of 96%.

INDEX TERMS Cloud computing, quality of service (QoS), Fuzzy Delphi method, cloud user requirement, cloud service ranking.

I. INTRODUCTION

Many companies and organizations have provided various cloud services to clients [1]. These services are classified, according to cloud computing architecture, into three layers namely Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [2]. Clients of the cloud can use services anywhere without any need for specific skills and spending too much money [3]. This has increased tendencies toward cloud computing [4]–[5]. Selection of the most suitable service from a great number of provided services according to quality of service (QoS) [6] attributes and users' requirements has attracted the researchers' attention [7]. Cloud Service Measurement Index (SMI) [8] has several attributes used to evaluate cloud services to compare their performance [9]. Each attribute consists of several sub-attributes as shown in Figure (1) [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

Furthermore, users' requirements for cloud services are mainly related to QoS. They are divided into functional and non-functional groups [10]–[12]. Several qualitative attributes should be considered to select a service among many others; hence, it is a multiple-attribute decision-making (MADM) problem [4], [13]. Various models have been proposed to solve this issue [4], [14]–[16]. However, they suffer from long execution times or low accuracy. We have experimentally demonstrated that they have difficulties in large-scale cloud services. Selecting specific quality attributes to assess cloud services is a challenge. Because each quality attribute affects the ranking results. Inaccurate user requirements must also be considered for ranking systems. In addition, the ranking system should have the least time complexity and high robustness.

This paper proposes a method for ranking cloud services based on clients' requirements in the service level agreement (SLA). Many Quality of Service (QoS) attributes lead to complexity in selecting and ranking cloud services. Due to the

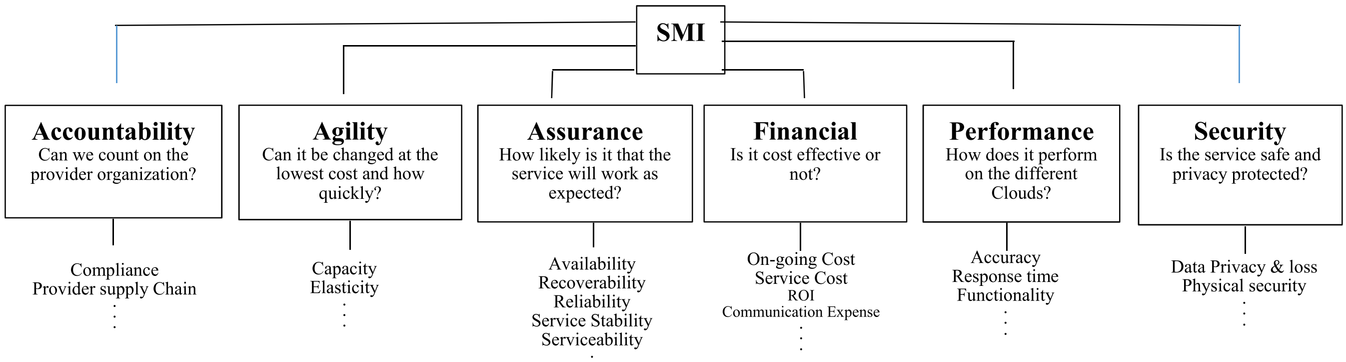

FIGURE 1. **Categories and attributes of service measurement index [17].**

hierarchical structure of SMI, the analytical hierarchy process (AHP) [18] is used to solve the uncertainty in calculations. To this end, the Fuzzy Delphi method (FDM) [19] infers a consensus in experts' opinions in a series of intensive questionnaires together with controlled feedback based on fuzzy techniques [20]. FDM is used with two goals [21]–[23]: Firstly, it uses the feedback from users who previously used cloud services to select the most popular and important qualitative attributes. Secondly, cloud services are ranked faster by removing less valuable qualitative attributes and reducing comparative computations. The main aim of this research is to provide a high-accuracy method with low processing time to select the most appropriate services. We also propose a new pre-processing method, which selects the attributes in the SMI framework and then prepares them for processing and ranking the cloud services by the Fuzzy analytical hierarchy process (FAHP) [15], [24] and AHP. Furthermore, the weights of the quality attributes are defined By Fuzzy user requirements based on FAHP. We use AHP for comparing services according to QoS values. Finally, we aggregate user requirements and service quality values to rank the services. In proposed method over existing works is to provide a ranking method that involves a pre-processing step and refined the number of attributes by FDM. The pre-processing step leads to more scalability and reduces execution time while keeping the accuracy. Furthermore, we use fuzzy set theory [25] to cover cloud user requirements and human thinking. Finally, we implement the proposed approach as well as AHP [14], [26]–[31] and FAHP [5], [16], [32]–[33] as similar methods in MATLAB software. The experimental results show that the proposed approach is more scalable and decreases the execution time with high accuracy. The main contributions of this paper are summarized as follows:

- A hybrid ranking method called Pre-decision Fuzzy Delphi Ranking (PFDR) has been proposed based on user requirements. PFDR uses appropriate attributes by the Delphi method. Also, it uses Fuzzy values to consider vague experts' opinions and imprecise user requirements.
- PFDR emphasizes preprocessing to improve the ranking results by selecting appropriate quality attributes.
- PFDR employs the Fuzzy AHP method to rank cloud services. The ranking results were obtained from imprecise user requirements and quality values.
- A comparison has been applied based on scalability, execution time, accuracy, and sensitivity metrics between PFDR and other existing methods.

The rest of the paper is organized as follows. Section 2 gives a brief overview of relevant literature. The proposed hybrid approach for cloud service ranking is described in Section 3. Section 4 presents a case study to select the best cloud services by using the proposed hybrid approach. Finally, we conclude the paper in Section 5.

II. RELATED WORK

Considering the diversity of cloud service providers, the task of choosing a service with the highest compliance with the users' requirements in the shortest time is facing big challenges. These challenges include a large number of QoS attributes, varied and ambiguous user requirements, and a long processing time for ranking cloud services [5], [34]. Here, we compare our work with the previous research works on the comparison of cloud services and solutions to cloud service ranking. Comparison of cloud services is a new approach (compared to the comparison of web services) that has attracted the attention of researchers. Considering that users' requirements must be considered in the ranking of cloud services, many of the presented methods used the AHP-based method. In [29], by considering a hierarchical structure for some of the users' non-functional requirements, an AHP method is used for weighing each of the QoS attributes. Also in [48], the authors tried to compare some multi-attribute decision-making (MADM) methods. The results show that AHP is the best method. The authors in [30] proposed a hierarchical structure for the functional and non-functional requirements of web services based on AHP to rank the cloud services in the SaaS and IaaS layers. An AHP-based method is presented in [31] for ranking cloud services in all cloud layers. In this structure, due to the consideration of a large number of attributes, the processing time of paired comparison in AHP becomes very long, which is not applicable in a space with a large number of cloud services.

Given the large number of QoS attributes that should be considered in the requirements of cloud users, the researchers used software quality models to provide a comprehensive model from the attributes. The International Standard of ISO/IEC 9126 [45]–[46] has a hierarchical structure that is used for quality evaluation of software products and has 6 quality attributes and 24 quality sub-attributes [28]. [47] Presented an AHP-based model, in which they changed the quality attributes in ISO/IEC 9126 to efficiency, reliability, reusability, availability, and scalability. Tran *et al.* [26], by providing a hierarchical structure, ranked the web services by the AHP method. The proposed methods for evaluation of quality attributes, in which many cloud attributes, such as VM capacity, are not considered. The SMICloud framework is presented to provide a comprehensive hierarchy of the quality attributes of cloud services [39]. Nejat *et al.* produced a fast cloud service ranking method based on encoding the requirements and eliminating inappropriate cloud services. The results show that using pre-processing step can reduce the execution time [49]. The multi-objective service selection (MOSS) was proposed to rank cloud services based on the quality of services and users' feedback [50]. New metrics for quality evaluation of cloud attributes in the SMICloud structure were presented in [14]. The authors used an AHP-based framework to rank cloud services. Given the fact that users' requirements are linguistic, AHP is not able to reflect the users' vague requirements that are linguistic. Researchers used the fuzzy set theory to solve the problems that have vague concepts from users' requirements. [33] used a fuzzy AHP method in the SMIcloud structure to rank the cloud services. In this method, due to the use of users' requirements, a more realistic ranking has been carried out. AHP method, due to numerous paired comparisons, is a time-consuming process and reduces system efficiency in a massive number of services. Therefore, execution time is another challenge in the ranking of cloud services. In [27], a multi-layer hybrid method has been used to reduce the number of paired comparisons and minimize the execution time in the AHP method. In this method, AHP has been used only to weigh the quality attributes, and ranking is done in a separated layer. In [16], the researchers have improved a multi-layer hybrid method for multi-layer fuzzy ranking. The ranking results in this method are different from the ranking results in [14], [33], [39]. Therefore, to solve this problem in [51] MADM methods are combined to rank cloud services accurately. According to the authors' best knowledge, it is obvious that diversity in services' attributes and ambiguity in the user's requirements is the main issue in ranking cloud services. In addition, the main disadvantage of the recent research is the fact that reduced execution time with high accuracy in the ranking of cloud services is an issue that has not been addressed in cloud systems. The sensitivity analysis is another important factor in the ranking system that needs to be addressed. Our proposed method is a new method of ranking cloud services that address these problems.

III. PROPOSED PFDR METHOD

The proposed Pre-decision Fuzzy Delphi Ranking (PFDR) includes four steps displayed in Algorithm 1. In general, at first, the most important qualitative key attributes are selected. At last, cloud services are ranked based on the vague requirements of users and qualitative attributes of services. PFDR inputs are user requirements, expert opinions, and a set of services. The output is the ranked services. All of the steps are depicted in Figure (2).

Algorithm 1 : PFDR Algorithm: Pre-Decision Fuzzy Delphi Ranking

Input: A full-service set, all attributes, all expert opinions, and user requirements.
Output: A Cloud service ranking.
1: Identify qualitative attributes.
2: Refine attributes by FDM.
3: Assign weights to attributes by user requirements by FAHP.
4: Rank cloud services.

In the following, these steps are elaborated:

Step 1- Identifying qualitative attributes: Qualitative attributes are adopted from SMICloud [14]. SMICloud with a hierarchy structure is designed especially for cloud services. These attributes are shown in Table (1).

