ITC 1/49

Information Technology and Control
Vol. 49 / No. 1 / 2020
pp. 113-126
DOI /10.5755/j01.itc.49.1.23251

Service Association Factor (SAF) for Cloud Service Selection and Recommendation

Received 2019/04/19 | Accepted after revision 2020/01/22

http://dx.doi.org//10.5755/j01.itc.49.1.23251

**HOW TO CITE:** Rabbani, I. M., Aslam, M., Martinez-Enriquez, A. M., Qudeer, Z. (2020). Service Association Factor (SAF) for Cloud Service Selection and Recommendation. *Information Technology and Control*, 49(1), 113-126. https://doi.org//10.5755/j01.itc.49.1.23251

# Service Association Factor (SAF) for Cloud Service Selection and Recommendation

**Imran Mujaddid Rabbani**

Dept. of Computer Science & Engineering, University of Engineering and Technology; Lahore;
e-mail: imranmrabbani@gmail.com

**Muhammad Aslam**

Dept. of Computer Science & Engineering, University of Engineering and Technology; Lahore;
e-mail: maslam@uet.edu.pk

**Ana Maria Martinez-Enriquez**

Dept. of CS, CINVESTAV-IPN, D.F. Mexico; e-mail: ammartin@cinvestav.mx

**Zeshan Qudeer**

Dept. of Computer Science, University of South Asia; Lahore; e-mail: shaniboje@icloud.com

**Corresponding author:** imranmrabbani@gmail.com

Cloud computing is one of the leading technology in IT and computer science domain. Business IT infrastructures are equipping themselves with modern regime of clouds. In the presence of several opportunities, selection criteria decision becomes vital when there is no supporting information available. Global clouds also need evaluation and assessment from its users that what they think about and how new ones could make their selection as per their needs. Recommended systems were built to propose best services using customer's feedback, applying quality of service parameters, assigning scores, trust worthiness and clustering in different forms and models. These techniques did not record and use interrelationships between the services that is true impact of service utilization. In the proposed approach, service association factor calculates value of interrelations among services used by the end user. An intelligent leaning based recommendation system is developed for assisting users to select services on their respective preferences. The evaluation of this technique, based on leading service providers, makes evident the excellent performance of the approach on all types of cloud models tested.

**KEYWORDS:** Service Selection, Recommendation Systems, Service Association Factor, Interrelationships.

## 1. Introduction

Cloud Computing (CC) is a recent advance in Computer Science, permitting its users the power of processing, information stockpiling, infrastructure modeling, platforms, and assets services on pay-per-use philosophy like electricity, gas, water consumption. CC aims to give secure, versatile, robust, feasible, and fault tolerant services through web-based applications [6, 18]. CC is on-demand remote access to a shared pool of configurable computing services [7]. It provides scalable resources from the well-maintained pool of virtual resources meeting user-defined system requirements and adopts utility type charging system [15]. CC is a fast growing computing platform which emphasizes on how consumers, proprietors, service providers, and government bodies store, process, and utilize their information and computations [11]. CC is a type of rental business, applications, processes, and infrastructures [16]. CC provides customers a pay-as-you-go all the time, and the provision of such service is becoming famous due to it is on-demand services [25].

CC is growing in popularity due to its fundamental characteristics of remoteness, security, rent-base, and dynamicity [10]. According to the National Institute of Standards and Technology (NIST), three models exist of CC:

1 Software as a service (SaaS) is a massive pool of built-in applications, e.g., GMAIL, Salesforce, Workday, Concur, WebEx, Ms. Office 365, Box, DocuSign, and Dropbox.

2 Platform as Service (PaaS) is a facility for the development of new applications for the end-user like Force.com, Google App Engine, CloudBees, Mendix, Redhat, Acquia, and SQL Azure.

3 Infrastructure as a Service (IaaS) is the utilization of virtual resources, for instance, AWS, Cisco Metaboard, Google Compute Engine, and Windows Azure.

Various cloud service providers in the market are Google storage (Google Drive), a software application (Google Apps on Play store), and in a platform (Google Apps Engine) [6]. Similarly, Microsoft providing Azure, Intune, Cloud Platform, while Amazon deals with EC2 and AWS, and many others discussed in [28]. The prominent vendors are Google, Microsoft, Flexiscale.com, salesforce.com, Rackspace, Amazon, and RightScale [7, 19].

The rapid growth of services has created a challenge in service selection for customers [8] and have generated a decision-making problem. SMICloud [6] provides a framework and a mechanism to index the cloud providers based on customer's needs. Cloud services are ranking based on quality parameters providing an index. The customers use them to compare cloud services according to their desired quality requirements, but parameters and services are not dependent on each other. Similarly, a Green Cloud [9] is a simulator that provides adequate information regarding the energy consumption of data center components (network hubs, servers, and other devices). This technique misses essential quality parameters, like security, speed, and performance. Another iCanCloud company predicts the trade-off between cost and performance of the application's pool on particular hardware [17]. It is hypervisor, customizable, user-friendly, and provides POSIX-based API. It is basically hardware dependent and does not support other hardware and quality parameters. Simulation Paradigm, Simulation Program for Elastic Cloud Infrastructures (SPECI) permits pre-information component scaling and performance attributes of concerning data centers [22]. SPECI works in two forms: a) Datacenter layout and topology to provide operations on observed data centers and facilitate to add monitoring points. b) A measuring and execution of experiments, as desired. The simulation does not recognize relationships among services.

This study resolves the issue of ranking made on feedback that may be biased and leading to misleading for new entrants. Also, the service recommendation based on service's interrelationships shows actual trust of the past customers. Besides, this learning-based approach allows dynamically to check the quality based parameter during the evaluation process to render the state of the art service recommendation.

The organization of the paper is as follows: Section 2 presents issues arisen in cloud service selection and motivation for this study. Section 3 explores different cloud service selection policies, while Section 4 defines the problem using the Service Association Factor (SAF). Section 5 presents the framework for the process recommendation, and Section 6 discusses the obtained results. Finally, Section 7 draws conclusions and future research work.

## 2. Cloud Services Issues and Motivation

CC is a newly emerging field of IT and growing at high speed. Recently, globalized services are available in a geo-distributed manner with minimum cost and high availability [20]. Issues in this domain are broadly categorized into three forms:

*CC Standardization* is the first issue related to survival. The customers start losing their trust on the vendors and commitment to cloud services. New customers hesitate to implement or shift their legacy applications into cloud's planet. Shortly, CC could suffer its effectiveness and market due to unavailability of standards especially in security and interoperability [6, 16].

*Selection and Recommendation* is the second issue. Signing online Service Level Agreement (SLA) is always at risk before knowing the cloud services and its vendors [1]. There is no measurement or mechanism tool available for evaluation of performance and reliability of cloud services. Due to the fast-growing cloud services, it is difficult to know which vendor has a reputation to trust in the particular quality of service (QoS) provided [6].

*Independence* and the *interrelationship* between services and users is the third issue. The user can block itself into the single provider's regime when it is supposed to use packaged services. On the other hand, the vendors are not able to have their service performance and utilization information in the market. If so then, they could revise their service packages, features, proper marketing strategies. Also, there are no cloud customer relationship Standard Operating Procedures (SOP's).

Some of the other issues are *Service Level Agreements* (SLA) and its *evaluation*, *access controls*, *resource management*, *security*, *QoS*, *scalability*, *unclear* user's requirements, *economics*, *energy* management [2, 15, 16].

The purpose of this study is to explore the potential solutions for each issue above mentioned and to precise limitations. The primary focus is to explore occurred interrelationships between services and lay down a new paradigm for service selection by users based on their dreamed outcomes using associations amongst.

## 3. Related Work

Internet provides cloud selection's information, but a typical user is unable to get and grasps it in short time. Some recommender systems have been developed for assistance and support based on users' needs and expectations and presented in [21]. The recommender chooses one option among the pool. The recommender systems can be collaborative, content-based, knowledge-based, and hybrid [5, 13, 28]. Current upcoming business systems require and demand seamless coordination among various business processes and online applications over the Internet.

The business intelligence (BI) inclusion for any business application becomes crucial. So, by meeting the end user needs and expectations, delivery of the efficient, and quality based application is a significant challenge for dynamic applications and traffic on the Internet [26, 27].

A cloud service selection policy to meet these concerns is a performance factor that is an agent-based approach to recommends cloud services on user's feedback against quality parameters [5]. The company value, defined in this approach, could not be materialized, and it should be independent of the vendor's perspectives. Historical data is another approach [7] utilized over text-based analysis due to synonyms polysemy problems and applied collaborating filtering on user's interactions. The primary focus is the users' IDs and their operational data, but the evaluation does not take into account the performance and other quality parameters.

The algorithm defined in [12, 14] works on QoS constraints like response time, cost-effectiveness, and delays. This algorithm does not include the users' interaction. The two approaches described are a) the combinatorial multi-choice knapsack; b) the Directed Acyclic Graph (DAG) constrained shortest path finding technique.

The broker system [24] generates Cloud Service Provider Index (CSP-Index) using its properties and focus remains on company provided properties and its services. Similarly Trust Evaluation Metric [1] is fuzzy logic-based approach for the evaluation of the service provider's trustworthiness and provides the degree of trust. In this approach relationship among the services is missing and degree of trust's value may vary from user to user.

The QoS historical based technique [8] ranks the services according to user's criteria in different time slots and may lead to worst rankings if the user has selected wrong time slots. A cloud recommender system mentioned in [19], grouped the user's feedback using K-mean technique and ranks the services into different clusters based on quality parameters. Fixed number of clusters may lead to confrontation of rankings. Table 1 shows some of the vital cloud service selection policies.

**Table 1**

Cloud Service Selection Approaches

| Name | Description |
|---|---|
| SMI Cloud [6] | Ranking (prioritize) the services on measurement index based on quality parameters. |
| Cooperative Relationship [29]. | Based on the historical usage and the recommendation of service. It calculates the relationship with each other on QoS similarities. |
| Financial based metrics [10] | The evaluation and measures of financial metrics like Cost-Benefit Analysis (CBA), Return on Investment (ROI), and Total Cost Ownership (TCO)). The are 18 key performance indicators (KPI) of Cloud Computing (CC). |
| Cloud workflow systems based constraints [15] | The user can also define the constraint conditions of function attributes. The Service Selection Algorithm based on Constraints (SSAC) for cloud workflow system makes the service choice by matching processes between services demands and their characteristics. |

Feedback and historical data may lead to inconsistent information regarding QoS because the survey forms are defined to get their interest but may not capture real usage of the service. One user may be impressed with the interface and other with Mean Time to Failure (MTTF) only. Without a careful administration of QoS, critical business applications may suffer impending execution debasements and results in practical disappointments and financial disasters.

Online resources provide rich information about services, but a typical user can be unable to find precise information on an urgent basis [3]. [28] categorizes many recommended systems, based on users' requirements and expectations, employed for specific scenarios where more than one option exist. Business intelligence (E-business applications) becomes crucial to know the end user requirements. Due to the dynamicity and impulsive form of business applications and the data load on the Internet, the provision of quality is a significant challenge [14, 20].

SMICloud [16] provides a framework and a mechanism to index the cloud providers on customer's needs and ranks services based on quality parameters. SMICloud identifies service measurement indexes which are essential for cloud evaluation. Customers use them to compare services on their desired quality requirements. However, these parameters and services are not dependent on each other.

Cloud services system applied for on-demand vehicles management [4] fulfills the request of users. The software components are monitoring of workflow systems, resource management, and high availability. The best service is the result of the evaluation and assessment of all cloud services analyzed.

Simulation Program for Elastic Cloud Infrastructures (SPECI) [24] allows previous information component scaling and parameters of performance for data centers. The recommendation uses these metrics, but post-evaluation of features does not exist.

Cloud service selection using agents in [14] through the recommendation system uses the feedback from users with the help of structured graphical user interface and also utilizes their past experiences. The multi-agent system applied K-mean clustering for its evaluation and assessment and is useful for SaaS, IaaS, and PaaS models. The system also utilizes trust parameters for enhancing its recommendation list of services to its esteemed clients.

Performance factor is an agent-based cloud service selection approach which, recommends services on user's feedback on different quality attributes [5]. The constant company value is useful to compare the feedback results. The value should not be imposing from the vendor's perspective. A similar approach [15] uses historical data over text-based analysis and applies collaborating filtering on user's interactions. The primary focus is on ID's of users and their operational data, but it lacks proper assessment of quality attributes. The algorithm defined in [14, 20] works on QoS constraints with response time and cost-effectiveness. These predefined clusters are based on the quality parameters that represent the performance. Fixation may lead to biases in the ranking of services. The broker system generates CSP-Index using the properties and focus remains on company provide properties and their services [29].