As illustrated in Table (1), quality attributes are organized hierarchically on three levels. For instance, capacity is a sub-attribute of agility, and agility is a top-level QoS attribute. Similarly, capacity includes CPU, memory, and disk, which are organized in the third level. Step 2- Refining attributes: There is a large number of qualitative attributes introduced in the previous step, which will result in an excess amount of time for processing them. Some of these attributes are not important in certain cloud applications. For example, "installability" is not an important attribute for cloud users. As a result, the qualitative attributes are evaluated through a Fuzzy Delphi Method (FDM). To this end, the expert's opinions regarding these attributes are gathered as Fuzzy linguistic variables including "Extremely important", "Very important", "Definitely important", "Somewhat important", and "Not Important". For simplicity, we use triangular membership functions to define these variables as shown in Table (2).

Algorithm 2 shows the proposed FDM method. It includes collecting experts' opinions based on a linguistic variable, reaching a consensus between them, evaluating the values extracted for each attribute, and defining the importance of each attribute.

The fuzzy collection of expert opinions is first shown as $A_i=(l_i.m_i.u_i)$. If there is a consensus on each attribute (line 9), the fuzzy number is de-fuzzified using the following formula

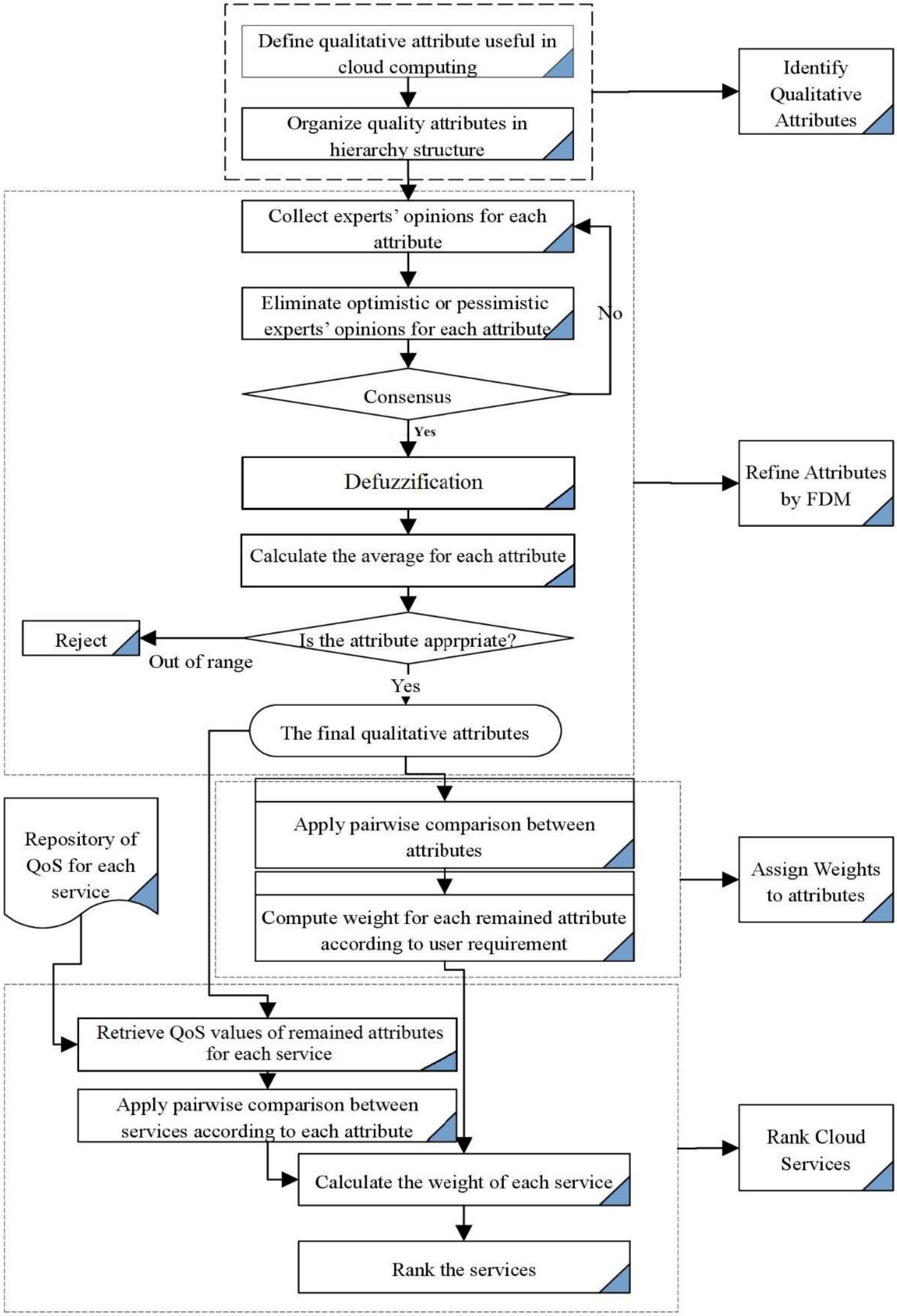

FIGURE 2. PFDR flowchart.

TABLE 1. Hierarchical structure of qualitative attributes [14].

Top Level QoS Groups	First Level Attributes	Second Level Attributes	Value Type
Accountability			Numeric
Agility	Capacity	CPU	Numeric
		Memory	Numeric
		Disk	Numeric
	Elasticity	Time	Range
	Transparency	Time	Number
Assurance	Availability		Range
	Interoperability		Number
	Service Stability	CPU	Numeric
		Memory	Numeric
		Upload Time	Numeric
	Serviceability	Free Support	Boolean
		Type of Support	Unordered set
	Reliability		Numeric
Cost	Service Cost	Storage	Numeric
		VM Cost	Numeric
		Data	Numeric
Performance	Service Response Time	Range	Numeric
		Average Value	Numeric
	Throughput		Numeric
	Accuracy		Numeric
Security			Numeric

TABLE 2. Triangular fuzzy numbers in five linguistic expressions.

Fuzzy Linguistic	Fuzzy Number	Fuzzy Score	Abbreviation
Extremely important	$\tilde{9}$	(7,9,9)	VI
Much Important	$\tilde{7}$	(5,7,9)	I
Important	$\tilde{5}$	(3,5,7)	MI
Somewhat important	$\tilde{3}$	(1,3,5)	U
Not important	$\tilde{1}$	(1,1,1)	VU

in line 10:

$$DF_{ij} = \frac{\left[\left(u_{ij} - l_{ij}\right) + \left(m_{ij} - l_{ij}\right)\right]}{3} + l_{ij}$$

Then, the average of expert opinions is calculated in line 1.

$$attr_{avg}\,(j) = \frac{\sum_{i=1}^{n} DF_{ij}}{n}$$

The experts who have the lowest and highest values of opinions are considered as optimistic and pessimistic on that cloud attribute, respectively. Therefore, their opinions on that attribute are removed (lines 14) and opinions are collected again in line 15. Finally, a desired attribute is either accepted or rejected by comparing it with the threshold (th) within lines 20 to 23. The value of th is calculated, experimentally. The advantage of the PDFDM algorithm is to eliminate optimistic and pessimistic comments (i.e. outliers) due to the unfair competition or advertisement of any cloud service. At last, the hierarchical structure of attributes is pruned, and the remaining attributes are depicted as a pruned hierarchical tree.

Step 3- Assigning weight to the attributes: After the hierarchical structure is prepared for attributes, the weights should be assigned to each attribute according to user requirements. Weight assignments show the importance of one attribute concerning another. Requirements of cloud users are represented by linguistic variables. Due to the fuzzy nature of the language of people, it is better to use fuzzy theory to obtain these requirements. Therefore, the fuzzy AHP (FAHP) method is used to improve decision-making because AHP cannot reflect well human thinking. FAHP is used to calculate the weight of each attribute based on a pairwise comparison matrix of quality attributes. Comparison matrices are used for modeling fuzzy values according to Table (2). Pairwise comparison matrices are created for attributes of the same group from the lowest level. The details of the FAHP method are presented in the next section.

Algorithm 2 : Refining Attributes Algorithm

Inputs: attributes; all attributes; experts: all expert opinions; n: number of experts; th: threshold

```
Function PDFDM(attributes, experts, n, th)
    Collect experts' opinions based on linguistic variables in table 2.
    For each i in attributes do
        Collect experts' opinions for attribute_i
        h (h_l.h_m.h_u) = _{i=1}^{n} max(attr_j);
        l (l_l.l_m.l_u) = _{i=1}^{n} min(attr_j);
        While consensus for attribute_i is not existed do
            if (|h_il − l_iu| > |h_im − l_im|)&&(l_ih < h_il)then
                //The consensus of experts' opinions exists
                DF_ij = [(u_ij−l_ij)+(m_ij−l_ij)]/3 + l_ij
                attr_avg (j) = (∑_{i=1}^{n} DF_ij)/n
            Else
                Sort the value of experts' opinions for attribute_i in S.
                experts=experts-{s_first, s_last}
                Collect experts' opinions for attribute_i
            End if
        end while
    End for
For each i in attributes
    if attr_avg (i) > th then
        Add attr(i) to New_attrib;
    else
        Ignore attr(i);
Next i
End function
```

Output: New_attrib;

Step 4- Ranking cloud services: In this step, cloud services are ranked based on the value of each qualitative attribute in each service and the weights assigned to each attribute in the previous step. To this end, the value of each cloud service from each particular attribute is calculated by the AHP method. Attributes have various types including Boolean, numerical, unordered list, and interval. The comparison of two services s_i and s_j is shown as ${}^{s_i}/_{s_j}$. To compare two services in a particular attribute, we must have the value of $service_i$ (v_i) and $service_j$(v_j) as well as the user's requirement for that attribute (v_r). If the type of an attribute is Boolean, the results are obtained according to Table (1). If v_i, v_j, and v_r are equal, the value of ${}^{s_i}/_{s_j}$ will be 1. If v_i and v_j are equal but v_r has a different value, the value of ${}^{s_i}/_{s_j}$ will be 0. If v_i and v_j are different but v_i and v_r are equal, ${}^{s_i}/_{s_j}$ is w_m. w_m is the real number associated with the fuzzy weight of the attribute, which is assigned by the user in step (1), e.g. 9 for $\tilde{9}$. Finally, if v_i and v_j are different but v_j and v_r are the same, the value of ${}^{s_i}/_{s_j}$ is ${}^{1}/_{w_m}$.