Performance base ranking model suggested by Sahar [20] for selecting SaaS applications. It applies relative service ranking vector and SMICloud toolkit and uses performance base quality parameters to assess the feedback gathered in tabular form. It produces service ranks for selecting best services. However, it is not covering other importance QoS parameters during its assessment.

Inter-trust relationship model permits the vendor to borrow further resources from external service providers using cloud federation framework and serves the client as a single point of resources [12].

Several, already proposed algorithms are unable to find a proper service selection on user requirements using associations hidden in the services utilization.

To overcome this issue, the study of several applied algorithms and heuristics shows that until now, there no exists one that can recommend the best service according to users' requirements. Besides, these solutions do not include associations or interrelations among services. The approach proposed here takes into account the association among services included in the package used by customers. Besides, the proposed technique will be indeed helpful to QoS brokers for making effective online decisions and will also increase their market.

## 4. Problem Formulation Using Service Association Factor (SAF)

The Service Association Factor (SAF) finds the associations/interrelations among the services. SAF is a degree or frequency of occurrences of services appeared together in a package provided by the vendor and utilized by the customer.

The formulation of the problem takes into account various possible issues by analyzing quality parameters associated with particular questions regarding their internal/external service structure. Each service consists of a set of characteristics of quality attributes.

**Cloud Service Provider (CSP):** company makes different available cloud services grouped into a single unit and served to customers as a package. These services can be SaaS, PaaS, IaaS. As per Microsoft Azure, CSP is a third-party company offering IaaS, PaaS or SaaS, as a homeowner as pay per usage units. Google, Amazon, and Microsoft are major service providers in the market. CSPs defines a set of services: $C = \{C_1, C_2 ... C_p\}$

**Services 'S':** is a facility/feature which is offered by a CSP via the internet. The service might be available on-demand and may be controlled/managed by its provider. The approach assume that the services are of different types: IaaS, SaaS, PaaS or DBaaS, and are possible to use and economically feasible to the general users. Formally the set of services: $S = \{S_1, S_2 ... S_m\}$

**User 'U':** is an entity that utilizes the services of CSP under the selected package. The set of 'n' users is $U = \{U_1, U_2 ... U_n\}$

**Package 'P':** is a group of services treated as a single unit which can be IaaS, SaaS, PaaS, and DBaaS. Mainly two types of packages exist.

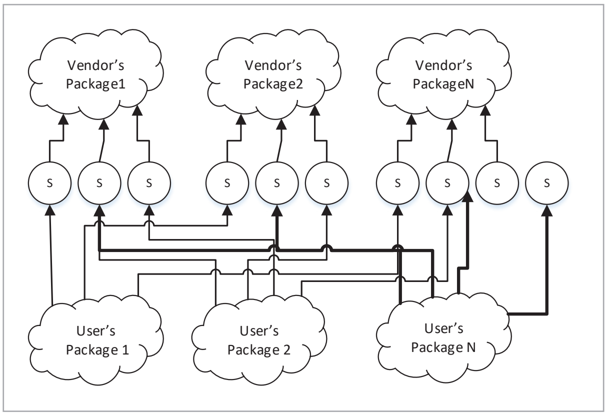
**Package defined by the Company [Pc]:** CSP selects related services and groups them as a single unit. A single vendor provides and controls $\mathbf{P_c}$.

**User-defined package [$\mathbf{P_u}$]:** defines the selected services according to the user's needs. $P_u$ comes from different providers, and a third party vendor or agent controls them.

Figure 1 shows the process of packages formation. There are three vendors of packages and three user-defined packages, along with their three services

**Figure 1**

Clarification of users' & Vendors' Packages



each. The user package has service from any of the vendors as this package reflects its perspectives. There is a discrete limit on the number of services available under any package.

**Service Association Factor SAF 'a':** The association **'a'** represents the relationship among the services included in a package or a module type. Also, it represents the users' trust in the service. The calculation of the association adds all appearances of services $S_i$ & $S_k$ included in the available packages. The SAF value is irrespective of packages' type being on use by customers. Figure 2 shows a service $S_i$ used along with $S_k$ service, and it counts to unit '1' (Equation (1)). This process applies to all packages where both services occurred together. Mathematically, it can be described as follows:

$$A(S_i, S_k) = 1+1+\ldots+1. \tag{1}$$

Mathematical Formulation:

There are some packages for user to make selection, so it can be: P = {$P_1$, $P_2$... $P_n$} .

Every package is by definition

$$P_i \,\epsilon\, P_{ui}\ \mathrm{V}\ P_{ci}, \tag{2}$$

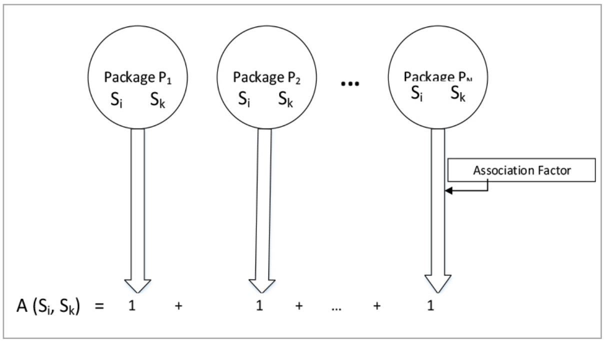
where $P_i$ is $i^{th}$ Package, $P_{ui}$ is ith user defined package and $P_{ci}$ is $i^{th}$ company defined package, for finding the association factor 'A' between the services as follows:

$$A(S_k) = \sum_{i=1}^{n} A(S_k, S_i), \tag{3}$$

where $S_i \,\epsilon\, P_i$ by definition.

**Figure 2**

Process for Association Factor's Calculation



**Recommendation Process:** There are two approaches to start with the recommendation process. In the first approach, no prior knowledge is available and is called base process. In the second, knowledge is available and recommendation is made on the maximum value of association factor of available services.

**Base Process:** When there is no knowledge available to the system and no service(s) or package(s) recommended in the past. In this scenario, the system has no service recommendation record. So;

$$S_i = \mathrm{Nil} \quad \text{and} \quad A= \{0 \quad if\ S_i = \phi\ ]. \tag{4}$$

This process occurred when the system runs first time, and no service is recommended because there no prior knowledge is available. In this scenario, the user selects the service or package by its own depending on its choice/desire/experience. Hence no suggestion will be provided.

**Knowledge Process:** Past information is available in the repository and recommendation of service / package is made on the available knowledge, and Si exists. The choice for next service Sk will be made on the highest value of the Association Ak. The above mentioned process can be defined in the matrix form as:

$$A= \{ \max(A_i) \quad if\ S_i = \{S_1, S_2 \ldots S_i\}. \tag{5}$$

The repository RA of all the association factors among two consecutive service Si & Sj is described by the following matrix.

$$R_A \begin{bmatrix} a_{11} & a_{12} \ldots & a_{1j} \\ a_{21} & a_{22} \ldots & a_{2j} \\ \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} \ldots & a_{ij} \end{bmatrix}. \tag{6}$$

where aij represents the value of association factor of service $S_i$ with $S_k$.

For recommendation, an array La of maximum SAF values is generated as follows:

$$L_a = \{a_1, a_2 \ldots a_n\}, \tag{7}$$

where an represents the recommended services with maximum SAF values and n is total number of recommended services. As a final step, to provide quality base service from user's perspectives, a Quality_Checker function is applied. A list of user's quality parameter Qu from defined array of quality parameters Q = {Q1, Q2...Qm}. The Quality_Checker function is defined as:

$$L_R = \forall_i \forall_k \text{Quality\_Checker}(L_i, Q_k), \tag{8}$$

where Quality_Checker is the quality filtration function, i is total number of services in La to be evaluated and k is the total number of quality parameters selected by the user.

The Quality_Checker function deduces all those services which are not fulfilling the user's quality and offers the best services. The SAF methodology also allows to recommend a complete package having best services on user's desires. It is defined by the following triplet:

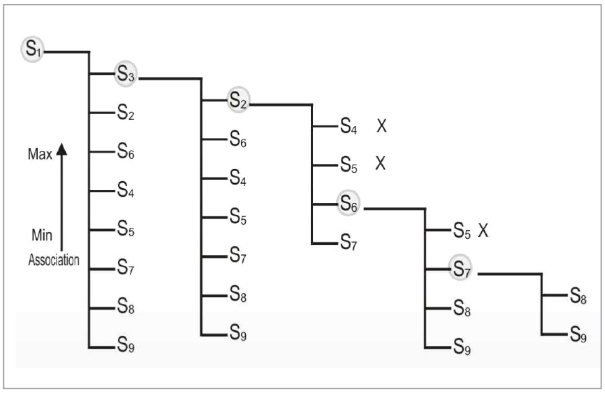
$$(P_R, S_R, A_R), \tag{9}$$

where $P_R$ is the package 'R' to be recommended, SR is a list of Service and $A_R$ is the association factor of '$S_R$' with all other services $S_1, S_2 \ldots S_n$.

Further, recommendation of a particular service is also described in Figure 3. Where after choosing service, all the associated services are ordered ascendingly with respective SAF values. So, if database of all services, their associations and relevant packages is maintained then applying the above mentioned process, one can find its best services of his / her own choice.This technique allows learning the past behavior of user which package(s) or service(s) he has used. When a new user wants to use this system, with the help of association factor, it proceeds towards best available services in the market. In Figure 3, user starts with one service say $S_1$, and after selection of this service, the system recommends those services which have association value with $S_1$, in ascending order. On top, services are listed with higher association value and help in choosing best service accordingly.

**Figure 3**

Process of recommendation using SaF



## 5. Cloud Service Recommendation Using SAF

The designed, developed and tested system offers registration to both customers and service providers. The registered users can set their criteria for selecting desired services.

Constraints are available in the form of Quality_of Service (QoS) parameters: performance, price, time frame, Availability, MTTF, MTTR, and some required resources The repository contains only the evaluated services that satisfy the constraints defined by the user. In the second step, all services which are going to propose or recommend have the respectively associated factor. At the end of the process, the system displays services ranked by the association factor in increasing order and recommends them accordingly.

The components of the system are as follow:

### 5.1. Registration

It is the process by which different CSPs register themselves with the object for advertising their businesses and enhance their interaction within the digital globe. On the other hand, users can also register to the system to find out best services available and to seek required services as well. Through this portal, the user can contact best service providers in the market. It has two main components.

a **CSP Registration:** Each Cloud Service Provider registers into the system. The information of a registered service includes functionality, QoS parameters, the price per usage, availability constraints if any, and standards. This information becomes public.

b **User Registration:** By this component, customers register into the system to have information about CSPs and services: functionalities, the process of adoption, and to select services accordingly.