If the attribute is numerical, s_i/s_j is the result of v_i/v_j. If the attribute is the unordered list, then the number of items of $service_i$, which are the same as the value of requirements, are defined and shown as $size(v_i \cap v_r)$. Therefore, the value of s_i/s_j is computed using the following formula:

$$s_i/s_j = \begin{cases} size\,(v_i \cap v_r) & if\ \ v_j \cap v_r = \emptyset \\ \dfrac{size\,(v_i \cap v_r)}{size\,(v_j \cap v_r)} & Otherwise \end{cases} \quad (1)$$

For interval type values, the length of overlap between s_i and user requirement is shown as $len(v_i \cap v_r)$. The value of s_i/s_j

TABLE 3. Results of comparing two services for a Boolean attribute.

v_r	v_i	v_j	${}^{s_i}/_{s_j}$
1	0	0	0
1	1	0	w_m
1	0	1	${}^{1}/_{w_m}$
1	1	1	1
0	0	0	1
0	0	1	${}^{1}/_{w_m}$
0	1	0	w_m
0	1	1	0

is calculated using the following formula:

$$ {}^{s_i}/_{s_j} = \begin{cases} \dfrac{len(v_i \cap v_r)}{len(v_j \cap v_r)}. & if\ v_j \cap v_r \neq \emptyset\ and\ v_i \cap v_r \neq \emptyset \\ 1. & if\ v_j \cap v_r = \emptyset\ and\ v_i \cap v_r = \emptyset \\ w_m. & if\ v_j \cap v_r = \emptyset\ and\ v_i \cap v_r \neq \emptyset \\ \dfrac{1}{w_m}. & if\ v_j \cap v_r \neq \emptyset\ and\ v_i \cap v_r = \emptyset \end{cases} \tag{2} $$

Finally, after the values of matrices for the particular attribute are calculated, the values are gathered in matrix C. It contains n columns and m rows. n is the number of attributes (or sub-attributes in the same group), while m is the number of services. Then, the values of C are normalized using the following formula:

$$ nc_{ij} = \frac{c_{ij}}{\sum_{i=1}^{m} c_{ij}}. \tag{3} $$

where the sum of j^{th} attribute for all services in C is computed using $\sum_{i=1}^{m} c_{ij}$, and c_{ij} presents the value of i^{th} service for the j^{th} attribute (or sub-attribute). The vectors calculated for sub-attributes are aggregated concerning the weights calculated in step 3. Aggregation is repeated hierarchically until top-level attributes are calculated. The weights of the top-level attributes and the values of qualitative attributes of services are considered to obtain the final value for each service, which is represented by $QoS_{n\times 1}$:

$$ \begin{bmatrix} QoS_1 \\ \vdots \\ QoS_m \end{bmatrix} = \begin{bmatrix} nc_{1.1} & \cdots & nc_{1.n} \\ \vdots & \ddots & \vdots \\ nc_{m.1} & \cdots & nc_{m.n} \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \tag{4} $$

Finally, the index of the biggest value of the QoS vector is identified as the most appropriate service.

IV. CASE STUDY: SERVICE RANKING

The evaluation of the proposed PFDR method is performed by three cloud service providers datasets including Amazon EC2 [42], Windows Azure [43], and Rackspace [44]. Firstly, we consider the qualitative attributes in Table (2). For simplicity, all attributes of Table (1) are presented in Table (2) in one level.

We have used the opinions of 20 experts in decision-making. The list of expert opinions on qualitative attributes is presented in Table (3). Each row is a collection of an expert's comments, and each column represents a qualitative attribute.

Fourteen qualitative attributes are selected by using the proposed FDM-based Algorithm. These qualitative attributes are presented in Table (4).

The hierarchical structure of selected qualitative attributes is shown in Figure (3). As can be seen, the top-level attributes are not changed. But some attributes in second and third level are removed. According to aggregation of these experts for 22 attributes, the remaining attributes are included: Accountability, security, capacity, availability, serviceability, cost, response time, and throughput.

The attribute values of the utilized cloud services are shown in Table (5) [14]. User weights are simulated (assigned randomly), and the service ranking is performed, accordingly.

In the proposed method, we get user requirements as the fuzzy number. However, the Fuzzy requirements are different for a different user, since the user can ignore certain attributes. The ignored attributes may have the lowest importance to users. A pairwise comparison matrix (C) is generated using fuzzy numbers. It contains elements to show the importance of one attribute to another in pairs according to user requirements. Each matrix is generated from the lowest level of the same group. For attributes in the same group, we consider pairwise comparison matrix, separately. One of the elements of C is $\tilde{a}_{ij} = \left(l_{ij}.m_{ij}.u_{ij}\right)$, where l_{ij} is the lower bound, u_{ij} is the upper bound, and m_{ij} is the median of upper and lower bounds. In this case, i denotes the attribute number and j denotes the service number. In general, if $\tilde{A}_1 = (l_1, m_1, u_1)$ and $\tilde{A}_2 = (l_2, m_2, u_2)$ are two triangular fuzzy numbers, the fuzzy operations are as follows:

$$ \tilde{A}_1 \oplus \tilde{A}_2 = (l_1 + l_2.m_1 + m_2.u_1 + u_2) \tag{5} $$
$$ \tilde{A}_1 \otimes \tilde{A}_2 = (l_1 \times l_2.m_1 \times m_2.u_1 \times u_2) \tag{6} $$

The FAHP method is previously presented by [5], [15], [24]. The fuzzy score number for each attribute (in each row) is shown by S_i, which is calculated in (7). Sum-up in each row of triangular fuzzy numbers of C is prepared by a vector and all of the vectors are introduced by SM in (10). The Sum of the triangular fuzzy number vectors of SM is shown as SSM in (11). The reverse of SSM is introduced by RSSM, which is presented in (12).

$$ S_i = SM \otimes RSSM \tag{7} $$

where

$$ SM = \begin{bmatrix} \sum_{j=1}^{m} \tilde{a}_{1,j} \\ \sum_{j=1}^{m} \tilde{a}_{2,j} \\ \cdots \\ \sum_{j=1}^{m} \tilde{a}_{n,j} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{m} l_{1j}.\sum_{j=1}^{m} m_{1j}.\sum_{j=1}^{m} u_{1j} \\ \sum_{j=1}^{m} l_{2j}.\sum_{j=1}^{m} m_{1j}.\sum_{j=1}^{m} u_{1j} \\ \cdots \\ \sum_{j=1}^{m} l_{nj}.\sum_{j=1}^{m} m_{nj}.\sum_{j=1}^{m} u_{nj} \end{bmatrix} \tag{8} $$

$$ SSM = \sum_{i=1}^{n}\sum_{j=1}^{m} \tilde{a}_{i.j} = \left(\sum_{i=1}^{n} l_i.\sum_{i=1}^{n} m_i.\sum_{i=1}^{n} u_i\right) \tag{9} $$

$$ RSSM = \left[\sum_{i=1}^{n}\sum_{j=1}^{m} \tilde{a}_{i.j}\right]^{-1} = \left(\frac{1}{\sum_{i=1}^{n} u_i}.\frac{1}{\sum_{i=1}^{n} m_i}.\frac{1}{\sum_{i=1}^{n} l_i}\right) \tag{10} $$

TABLE 4. List of the numbered qualitative attributes.

A_1	Accountability	A_{12}	Stability of Memory
A_2	Security	A_{13}	Reliability
A_3	Capacity of CPU	A_{14}	Cost of storage
A_4	Capacity of Memory	A_{15}	Transparency
A_5	Capacity of Disk	A_{16}	Interoperability
A_6	Elasticity of Time	A_{17}	Cost of VM
A_7	Stability of Upload Time	A_{18}	Cost of Data
A_8	Stability of CPU	A_{19}	Range of Response Time
A_9	Availability	A_{20}	Average of Response Time
A_{10}	Serviceability free support	A_{21}	Throughput
A_{11}	Serviceability Type of Support	A_{22}	Accuracy

TABLE 5. List of comments by 20 experts for 22 qualitative attributes.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22
Expert 1	VI	VI	I	I	I	MI	UI	NU	I	I	VI	UI	MI	I	UI	NU	I	I	I	VI	I	MI
Expert 2	VI	I	VI	VI	VI	I	UI	NU	VI	I	VI	UI	MI	I	NU	MI	I	I	I	VI	I	MI
Expert 3	I	VI	I	VI	I	I	UI	NU	VI	I	VI	MI	MI	VI	UI	MI	VI	VI	VI	I	VI	MI
Expert 4	VI	I	I	I	I	MI	UI	UI	MI	VI	I	UI	I	MI	NU	UI	I	I	I	I	I	MI
Expert 5	I	VI	VI	I	I	VI	UI	NU	I	VI	MI	NU	UI	I	NU	NU	I	I	I	I	I	UI
Expert 6	VI	VI	VI	VI	I	MI	NU	UI	I	I	I	MI	MI	I	NU	NU	VI	I	I	VI	VI	UI
Expert 7	I	I	I	I	I	I	NU	UI	VI	I	VI	MI	UI	MI	UI	UI	I	I	I	I	I	MI
Expert 8	VI	I	I	I	I	MI	NU	NU	VI	I	MI	MI	I	I	UI	UI	I	I	VI	I	I	UI
Expert 9	I	I	I		VI	VI	NU	NU	VI	VI	VI	MI	MI	VI	UI	UI	VI	VI	MI	I	VI	NU
Expert 10	I	VI	I	I	VI	I	UI	NU	MI	VI	VI	MI	MI	I	UI	UI	I	I	VI	I	I	MI
Expert 11	VI	VI	VI	I	MI	I	NU	UI	I	VI	VI	UI	MI	I	UI	MI	I	I	VI	I	I	I
Expert 12	VI	I	VI	VI	MI	MI	UI	UI	I	I	I	UI	MI	I	NU	MI	I	I	VI	VI	VI	UI
Expert 13	VI	I	I	VI	I	I	NU	NU	MI	MI	I	MI	UI	MI	NU	MI	VI	VI	I	I	I	UI
Expert 14	I	I	I	VI	I	I	UI	MI	VI	I	I	UI	UI	I	NU	MI	I	I	VI	I	VI	UI
Expert 15	VI	VI	I	VI	I	I	NU	UI	MI	VI	I	UI	MI	VI	NU	UI	I	I	MI	VI	MI	MI
Expert 16	VI	VI	VI	MI	MI	VI	NU	MI	MI	MI	VI	UI	UI	MI	UI	UI	I	I	UI	VI	MI	MI
Expert 17	VI	VI	I	I	VI	MI	NU	NU	VI	VI	VI	UI	I	VI	NU	MI	VI	VI	I	I	UI	MI
Expert 18	I	I	I	MI	I	I	UI	NU	VI	VI	I	MI	MI	VI	NU	UI	VI	I	MI	I	I	MI
Expert 19	VI	I	I	VI	VI	MI	UI	UI	I	VI	I	MI	MI	VI	UI	NU	I	I	MI	I	MI	UI
Expert 20	VI	VI	I	VI	VI	I	UI	UI	VI	I	I	MI	MI	I	UI	MI	VI	VI	MI	I	MI	UI

TABLE 6. List of refined qualitative attributes.