## 5.2. Evaluation and Assessment

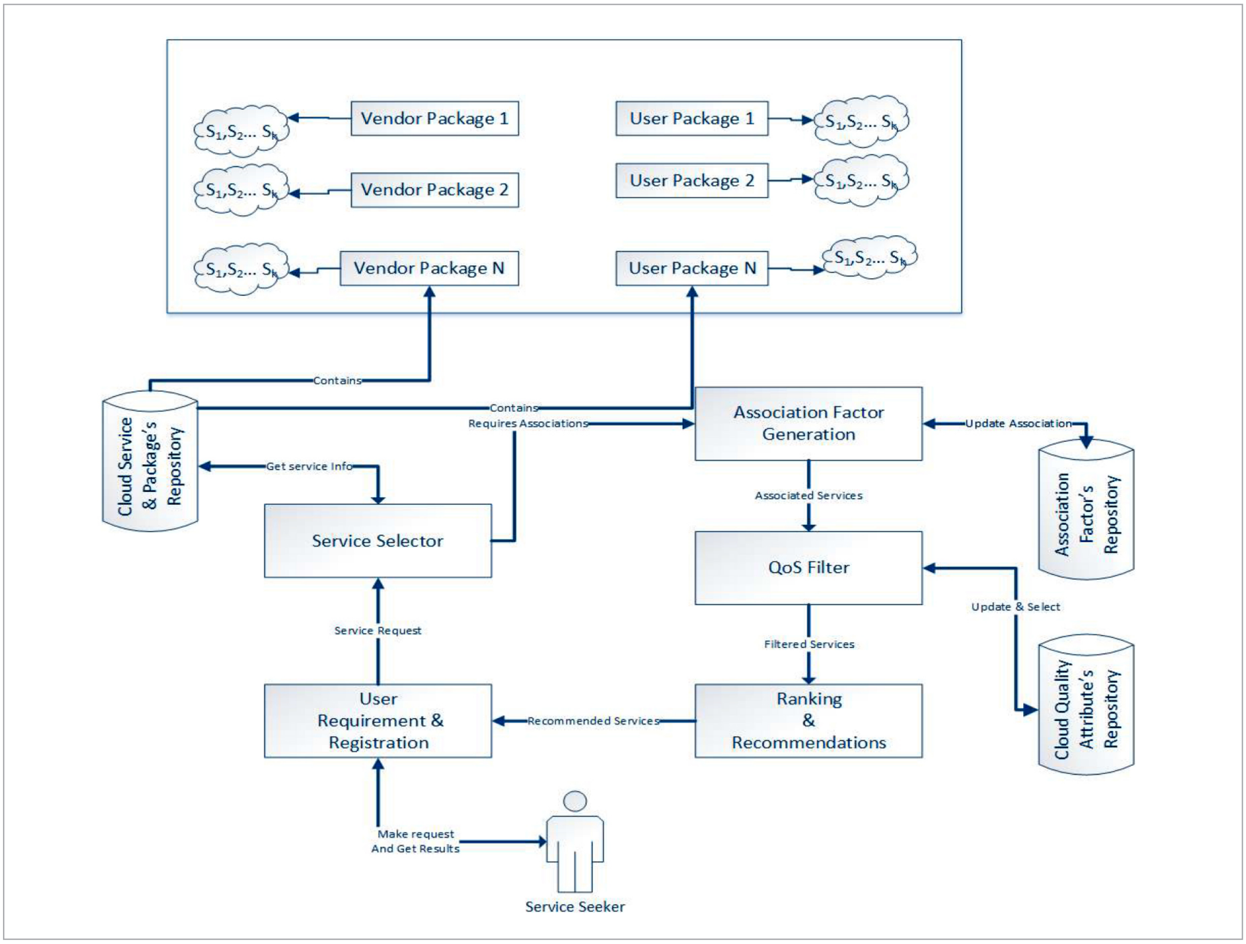
The evaluation of services takes into account the different quality parameters and standards requested by users. The parameters come from either side of the solution, but the client side provides real and accurate results of users' experience and feedback [23]. The quality parameters are performance, reliability, fault tolerance, interoperability, among others. The assessment follows the proposed paradigm. The system mainly focuses on the client side (user experience and requirements). The system provides recommendations after drawing the ranking of the best available cloud service based on user requirements.

## 5.3. Association Generation

The goal of the analysis of user services' history is to find hidden relationships. Each new creation or selection of a package, either by provider or user, implies the computation and registration of SAF to maintain the repository updated. The system preserves a re-

**Figure 4**

Cloud Service Selection System



pository to hold the SAF values between cloud services occurred together. The two types of packages: User and Vendor (as defined in the previous section) have the same SAF due to the service occurs in either package used by the end user.

### 5.4. Recommendations

The system holds the repository of services, association factors, and QoS attributes. The users can consult them along-with their feedback and customary history. It also contains 'Cloud Package's Repository'. On the request of a seeker, the system calculates the SAF for all related services and test them. This process is known as QoS_check, as shown in Figure 4. The system displays a recommended ranked list of services based on SAF criteria.

### 5.5. Implementation of Repositories

The database or repository is an essential component of the system to maintain pratcial informationl. The system has a database layer for storing its ongoing information in the proper form. The Database as a service (DBaaS) available on the Microsoft Azure platform achieves this goal, and all its services are being on use under registered login provided by the university (student login). The Azure platform created on online service of the 'CSSA' database (on the link rabbani.database.windows.net). The central database contains sub-databases used to maintain and control the information being on use in the system defined as follows (see Figure 5):

#### Cloud Service Provider's Database 'CSPDb'

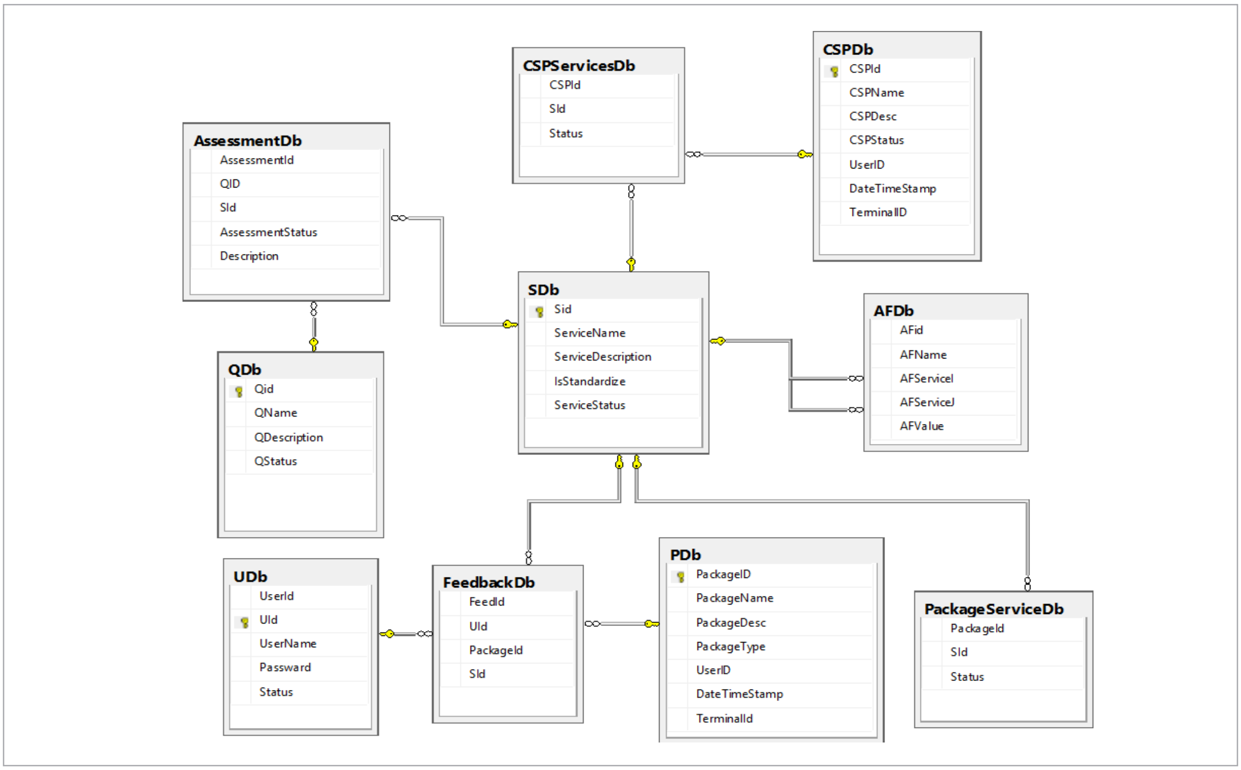
CSPDb contains a list of CSPs along with their work description and services to the public. CSPDb includes: 'Name', Unique 'Identification Number', CSP description, and 'active-status' shows whether the provider is live or disconnected.

#### Cloud Service Database 'SDb'

SDb contains complete detail of different services provided by different CSPs. Whenever a new service

**Figure 5**

Database Structure of Proposed System



appears into the system, SDb retains it. It has the 'Name' of the service as Google Drive; the 'Description' has complete service information so that a new user could well inform the features and functionality attached; 'IsStandardize' depicts whether or not the service follows some standard; 'ServiceStatus' has the availability status.

**Customer / User's Database 'UDb'**

Udb holds the data of all customers of respective services and service provider on the Internet community along with their respective information like contact and service information and nature of the job. Udb shows which user is using which service.

**Package's Database 'PDb'**

Pdb is a very important database of the system that holds services used together and formed into a single object known as a package. Pdb contains 'Name' of the package and a list of services in the form of 'serviceList' to store all information services packed as a single object.

**Association Factor's Database 'AFDb'**

AFDb contains the associations between the services available in the repository. It has two services' named as 'ServiceI' & 'ServiceJ' to whom the association is computing. 'AFValue' holds the value of the association factor among these services. The 'AFid' records the relation for identification.

**Quality of Service Attribute's Database 'QDb'**

Qdb contains the QoS parameters, essential to computes assessment and evaluation of services. They play an essential role in the service recommendation process. It has 'QId' as an identification number, 'QName' serves as the name of the quality attribute and similarly 'QStatus' record the activeness of the attribute for future correspondence.

**Assessment's Database 'AssessmentDb'**

AssessmentDb carries the information regarding service assessment and evaluation on different quality parameters required by users. 'QId' denotes the quality attribute, whereas 'Sid' depicts the service on which evaluation takes place.

**Feedback Database 'FeedbackDb'**

The FeedbackDb repository records information about the user feedback on the service provider or user-defined package and its services as well. It contains 'Sid' for service, 'UId' of the User, and 'Packaged' for the package.

## 5.6. Algorithm

The primary objective of the system is to recommend services based on the required user package. The **Make_User_Package** procedure computes the association factor SAF of services package and the user itself. The following pseudo-code describes the process for making a new package:

```
1. Procedure Make_User_Package ()
2. Service $S_1, S_2 ... S_k$
3. Package $P_1, P_2 ... P_k$
4. Select Service s
5. Select Package p
6.     Flag IsEnd ← False
7.     p.Add(s)
8.     While (Not(IsEnd))
9.         Association a[] ← Calculate_Association(s)
10.        Sort(a)
11.        s ← User_Choice(a) // select the user's choice
12.        p.Add(s)
13.                if (choiceEnds) IsEnd ← True
14.    return p // having all services based on
                users' choices
15. End
```

The procedure defines *Services* at line 2; Packages at line 3; The selection of first service **s** is at line 4; and similarly, a selected blank package '**p**' at line 5; a Boolean flag 'IsEnd' initialized with False at line 6 determines when the user has found the desired service. The first selected service 's' initializes the package 'p' at line 7. The system generates associations w.r.t 's' through the 'Calculate_Association(s)' function and stores them in the array Association 'a[ ]' at line 9. the *Sort* function of these associations '*a* ' are at line 10 and presented to the user for its selection at line 11. Once the user made its choice, the selected service becomes part of the package at line 12. This process continues until the user terminates the selection. Finally, the full 'p' package returned to the user of its choice is at line 14.

## 5.7. Experimentation of SAF Method

The method finds the association among the services and tracks the trends of users' service usage in the past. This association represents the relevancy between two services ($S_i$, $S_j$). It represents the occurrence of services used together in the contract/package (from service provider).

It shows the occurrence of services ($S_i$, $S_j$... $S_K$), of the packages: $P_1$, $P_2$ ... $P_N$, suppose that there are three packages:

$P_1 = S_1, S_5, S_7, S_{13}$ $P_2 = S_5, S_7, S_{15}, S_{22}$ $P_3 = S_1, S_5, S_{10}, S_{15}$

**Table 2**

Determination of Association among Services

| Services | $S_1$ | $S_5$ | $S_7$ | $S_{10}$ | $S_{13}$ | $S_{15}$ | $S_{22}$ |
|---|---|---|---|---|---|---|---|
| $S_1$ | --- | 2 | 1 | 1 | 1 | 1 | 0 |
| $S_5$ | 2 | --- | 2 | 1 | 1 | 2 | 1 |
| $S_7$ | 1 | 2 | --- | 0 | 1 | 1 | 1 |
| $S_{10}$ | 1 | 1 | 0 | --- | 0 | 1 | 0 |
| $S_{13}$ | 1 | 1 | 0 | 0 | --- | 0 | 0 |
| $S_{15}$ | 1 | 2 | 1 | 1 | 0 | --- | 1 |
| $S_{22}$ | 0 | 1 | 1 | 0 | 0 | 1 | --- |

Given package Pi, the cell (Si, Sj) includes all mutual occurrence of Si with other Sj (see Table 2). This process repeats for all 'i' from 1 to the maximum number of available services, in the present case 22. For instance, the association between services $S_5$ and $S_7$ is two because they occurred twice, represented in the cell (2, 3) of Table 2. Similarly, $S_5$ and $S_{15}$ is also two in cell (2, 6) where cell (x, y) represents as 'x' is the row number and 'y' is column number.

## 6. Experimental Results and Discussions

The experimentation processes the different CSP information; it recovers the user feedback regarding CSP vendors, as well as the utilization and working of their services. Table 3 presents the average values obtained in this study and shows the complete analysis of Google, Microsoft & Amazon, along with the selected services.