A_1	Accountability	A_{11}	Serviceability Type of Support
A_2	Security	A_{14}	Cost of storage
A_3	Capacity of CPU	A_{17}	Cost of VM
A_4	Capacity of Memory	A_{18}	Cost of Data
A_5	Capacity of Disk	A_{19}	Range of Response Time
A_9	Availability	A_{20}	Average of Response Time
A_{10}	Serviceability free support	A_{21}	Throughput

The amount of superiority of A_2 to A_1 is introduced by $V(A_2 \geq A_1)$ and the superiority of each attribute to another attribute is stored in $V_{n\times n}$n, where n is related to the number of quality attributes in each level or the number of sub-attributes

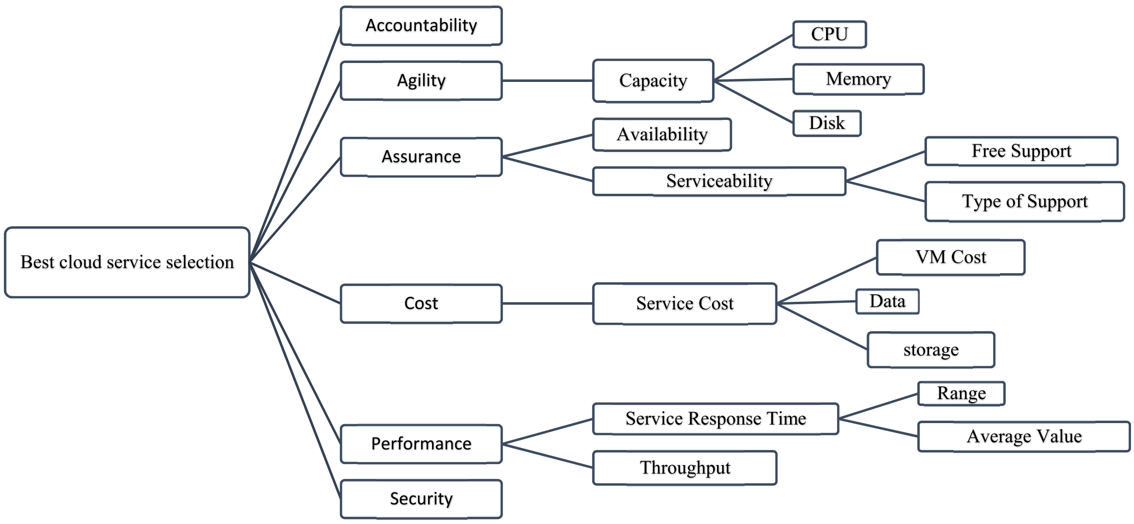

FIGURE 3. Hierarchical structure of qualitative attributes obtained from Algorithm 2.

TABLE 7. The utilized cloud services.

Top level QoS Group	QoS attributes	Service 1 (S1)	Service 2 (S2)	Service 3 (S3)
Accountability	Accountability level	4	8	4
Agility	CPU Capacity	9.6	12.8	8.8
	Memory Capacity	15	14	15
	Disk Capacity	1690	2040	630
Assurance	Availability	99.95%	99.99%	100%
	Serviceability Free Support	0	1	1
	Serviceability Type of support	24/7, Diagnostic Tools, Phone, Urgent Response	24/7, Diagnostic Tools, Phone, Urgent Response	24/7, hone, Urgent Response
	Service stability upload time	13.6	15	21
	Service stability CPU	17.9	16	23
	Service stability Memory	7	12	5
Cost	VM Cost	0.68	0.96	0.96
	Storage Cost	12	15	15
	Data Cost	10.5	12.5	12.5
Performance	Response Time	100	600	30
	Throughput	0.85	0.9	0.95
Security	Security level	4	8	4

in an attribute. It is defined by (11). Matrix V is shown in (12). The score of each quality attribute is calculated by minimizing each row of matrix V (MinV), which is shown in (13).

$$V(A \geq A_1) = \begin{cases} 1 & \textit{if } m_2 \geq m_1 \\ 0 & \textit{if } l_1 \geq u_2 \\ \dfrac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)} & \textit{otherwise} \end{cases} \quad (11)$$

$$V = \begin{bmatrix} v_{1.1} & \cdots & v_{1.n} \\ \vdots & \ddots & \vdots \\ v_{n.1} & \cdots & v_{n.n} \end{bmatrix} \quad (12)$$

$$MinV = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \end{bmatrix} = \begin{bmatrix} \min_{i=1}^{n}(v_{1.i}) \\ \min_{i=1}^{n}(v_{2.i}) \\ \cdots \\ \min_{i=1}^{n}(v_{n.i}) \end{bmatrix} \quad (13)$$

At last, MinV is normalized to calculate the weight of each attribute. The normalization weight vector for each attribute is defined as follows.

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \end{bmatrix} = \begin{bmatrix} v_1 / \sum_{i=1}^{n} v_i \\ v_2 / \sum_{i=1}^{n} v_i \\ \cdots \\ v_n / \sum_{i=1}^{n} v_i \end{bmatrix} \quad (14)$$

For example, the QoS data for sub-attributes of agility are shown in Table (8-a). Furthermore, the user requirements are shown in paired comparison matrix in Table (8-b).

We must first calculate the weight of each sub-attribute of capacity. Hence, we convert Table (8.b) to triangular fuzzy numbers. The triangular fuzzy matrix is as follows:

S, SM, SSM, and RSSM are computed according to (2), (3), (4), and (5) as follows:

$$SM = \begin{pmatrix} 3 & 7 & 11 \\ 2.2 & 4.33 & 7 \\ 1.4 & 1.67 & 3 \end{pmatrix} \tag{15}$$

$$SSM = \begin{pmatrix} 6.6 & 13 & 21 \end{pmatrix} \tag{16}$$

$$RSSM = (0.047619 \quad 0.076923 \quad 0.151515) \tag{17}$$

$$S = \begin{pmatrix} 0.1429 & 0.5385 & 1.6667 \\ 0.1048 & 0.3333 & 1.06061 \\ 0.0667 & 0.1282 & 0.4545 \end{pmatrix} \tag{18}$$

We compute the amount of superiority of these sub-attributes according to (6) as follows:

$$V = \begin{pmatrix} 1 & 1 & 1 \\ 0.8173 & 1 & 1 \\ 0.4319 & 0.6303 & 1 \end{pmatrix} \tag{19}$$

Finally, the score of each sub-attribute is computed according to (7), while normalizing each sub-attribute is given by:

$$score = \begin{pmatrix} 1 \\ 0.8173 \\ 0.4319 \end{pmatrix} = \begin{pmatrix} 0.4446 \\ 0.3637 \\ 0.1920 \end{pmatrix} \tag{20}$$

In the next step, the value of each cloud service from a particular attribute perspective is calculated by the AHP method. The values of CPU capacity according to Table (8-a) are given by:

$$\begin{array}{cccc} & S_1 & S_2 & S_3 \\ S_1 & \frac{1}{1} & \frac{9.6}{12.8} & \frac{9.6}{8.8} \\ S_2 & \frac{12.8}{9.6} & \frac{1}{1} & \frac{12.8}{8.8} \\ S_3 & \frac{8.8}{9.6} & \frac{8.8}{12.8} & \frac{1}{1} \end{array} \tag{21}$$

The sum of each row is calculated as (2.8409 3.7845 2.6042), and the values are normalized as:

$$\text{Vector}_{\text{CPU}} = (0.30780.410.2822) \tag{22}$$

The rest of the vectors are calculated and aggregated with the score of each sub-attribute.

$$vector_{Agility} = \begin{pmatrix} 0.3078 & 0.41 & 0.2822 \\ 0.3409 & 0.3181 & 0.3409 \\ 0.3623 & 0.4373 & 0.2002 \end{pmatrix} \times \begin{pmatrix} 0.4446 \\ 0.3637 \\ 0.1920 \end{pmatrix}$$

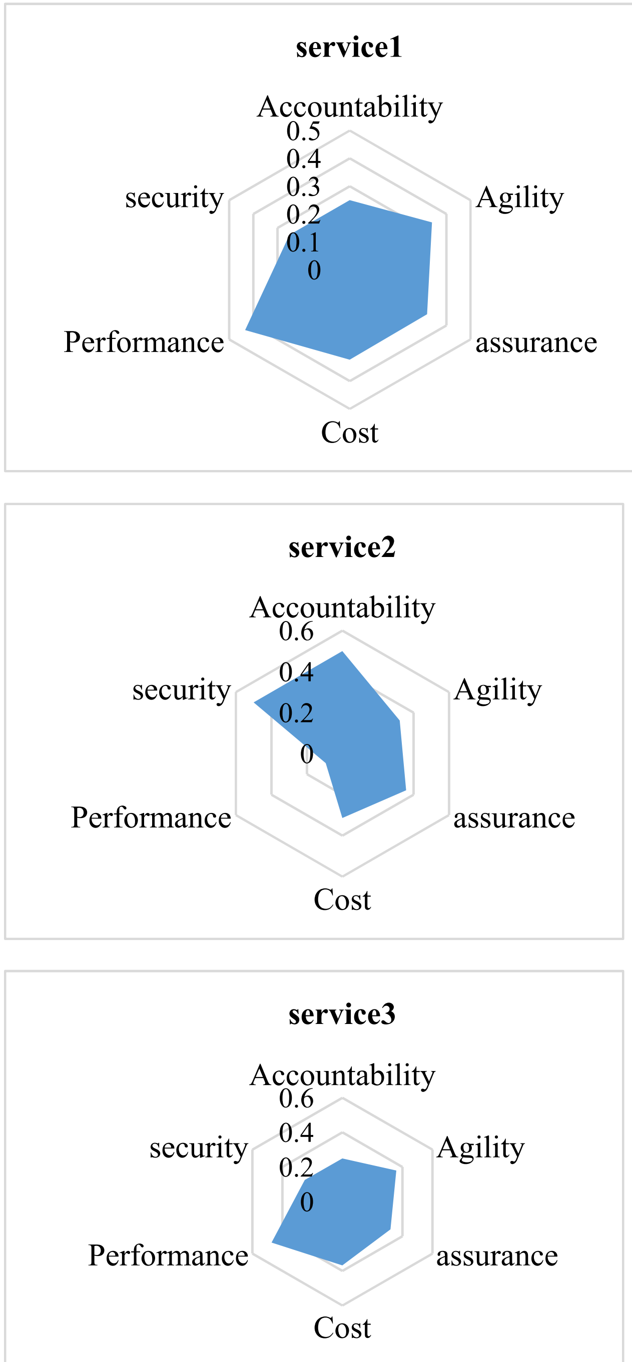

FIGURE 4. Comparison of cloud services based on top-level attributes (Three cloud service).

$$= \begin{pmatrix} 0.3402 \\ 0.3327 \\ 0.3586 \end{pmatrix} \tag{23}$$

Other quality attributes are computed and aggregated for three services. The results of the top-level indicate a multi-dimensional problem for ranking cloud services. The results can be analyzed in Figure (4).