**Table 3**

Cloud Service Vendors with Services

| Methods | Google | Amazon |
|---|---|---|
| Google Search | Bing | Amazon Cloud Search |
| Google Drive | SkyDrive | Simpe Storage Service |
| Gmail | SendGrid Email | Amazon SES |
| Chrome OS | Windows Azure | Elastic Compute Cloud |

**Table 4**

Comparison of Rankings

| Methods | Google | Microsoft | Amazon |
|---|---|---|---|
| k-Mean Clustering [19] | 2.75 | 3 | 1 |
| Performance Factor [18] | 4.35 | 2 | 0.3875 |
| Service Association Factor (SAF) | 4.2 | 3.6 | 3.6 |

Three approaches have been selected as mentioned, in cloud vendors and their services respectively on 20-user data set. Table 4 shows the obtained results. The first two techniques are dependent on user feedback. In [19] defines a fixed number of clusters and K-mean clustering evaluates the feedback. The produced recommendation results are in the form of rankings of cloud service vendors, whereas, in performance factor [18] approach, there is a company value as a fixed element. The calculation uses user feedback and ranking accordingly. The service association factor 'SAF' is independent of all defined values to calculate the frequency of service utilization by the end users. Table 4 shows the obtained results, which draw the following conclusions:

- On average, the Google cloud service provider is best among the other two approaches.
- Amazon is the worst case according to the standard deviation on three factors of different providers.
- SAF is most consistent among different vendors with the least standard deviation.
- Inconsistency found on service feedback because it does not represent a real picture of service utilization (user may provide feedback on his/her own choice). On the other side, SAF shows an accurate assessment of service utilization because the calculation takes into account the services utilized by the customer, so SAF produces more precise and accurate results.

The analysis includes three datasets: a group of 5-users, 10-users, and 20-users, to see the effectiveness of SAF on the user-defined packages. The used techniques are feedback (Performance Factor/K-Mean Clustering), SAF with vendor's package, and SAF with user-defined package. The graphs depicted in Figure 6 (a, b, c) show the results.

The *feedback* technique remains at the lower level, while SAF recommendation using defined vendor's package works far better. Furthermore, the results are more efficient than the rest of both approaches when the association factor includes the user-defined packages. SAF remains consistently at the higher end, and recommendations are more effective and efficient.

**Figure 6 (a)**

Comparison of techniques with 5-Users' Dataset

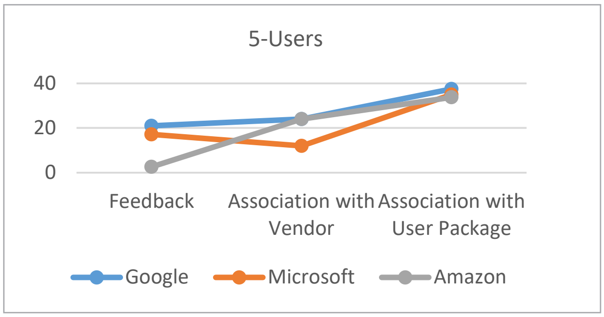


**Figure 6 (b)**

Comparison of techniques with 10-Users' Dataset

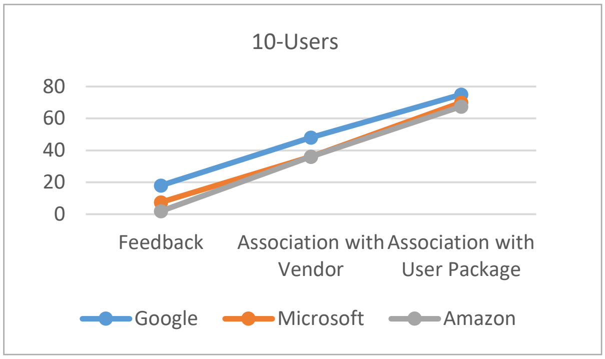
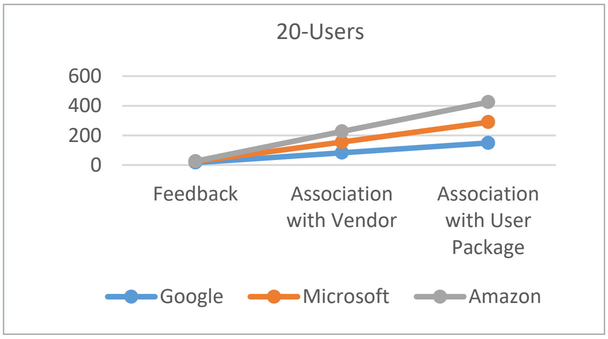


**Figure 6 (c)**

Comparison of techniques with 20-Users' Dataset



## 7. Conclusion

Cloud service selection takes a new dimension for itinerent users to filter out the best services according to their needs. The present approach solves this issue by introducing Service Association Factor (SAF) to calculate the occurrence of services appeared together within a package. SAF is independent of any fixation of firm parameters. The developed learning system assesses and evaluates the available services on quality parameters and customer criteria demand. The results show an excellent performance of the system and techniques applied. The applied techniques have limitations in terms of sole type of services, quality, defined values, and bondage of a single service provider.

Future work will include more QoS parameters and also enhance the approach to the vendors' side, for instance, how a vendor can increase the perspectives of the market.

## References

1. Alhamad, M., Dillon, T., Chang, E. A Trust-Evaluation Metric for Cloud Applications. International Journal of Machine Learning and Computing, 2011, 1(4), 416-421. https://doi.org/10.7763/IJMLC.2011.V1.62
2. Benslimane, Y., Plaisent, M., Bernard, P., Bahli, B. Key Challenges and Opportunities in Cloud Computing and Implications on Service Requirements: Evidence from a Systematic Literature Review. 2014 IEEE 6th Inter-

national Conference on Cloud Computing Technology and Science. 2014. https://doi.org/10.1109/CloudCom.2014.115