Figure (4) shows that cloud services can be ranked differently according to user requirements. For example, if performance and agility have more priority for a user, cloud

TABLE 8. a) Comparison based on quality values. b) Fuzzy comparison between sub-attributes of agility based on user requirements.

	S_1	S_2	S_3
CPU Capacity	9.6	12.8	8.8
Memory Capacity	15	14	15
Disk Capacity	1690	2040	630

a

	CPU Capacity	Memory Capacity	Disk Capacity
CPU Capacity	$\tilde{1}$	$\tilde{3}$	$\tilde{3}$
Memory Capacity	$\tilde{3}^{-1}$	$\tilde{1}$	$\tilde{3}$
Disk Capacity	$\tilde{3}^{-1}$	$\tilde{3}^{-1}$	$\tilde{1}$

b

TABLE 9. Pairwise comparison between services for agility.

$C_{Agility}$	CPU Capacity			Memory Capacity			Disk Capacity		
CPU Capacity	1/00	1/00	1/00	1/00	3/00	5/00	1/00	3/00	5/00
Memory Capacity	0/20	0/33	1/00	1/00	1/00	1/00	1/00	3/00	5/00
Disk Capacity	0/20	0/33	1/00	0/20	0/33	1/00	1/00	1/00	1/00

service 1 will be the best selection. If assurance has more priority for a user, cloud service 1 will be the worst selection. The paired comparison matrix of top-level attributes is presented in Table (10).

The weights of the top-level qualitative attributes are calculated based on user requirements. Moreover, the values of qualitative attributes of services are considered. The final result is obtained by multiplying the weight of the requirement of cloud users with the quality of the cloud service. Finally, cloud services will be sorted as S3 > S1 > S2 where the ranking results are equal to the results of [14], [33].

$$\begin{bmatrix} 0.25 & 0.3402 & 0.3208 & 0.3229 & 0.4341 & 0.25 \\ 0.50 & 0.3327 & 0.3584 & 0.3143 & 0.0934 & 0.50 \\ 0.25 & 0.3586 & 0.3208 & 0.3629 & 0.4725 & 0.25 \end{bmatrix} \times \begin{bmatrix} 0.03 \\ 0.09 \\ 0.22 \\ 0.31 \\ 0.31 \\ 0.03 \end{bmatrix} = \begin{bmatrix} 0.3509 \\ 0.2652 \\ 0.3768 \end{bmatrix} \quad (24)$$

V. EVALUATION

To evaluate the proposed PFDR, we have implemented four experiments in MATLAB R2013a for evaluating scalability, execution time, accuracy, and sensitivity. Moreover, we experimentally obtain an optimal value of the threshold, which is mentioned in Algorithm 2. The results of experiments are compared with AHP [39] and FAHP [15].

A. SCALABILITY

In this test, scalability involves an increase in the number of services. The number of services varies from 3 to 1000. When in Figure (5) the number of services is 30, the execution time of AHP, FAHP, and PDFR are almost the same. When the number of services is more than 30, we can see a shorter execution time for PFDR.

With a large number of services, execution time is significantly different in the three methods. This difference is obvious from 300 services. The reason is that pairs of matrix calculations are performed more in AHP and FAHP. By reducing the number of quality attributes with PFDR, these pairwise comparisons will decrease, which shortens the execution time. Furthermore, FAHP has a longer execution time than AHP. FAHP has more calculations in its fuzzy commands, and it leads to a longer execution time.

B. EXECUTION TIME

In this test, we compute the time of receiving top-level attributes and compare the execution time for each attribute between PFDR, FAHP, and AHP. The results in Figure (6-a) show that the proposed method has less execution time. The reason is the use of fewer attributes for ranking the services, Also AHP has less execution time than FAHP. That is because of the Fuzzy computational and logical operations in the FAHP method. The total time of ranking calculation is shown in Figure (6-b). AHP uses matrices to compare the services and each service is compared according to quality attribute. Therefore, the use of fewer quality attributes causes less execution time.

In the PFDR, the execution time for the Assurance, Agility, and performance attributes is far shorter than the AHP method. That is because the number of quality attributes at the lower levels of the AHP method is more than the PFDR. In PFDR, some qualitative attributes are removed with the FDM-based algorithm. Therefore, AHP and FAHP methods need more calculations. In the cost attribute, the execution

TABLE 10. Paired comparison matrix between the top-level attributes based on SLA.

	Accountability	Agility	Assurance	Cost	Performance	Security
Accountability	$\tilde{1}$	$\tilde{3}^{-1}$	$\tilde{7}^{-1}$	$\tilde{9}^{-1}$	$\tilde{9}^{-1}$	$\tilde{1}$
Agility	$\tilde{3}$	$\tilde{1}$	$\tilde{3}^{-1}$	$\tilde{5}^{-1}$	$\tilde{5}^{-1}$	$\tilde{3}$
Assurance	$\tilde{7}$	$\tilde{3}$	$\tilde{1}$	$\tilde{3}^{-1}$	$\tilde{3}^{-1}$	$\tilde{7}$
Cost	$\tilde{9}$	$\tilde{5}$	$\tilde{3}$	$\tilde{1}$	$\tilde{1}$	$\tilde{9}$
Performance	$\tilde{9}$	$\tilde{5}$	$\tilde{3}$	$\tilde{1}$	$\tilde{1}$	$\tilde{9}$
Security	$\tilde{1}$	$\tilde{3}^{-1}$	$\tilde{7}^{-1}$	$\tilde{9}^{-1}$	$\tilde{9}^{-1}$	$\tilde{1}$

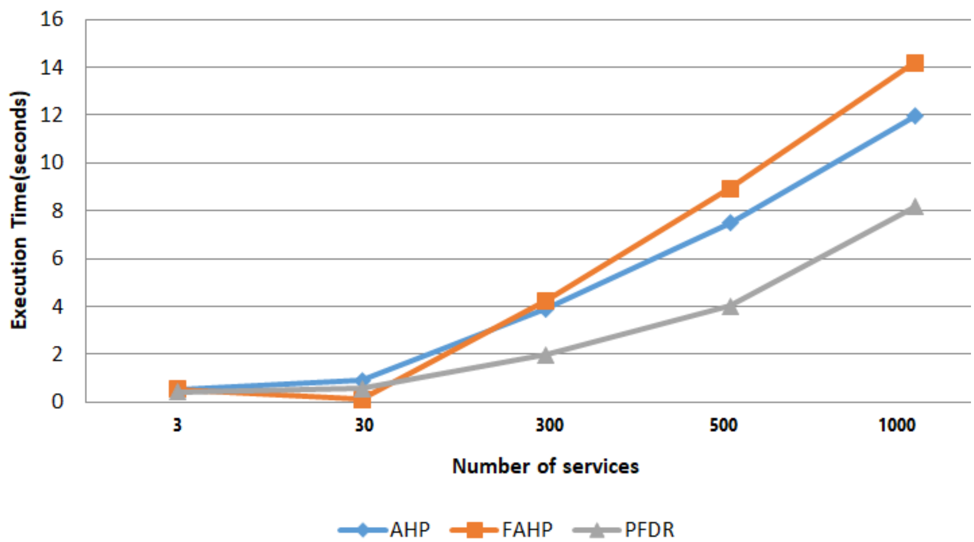

FIGURE 5. Execution time comparison of AHP, FAHP, and PFDR with increasing the number of services.

time of AHP is slightly shorter than PFDR. The reason is that the number of lower-level attributes in the PFDR and AHP is equal, and besides, time is needed in PFDR to execute fuzzy commands.

Pre-processing is related to the number of experts (e) and the number of initial attributes (a) which is e∗a. Calculating the weight of services (s) is related to the number of services and the number of quality attributes(m). Therefore, the time complexity is $O(m^2 \times s^3)$. Aggregation of the ranking results is related to the number of services and the number of quality attributes. Therefore, the time complexity will be $O(m^3 \times s^2)$.

C. ACCURACY

In this part, the accuracy of these methods is compared. Figure (7) depicts the results of comparing accuracy between FAHP, PFDR, and AHP for cloud service ranking. As can be seen, the quality results calculated by different methods are almost the same. The quality values calculated with AHP, FAHP, and PFDR for s1, s2, and s3 are cal almost equal.

This test reveals the advantages behind the objective of this research, which is to eliminate certain attributes to reduce the execution time without much influencing the quality of cloud service ranking. Table (11) shows the quality obtained for each cloud service. As can be seen, the best service ranking is almost the same in all methods.

To validate the accuracy of PFDR, we consider 100 Random SLAs for three cloud services. We compare the proposed algorithm with the methods obtained by AHP and FAHP.

As can be seen, there are four results of ranking among 100 scenarios that are different between PFDR and others. Three of them are highlighted colorless, indicating the best services similarly, but with different rankings. For example, both methods in the second SLA have the same service selection which is 2, but their ranking of them is different. One of the SLAs is marked darker. It shows the selection of the best services and ranking of them are distinguished differently. The results show a similarity of 96% in the cloud services ranking. There is also a 99% similarity in choosing the best cloud service.

D. SENSITIVITY ANALYSIS

The sensitivity analysis is used to assess the consistency of the results and to analyze how small changes will affect the newly proposed decision-making method. For this purpose, we consider several scenarios in such a way that each scenario creates a new situation with a change in weight priorities. Each scenario is examined whether the changes

TABLE 11. Results of ranking cloud services with different SLAs.

SLA#	PFDR	FAHP	AHP	SLA#	PFDR	FAHP	AHP	SLA#	PFDR	FAHP	AHP	SLA#	PFDR	FAHP	AHP	SLA#	PFDR	FAHP	AHP
1	3-1-2	3-1-2	3-1-2	21	2-3-1	2-3-1	2-3-1	41	2-3-1	2-3-1	2-3-1	61	3-2-1	3-2-1	3-2-1	81	2-3-1	2-3-1	2-3-1
2	2-1-3	2-3-1	2-3-1	22	2-3-1	2-3-1	2-3-1	42	3-2-1	3-2-1	3-2-1	62	2-3-1	2-3-1	2-3-1	82	2-3-1	2-3-1	2-3-1
3	2-1-3	2-1-3	2-1-3	23	2-3-1	2-3-1	2-3-1	43	2-1-3	2-3-1	2-3-1	63	2-3-1	2-3-1	2-3-1	83	2-3-1	2-3-1	2-3-1
4	2-1-3	2-1-3	2-1-3	24	2-3-1	2-3-1	2-3-1	44	2-3-1	2-3-1	2-3-1	64	2-3-1	2-3-1	2-3-1	84	3-2-1	3-2-1	3-2-1
5	3-2-1	3-2-1	3-2-1	25	2-3-1	2-3-1	2-3-1	45	2-3-1	2-3-1	2-3-1	65	2-3-1	2-3-1	2-3-1	85	2-3-1	2-3-1	2-3-1
6	2-1-3	2-1-3	2-1-3	26	2-3-1	2-3-1	2-3-1	46	2-3-1	2-3-1	2-3-1	66	3-2-1	2-3-1	2-3-1	86	2-3-1	2-3-1	2-3-1
7	2-3-1	2-3-1	2-3-1	27	2-3-1	2-3-1	2-3-1	47	2-3-1	2-3-1	2-3-1	67	2-3-1	2-3-1	2-3-1	87	2-1-3	2-1-3	2-1-3
8	3-2-1	3-2-1	3-2-1	28	2-3-1	2-3-1	2-3-1	48	2-3-1	2-3-1	2-3-1	68	2-1-3	2-1-3	2-1-3	88	3-2-1	3-2-1	3-2-1
9	2-3-1	2-3-1	2-3-1	29	2-3-1	2-3-1	2-3-1	49	3-2-1	3-2-1	3-2-1	69	2-3-1	2-3-1	2-3-1	89	2-3-1	2-3-1	2-3-1
10	3-2-1	3-2-1	3-2-1	30	2-3-1	2-3-1	2-3-1	50	3-2-1	3-2-1	3-2-1	70	2-3-1	2-3-1	2-3-1	90	3-2-1	3-2-1	3-2-1
11	2-1-3	2-1-3	2-1-3	31	2-3-1	2-1-3	2-1-3	51	2-3-1	2-3-1	2-3-1	71	2-1-3	2-1-3	2-1-3	91	2-3-1	2-3-1	2-3-1
12	2-3-1	2-3-1	2-3-1	32	2-1-3	2-1-3	2-1-3	52	2-1-3	2-1-3	2-1-3	72	3-2-1	3-2-1	3-2-1	92	2-3-1	2-3-1	2-3-1
13	3-1-2	3-1-2	3-1-2	33	2-3-1	2-3-1	2-3-1	53	2-3-1	2-3-1	2-3-1	73	2-3-1	2-3-1	2-3-1	93	2-3-1	2-3-1	2-3-1
14	3-2-1	3-2-1	3-2-1	34	2-3-1	2-3-1	2-3-1	54	3-2-1	3-2-1	3-2-1	74	2-3-1	2-3-1	2-3-1	94	2-1-3	2-1-3	2-1-3
15	2-3-1	2-3-1	2-3-1	35	3-2-1	3-2-1	3-2-1	55	2-3-1	2-3-1	2-3-1	75	2-3-1	2-3-1	2-3-1	95	2-1-3	2-1-3	2-1-3
16	2-3-1	2-3-1	2-3-1	36	2-3-1	2-3-1	2-3-1	56	2-3-1	2-3-1	2-3-1	76	3-2-1	3-2-1	3-2-1	96	2-1-3	2-1-3	2-1-3
17	3-2-1	3-2-1	3-2-1	37	2-1-3	2-1-3	2-1-3	57	2-3-1	2-3-1	2-3-1	77	3-2-1	3-2-1	3-2-1	97	2-3-1	2-3-1	2-3-1
18	2-3-1	2-3-1	2-3-1	38	2-3-1	2-3-1	2-3-1	58	2-1-3	2-1-3	2-1-3	78	2-3-1	2-3-1	2-3-1	98	2-3-1	2-3-1	2-3-1
19	2-3-1	2-3-1	2-3-1	39	2-3-1	2-3-1	2-3-1	59	2-3-1	2-3-1	2-3-1	79	2-3-1	2-3-1	2-3-1	99	2-3-1	2-3-1	2-3-1
20	2-3-1	2-3-1	2-3-1	40	2-3-1	2-3-1	2-3-1	60	2-3-1	2-3-1	2-3-1	80	2-1-3	2-1-3	2-1-3	100	2-3-1	2-3-1	2-3-1

make an impact on the decisions. According to [16], if the changes affect the decision-making and ranking, the model is sensitive, otherwise, it is robust. In various experiments, we examine the sensitivity by making small changes to evaluate the decision of the proposed method. Different scenarios are meant to change the weights of qualitative attributes derived from the SLA. Figure (9) shows the details of the sensitivity experiments according to the scenarios of Table (12).

As depicted in Table (12), fifteen different scenarios are considered. In each scenario, the user weights of two top-level attributes are replaced. For example, Scenario c1-c2 means that the weights of attribute c1 and attribute c2 have been interchanged. Results are obtained for the three cloud services.

Figure (8-a) shows the details of the sensitivity experiments based on ranking the services with PFDR according to Table (11). Figure (8-b) shows the details of the sensitivity experiments based on ranking the services with AHP. The results of 15 scenarios show that the results of ranking were unchanged in 14 experiments (i.e. all experiments have been equal except c1-c5), which was the same as in the AHP (all experiments have been equal except for c1-c5). The obtained results yield more than 93.3% robustness and less than 6.7% sensitivity to the attribute weights. Figures (8-c) and (8-d) show that the results of ranking between PFDR and AHP are the same, suggesting equal robustness.

TABLE 12. Fifteen scenarios for sensitivity analysis.

Scenario no.	Definition	S1	S2	S3
1	c1-c2	0.22191	0.41854	0.35955
2	c1-c3	0.20382	0.42875	0.36743
3	c1-c4	0.13434	0.52278	0.34289
4	c1-c5	0.19873	0.34411	0.45715
5	c1-c6	0.24025	0.42264	0.33711
6	c2-c3	0.2621	0.43784	0.30006
7	c2-c4	0.22646	0.5177	0.25584
8	c2-c5	0.284119	0.37645	0.33936
9	c2-c6	0.22985	0.43357	0.33658
10	c3-c4	0.22601	0.44543	0.32855
11	c3-c5	0.24359	0.39597	0.36043
12	c3-c6	0.23266	0.4833	0.28404
13	c4-c5	0.24586	0.40833	0.34582
14	c4-c6	0.17268	0.59529	0.23203
15	c5-c6	0.24025	0.42264	0.33711

E. EFFECT OF *th* IN PDFDM

This section investigates accuracy and execution time to identify the best value for *th* as mentioned in the PDFDM algorithm.

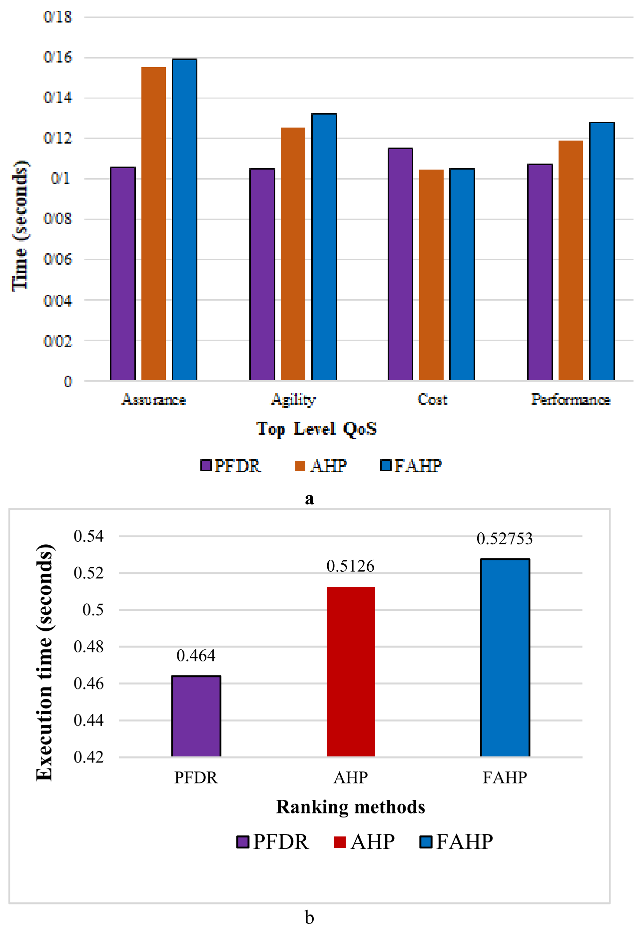

FIGURE 6. a) Comparison of the preparation time of weight values for top-level attributes. b) Comparison of total execution time for ranking.

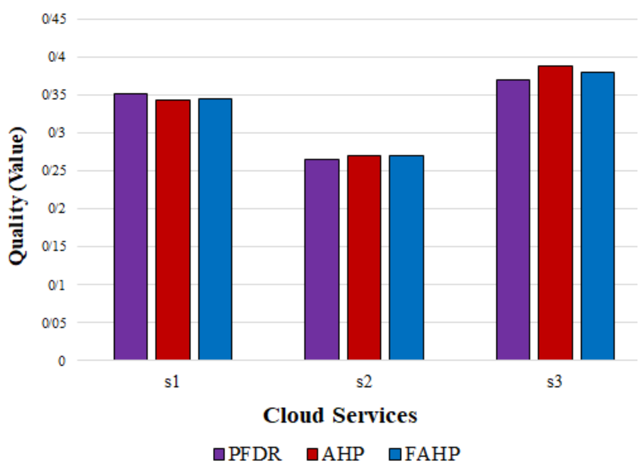

FIGURE 7. Comparing the results for cloud service ranking.