3. Chan, N. N., Gaaloul, W., Tata, S. A Recommender System Based On Historical Usage Data for Web Service Discovery. Service Oriented Computing and Applications, 2012, 6(1), 51-63. https://doi.org/10.1007/s11761-011-0099-2
4. Fornaia, A., Napoli, C., Tramontana, E. Cloud Services for On-Demand Vehicles Management. Information Technology And Control, 2017, 46(4), 484-498. https://doi.org/10.5755/j01.itc.46.4.17331
5. Francesco R., Lior R., Bracha S., Paul B. K. Recommender Systems Handbook. 2011. ISBN 978-0-387-85819-7. http://doi.org/10.1007/978-0-387-85820-3, Springer New York Dordrecht Heidelberg London. https://doi.org/10.1007/978-0-387-85820-3
6. Garg, S. K., Versteeg S., Buyya R. SMICloud: A Framework for Comparing and Ranking Cloud Services. Fourth IEEE International Conference on Utility and Cloud Computing, 2011, 210-218. https://doi.org/10.1109/UCC.2011.36
7. Goga, K., Terzo, O., Ruiu, P., Xhafa, F. Simulation, Modeling and Performance Evaluation Tools for Cloud Applications. In Proceedings of the 8th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS2014). 2014, 226-232. https://doi.org/10.1109/CISIS.2014.32
8. Hussain, O. K., Hussain, F. K. Multi-Criteria IaaS Service Selection Based On QoS History. IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, 1129-1135. https://doi.org/10.1109/AINA.2013.158
9. Kliazovich, D., Bouvry, P., Khan, S. U. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. The Journal of Supercomputing, 2010, 62(3), 1263-1283. https://doi.org/10.1007/s11227-010-0504-1
10. Kornevs, M., Minkevica, V., Holm, M. Cloud Computing Evaluation Based on Financial Metrics. Information Technology and Management Science, 2012, 15(1), 87-92. https://doi.org/10.2478/v10313-012-0013-8
11. Kushida, K. E., Murray, J., Zysman J. Diffusing the Cloud: Cloud Computing and Implications for Public Policy. Journal of Industry, Competition and Trade, 2011, 11(3), 209-237. https://doi.org/10.1007/s10842-011-0106-5
12. Li, L., Shuxian, G., Dongmei, F., Miao, Z., Rajkumar, B. A New Multi-Objective Evolutionary Algorithm for Inter-Cloud Service Composition. KSII Transactions on Internet and Information Systems, 2018, 12(1), 2018. https://doi.org/10.3837/tiis.2018.01.001
13. Lu, J., Shambour, Q., Xu, Y., Lin, Q., Zhang, G. A Web-Based Personalized Business Partner Recommendation System Using Fuzzy Semantic Techniques. Computational Intelligence, 2012, 29(1), 37-69. https://doi.org/10.1111/j.1467-8640.2012.00427.x
14. Mahmood, A., Shoiab, U., Shahzad, M. S. A Recommendation System for Cloud Services Selection Based on Intelligent Agents. Indian Journal of Science and Technology, 2018, 11(9), 1-6. https://doi.org/10.17485/ijst/2018/v11i9/119843
15. Mao, L., Yang, Y., Xu, H., Chen, Y. Service Selection Algorithm Based on Constraint for Cloud Workflow System. Journal of Software, 2013, 8(5), 1124-1131. https://doi.org/10.4304/jsw.8.5.1124-1131
16. Nazir, M., Nitish, B., Rahul, K. C, Raj, G. M. Cloud Computing: Current Research Challenges. Book Chapter of Cloud Computing: Reviews, Surveys, Tools, Techniques and Applications. An Open-Access eBook published by HCTL Open, 2015
17. Núñez, A., Vázquez-Poletti, J. L., Caminero, A. C., Castañé, G. G., Carretero, J., Llorente, I. M. iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator. Journal of Grid Computing, 2012, 10(1), 185-209. https://doi.org/10.1007/s10723-012-9208-5
18. Rabbani, I. M., Muhammad, A., Martinez, E. Intelligent Cloud Service Selection Using Agents. Advances in Intelligent Systems and Computing, 2013, 105-114. https://doi.org/10.1007/978-3-642-37371-8_14
19. Rabbani, I. M., Shah, A. A., Aslam, M. Evolution of Cloud Computing & Its Future. In Proceedings of the 2012 1st International Conference on Future Trends in Computing and Communication Technologies, IEEE, 2013, 111-115. OAI Identifier: oai.CiteSeerX.psu:10.1.1.303.5175
20. Sahar, A. E., Adil, Y., Mohammed, B. B. Performance Based Ranking Model for Cloud SaaS Services. International Journal of Information Technology and Computer Science, 2017, 9(1), 65-71, https://doi.org/10.5815/ijitcs.2017.01.08
21. Shrivastava, A., Rajawat, A. S. A Review on Web Recommendation System. International Journal of Computer Applications, 2013, 83(17), 14-17. https://doi.org/10.5120/14668-2842

22. Sriram, I. SPECI, A Simulation Tool Exploring Cloud-Scale Data Centres. Cloud Computing, 2009, 381-392. https://doi.org/10.1007/978-3-642-10665-1_35

23. Subha, M., Saravanan, K. Achieve Better Ranking Accuracy Using Cloudrank Framework for Cloud Services. International journal of Emerging Trends and Technology (IJETT), 2014, 307-312. arXiv preprint arXiv:1402.2509

24. Sundareswaran, S., Squicciarini, A., Lin, D. A Brokerage-Based Approach for Cloud Service Selection. IEEE Fifth International Conference on Cloud Computing, 2012, 558-565. https://doi.org/10.1109/CLOUD.2012.119

25. Whaiduzzaman, M., Gani, A., Anuar, N. B., Shiraz, M., Haque, M. N., Haque, I. T. Cloud Service Selection Using Multicriteria Decision Analysis. The Scientific World Journal, 2014, 1-10. https://doi.org/10.1155/2014/459375

26. Yu, T. Lin, K.-J. Service Selection Algorithms for Web Services with End-to-end QoS Constraints. Information Systems and e-Business Management, 2005, 3(2), 103-126. https://doi.org/10.1007/s10257-005-0052-z

27. Yu, T., Zhang, Y., Lin, K.-J. Efficient Algorithms for Web Services Selection with End-to-end QoS Constraints. ACM Transactions on the Web, 2007, 1(1), 1-6. https://doi.org/10.1145/1232722.1232728

28. Zain, T., Aslam, M., Imran, M. R., Martinez, E. Cloud Service Recommender System Using Clustering. 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2014, 1-6. https://doi.org/10.1109/ICEEE.2014.6978334

29. Zhang, C., Bian, J., Cheng, B., Li, L. A Personalized Cloud Services Recommendation Based on Cooperative Relationship between Services. Journal of Software Engineering and Applications, 2013, 06(12), 623-629. https://doi.org/10.4236/jsea.2013.612074