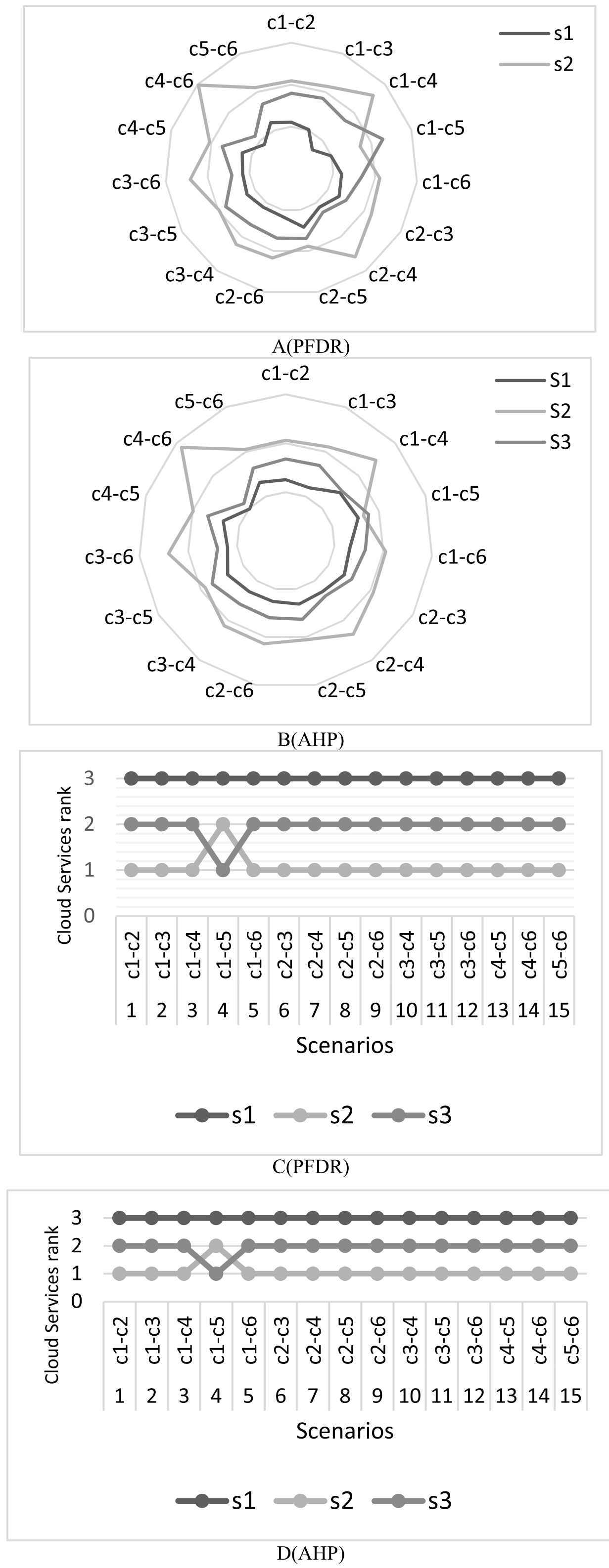

FIGURE 8. a) Sensitivity analysis for PFDR. b) Sensitivity analysis for AHP. c) The rank of cloud services in sensitivity analysis for PFDR. d) The rank of cloud services in sensitivity analysis for AHP.

We must consider threshold (*th*) in the PDFDM algorithm to determine whether the attribute must be accepted or rejected. The value of *th* ranges from 0.2 to 0.8. If the value of *th* is 0.2, all attributes will be selected, and the execution time will be 0.7444 as mentioned earlier. Furthermore, if the value of *th* is 0.9, none of the attributes will be selected. Therefore, we consider the values of *th* from 0.2 to 0.8. When all of the

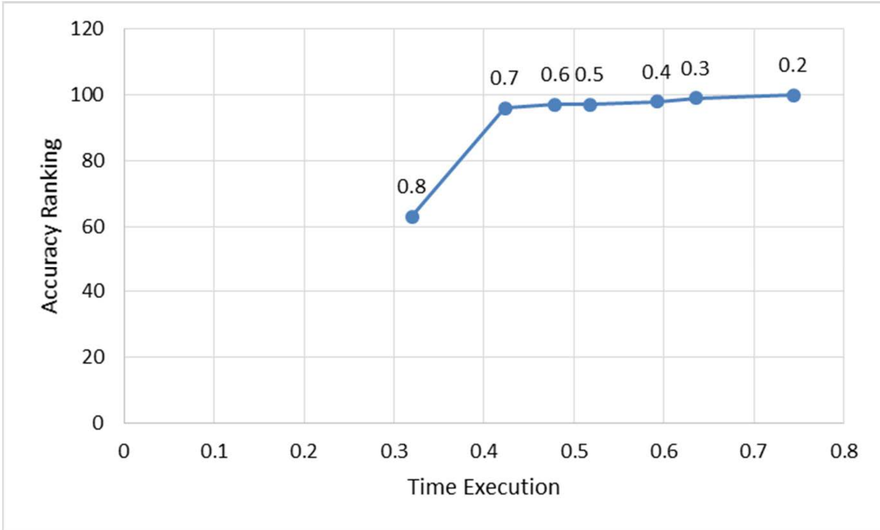

FIGURE 9. Effect of *th* on accuracy ranking and execution time.

attributes are selected, we consider 100 percent for accuracy. The other results are calculated with different values of the *th*. Furthermore, we consider three services and 100 different random user requirements for this test. The results for *th* from 0.2 to 0.8 are presented in Figure (9).

As can be found in Figure (9), if the value of *th* is 0.7, the accuracy will be decreased by 33 percent, and the execution time is underestimated to decrease 0.1 in comparison by the value of the *th* is 0.8. In comparison between 0.6 and 0.7, the ranking accuracy is changed in an underestimated way by 1 percent and execution time decreased about 0.05. Therefore, we can understand that the optimum value for *th* is 0.7 considering the execution time and accuracy.

VI. CONCLUSION

In this paper, we have presented the PFDR algorithm to perform a fast and accurate cloud service ranking. To this end, it eliminates the attributes that are not necessary for cloud users and are also unused in cloud ranking. The experimental results demonstrate the efficiency of our work in comparison with the existing methods. PFDR has achieved the same result (96%) in a shorter time. Also, the proposed method is robust. The results show that the robustness of the PFDR method is 93.6%.

In the future, we will extend our research with the aim of service selection in a faster and more accurate system. We plan to encode the user requirements and cloud services into interval code to ignore services that fall outside the range of user requirements.

In addition, the meta heuristic ant-colony optimization-based placement method performs better than the modified GA in terms of energy and SLA metrics whereas these techniques perform equally in terms of execution time for VM consolidation. Thus, increasing the number of objectives results in an escalation of execution time and degraded performance for some of the considered metrics. This is attributed to the fact that these methods attempt to figure out solutions that fulfill all the requirements at the same time.

REFERENCES

[1] M. Liaqat, V. Chang, A. Gani, S. H. A. Hamid, M. Toseef, U. Shoaib, and R. L. Ali, "Federated cloud resource management: Review and discussion," *J. Netw. Comput. Appl.*, vol. 77, pp. 87–105, Jan. 2017, doi: 10.1016/J.JNCA.2016.10.008.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009, doi: 10.1016/J.FUTURE.2008.12.001.

[3] H. Vahdat-Nejad, S. Izadpanah, and S. Ostadi-Eilaki, "Context-aware cloud-based systems: Design aspects," *Cluster Comput.*, vol. 22, no. S5, pp. 11601–11617, Sep. 2019.

[4] A. Jahani and L. M. Khanli, "Cloud service ranking as a multi objective optimization problem," *J. Supercomput.*, vol. 72, no. 5, pp. 1897–1926, May 2016.

[5] S. Lee and K.-K. Seo, "A hybrid multi-criteria decision-making model for a cloud service selection problem using BSC, fuzzy Delphi method and fuzzy AHP," *Wireless Pers. Commun.*, vol. 86, no. 1, pp. 57–75, Jan. 2016.

[6] J. C. Dueñas, W. L. de Oliveira, and A. Juan, "A software architecture evaluation model," presented at the Int. Workshop Architectural Reasoning Embedded Syst., 1998.

[7] S. Izadpanah, H. Vahdat-Nejad, and H. Saadatfar, "A framework for ranking ubiquitous computing services by AHP analysis," *Int. J. Model., Simul., Sci. Comput.*, vol. 9, no. 4, Aug. 2018, Art. no. 1850023.

[8] J. Siegel and J. Perdue, "Cloud services measures for global use: The service measurement index (SMI)," in *Proc. Annu. SRII Global Conf.*, Jul. 2012, pp. 411–415.

[9] M. Oriol, X. Franch, and J. Marco, "Monitoring the service-based system lifecycle with SALMon," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6507–6521, Nov. 2015.

[10] W. Halboob, H. Abbas, M. K. Khan, F. A. Khan, and M. Pasha, "A framework to address inconstant user requirements in cloud SLAs management," *Cluster Comput.*, vol. 18, no. 1, pp. 123–133, Mar. 2015.

[11] S. Singh and I. Chana, "Cloud resource provisioning: Survey, status and future research directions," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 1005–1069, Dec. 2016.

[12] D. Wang, Y. Yang, and Z. Mi, "A genetic-based approach to web service composition in geo-distributed cloud environment," *Comput. Electr. Eng.*, vol. 43, pp. 129–141, Apr. 2015.

[13] U. Shivakumar, V. Ravi, and G. R. Gangadharan, ''Ranking cloud services using fuzzy multi-attribute decision making,'' in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2013, pp. 1–8.
[14] S. Garg, S. Versteeg, and R. Buyya, ''A framework for ranking of cloud computing services,'' *Future Gener. Comput. Syst.*, vol. 29, no. 4, pp. 1012–1023, 2013.
[15] A. Ishizaka, ''Comparison of fuzzy logic, AHP, FAHP and hybrid fuzzy AHP for new supplier selection and its performance analysis,'' *Int. J. Integr. Supply Manage.*, vol. 9, nos. 1–2, pp. 1–22, 2014.
[16] R. R. Kumar, S. Mishra, and C. Kumar, ''Prioritizing the solution of cloud service selection using integrated MCDM methods under fuzzy environment,'' *J. Supercomput.*, vol. 73, no. 11, pp. 4652–4682, Nov. 2017.
[17] M. Goulão, V. Amaral, and M. Mernik, ''Quality in model-driven engineering: A tertiary study,'' *Softw. Quality J.*, vol. 24, no. 3, pp. 601–633, Sep. 2016.
[18] T. L. Saaty, ''Decision making with the analytic hierarchy process,'' *Int. J. Services Sci.*, vol. 1, no. 1, pp. 83–98, 2008.
[19] N. Dalkey, O. Helmer, and O. Helmer, ''An experimental application of the Delphi method to the use of experts,'' *Manage. Sci.*, vol. 9, no. 3, pp. 458–467, Apr. 1963.
[20] A. Habibi, F. F. Jahantigh, and A. Sarafrazi, ''Fuzzy Delphi technique for forecasting and screening items,'' *Asian J. Res. Bus. Econ. Manage.*, vol. 5, no. 2, pp. 130–143, 2015.
[21] A. Habibi, A. Sarafrazi, and S. Izadyar, ''Delphi technique theoretical framework in qualitative research,'' *Int. J. Eng. Sci.*, vol. 3, no. 4, pp. 8–13, 2014.
[22] V. D. Gracht and A. Heiko, ''Consensus measurement in Delphi studies: Review and implications for future quality assurance,'' *Technol. Forecasting Social Change*, vol. 79, no. 8, pp. 1525–1536, 2012.
[23] J. A. Somerville, *Effective use of the Delphi Process in Research: Its Characteristics, Strengths and Limitations*. Corvallís, OR, USA, 2018.
[24] D.-Y. Chang, ''Applications of the extent analysis method on fuzzy AHP,'' *Eur. J. Oper. Res.*, vol. 95, no. 3, pp. 649–655, 1996.
[25] L. A. Zadeh, ''Fuzzy sets,'' *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[26] V. X. Tran, H. Tsuji, and R. Masuda, ''A new QoS ontology and its QoS-based ranking algorithm for web services,'' *Simul. Model. Pract. Theory*, vol. 17, no. 8, pp. 1378–1398, Sep. 2009.
[27] R. Achar and P. S. Thilagam, ''A broker based approach for cloud provider selection,'' in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 1252–1257.
[28] C.-R. Choi and H.-Y. Jeong, ''Quality evaluation and best service choice for cloud computing based on user preference and weights of attributes using the analytic network process,'' *Electron. Commerce Res.*, vol. 14, no. 3, pp. 245–270, Nov. 2014.
[29] R. Gonçalves, Jr., T. Rolim, A. Sampaio, and N. C. Mendonça, ''A multi-criteria approach for assessing cloud deployment options based on non-functional requirements,'' in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 1383–1389.
[30] R. Karim, C. Ding, and A. Miri, ''An end-to-end QoS mapping approach for cloud service selection,'' in *Proc. IEEE 9th World Congr. Services*, Jun. 2013, pp. 341–348.
[31] J. Repschläger, S. Wind, R. Zarnekow, and K. Turowski, ''Decision model for selecting a cloud provider: A study of service model decision priorities,'' in *Proc. 19th Americas Conf. Inf. Syst. (AMCIS)*, vol. 2, 2013, pp. 1031–1041.
[32] H.-K. Kwon and K.-K. Seo, ''A decision-making model to choose a cloud service using fuzzy AHP,'' *Adv. Sci. Technol. Lett.* vol. 35, no. 1, pp. 93–96, 2013.
[33] M. Tajvidi, R. Ranjan, J. Kolodziej, and L. Wang, ''Fuzzy cloud service selection framework,'' in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 443–448.
[34] M. Abdel-Basset, M. Mohamed, and V. Chang, ''NMCDA: A framework for evaluating cloud computing services,'' *Future Gener. Comput. Syst.*, vol. 86, pp. 12–29, Sep. 2018.
[35] M. Godse, R. Sonar, and S. Mulik, ''Web service selection based on analytical network process approach,'' in *Proc. IEEE Asia–Pacific Services Comput. Conf.*, Dec. 2008, pp. 1103–1108.
[36] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, ''Qos ranking prediction for cloud services,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1213–1222, Jun. 2013.
[37] M. Ribas, C. G. Furtado, J. N. de Souza, G. C. Barroso, A. Moura, A. S. Lima, and F. R. C. Sousa, ''A Petri net-based decision-making framework for assessing cloud services adoption: The use of spot instances for cost reduction,'' *J. Netw. Comput. Appl.*, vol. 57, pp. 102–118, Nov. 2015.
[38] Z. U. Rehman, O. K. Hussain, and F. K. Hussain, ''Parallel cloud service selection and ranking based on QoS history,'' *Int. J. Parallel Program.*, vol. 42, no. 5, pp. 820–852, 2014.
[39] S. K. Garg, S. Versteeg, and R. Buyya, ''SMICloud: A framework for comparing and ranking cloud services,'' in *Proc. 4th IEEE Int. Conf. Utility Cloud Comput.*, Dec. 2011, pp. 210–218.
[40] C. Jatoth, G. R. Gangadharan, U. Fiore, and R. Buyya, ''SELCLOUD: A hybrid multi-criteria decision-making model for selection of cloud services,'' *Soft Comput.*, vol. 23, no. 13, pp. 4701–4715, Jul. 2019.
[41] D. Julong, ''Grey control system,'' *J. Huazhong Univ. Sci. Technol.* vol. 3, no. 9, p. 18, 1982.
[42] (2019). *Amazon EC2*. Accessed: May 30, 2019. [Online]. Available: https://aws.amazon.com/ec2/
[43] (2019). *Microsoft Azure*. [Online]. Available: https://azure.microsoft.com/en-us/
[44] (2019). *Rackspace*. [Online]. Available: https://www.rackspace.com/
[45] *Software Engineering-Productquality—Part 1: Quality Model*, Standard ISO, IEC 9126-1, ISO/IEC 9126-1:2001, International Standards Organization, Geneva, Switzerland, 2001.
[46] *Software Engineering-Product Quality—Part 2: External Metrics*, Standard ISO, IEC 9126-2, ISO/IEC 9126-2:2003, International Standards Organization, Geneva, Switzerland, 2003.
[47] J. Y. Lee, J. W. Lee, D. W. Cheun, and S. D. Kim, ''A quality model for evaluating software-as-a-service in cloud computing,'' in *Proc. 7th ACIS Int. Conf. Softw. Eng. Res., Manage. Appl.*, 2009, pp. 261–266.
[48] M. S. Goraya and D. Singh, ''A comparative analysis of prominently used MCDM methods in cloud environment,'' *J. Supercomput.*, vol. 77, no. 4, pp. 3422–3449, Apr. 2021.
[49] M. H. Nejat, H. Motameni, H. Vahdat-Nejad, and B. Barzegar, ''Efficient cloud service ranking based on uncertain user requirements,'' *Cluster Comput.*, vol. 25, no. 1, pp. 485–502, Feb. 2022.
[50] A. Hussain, J. Chun, and M. Khan, ''A novel customer-centric methodology for optimal service selection (MOSS) in a cloud environment,'' *Future Gener. Comput. Syst.*, vol. 105, pp. 562–580, Apr. 2020.
[51] M. H. Nejat, H. Motameni, and H. Vahdat-Nejad, ''An integrated optimal method for cloud service ranking,'' *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 29, no. 7, pp. 2983–3003, Nov. 2021.

ZONGBAO MA was born in Qinghai, in 1974. He received the Bachelor of Science degree from Shaanxi Normal University, in 1995, and the master's degree in education from Shaanxi Normal University, in 2006.

From 1997 to 2022, he was a Lecturer with the Department of Computer and Science, School of Information Engineering, Xi'an University of Arts and Science. He is the author of two books and more than ten articles. His research interests include computer networking, software modeling, software engineering, programming, and algorithm analysis. His research interests also include mathematical modeling, optimization algorithms, information theory, and cryptography theory.

MOHAMMAD HOSSEIN NEJAT received the M.S. degree in computer engineering–software engineering from Islamic Azad University, Zanjan, Iran. He is a Software Engineering with Islamic Azad University, Sari, Iran. His current research interests include software quality, cloud computing, and evolution algorithms.

HAMED VAHDAT-NEJAD received the bachelor's degree from the Sharif University of Technology, in 2004, the master's degree from the Ferdowsi University of Mashhad, in 2007, and the Ph.D. degree from the Computer Engineering Department, University of Isfahan, in 2012. He was a Research Scholar with the Middleware Laboratory, Sapienza University of Rome, from 2011 to 2012. He leads the Pervasive and Cloud Computing Laboratory, University of Birjand. He was a Visiting Professor with Superior University, Lahore, in 2018; and Daffodil International University, Dhaka, in 2017. He is currently an Assistant Professor with the Computer Engineering Department, University of Birjand. He has (co)published about 50 papers in conferences and journals. His research interests include pervasive computing, cloud computing, and context awareness. He has served as a TPC Member for many conferences. He has served as the Chairperson for the 1st and 2nd International Workshop on Context-Aware Middleware for Ubiquitous Computing Environments, the "3rd, 4th, and 5th International Workshop on Pervasive and Context-Aware Middleware" as well as the 1st Conference on Healthcare Computing Systems and Technologies. Previously, he has served as an Associate and Guest Editor for *Computers and Electrical Engineering* (Elsevier) and *Journal of Computing and Security*.

BEHNAM BARZEGAR received the B.S. and M.S. degrees in computer engineering from Islamic Azad University, Sari, Iran, in 2006 and 2009, respectively, and the and Ph.D. degree in computer engineering from Islamic Azad University, Najafabad, in 2018. He is an Associate Professor with the Department of Computer Engineering, Islamic Azad University, Babol, Iran. His research interests include green cloud computing, task scheduling, and formal methods (Petri Net).

SAEED FATEHI received the M.S. degree in computer engineering–software engineering from Islamic Azad University, Iran. He is a Faculty Member with the Department of Computer Engineering, University College of Allameh Helli Chaloos, Chalus, Iran. His current research interests include software quality and cloud computing.