

**ITC 1/48**

Journal of Information Technology  
and Control  
Vol. 48 / No. 1 / 2019  
pp. 104-114  
DOI 10.5755/j01.itc.48.1.21272

**Internet Service Selection Using Service  
Association Factor (SAF)**

Received 2018/07/18

Accepted after revision 2019/01/08


<http://dx.doi.org/10.5755/j01.itc.48.1.21272>

# Internet Service Selection Using Service Association Factor (SAF)

## 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

Corresponding author: imranmrabbani@gmail.com

The modern organizations are equipping themselves with modern current regime of online application infrastructures through cloud, fog, and edge computing. In the presence of several opportunities, the selection criterion for such Internet-based services becomes vital, especially, when there is no supporting information available. Existing recommender systems provide services by evaluating the quality of service parameters, k-mean clustering, and fuzzy logic techniques on customer's feedback. However, these schemes typically rely on customers' feedback and do not provide any information on the interrelationship between the services. Feedback may be self-generated or biased and leading to improper recommendation to service seekers. To resolve the issue, we propose an innovative service association factor method that calculates the value of interrelationships among services appearing together as a package. This technique is implemented based on an intelligent agent that evaluates the values of the associations on standards and quality attributes. It enables the users to select the best services on their preferences. The proposed agent works on fog environment near to the customers. The technique is tested on leading cloud vendors. The results show that the system meets the desires of service seekers in all service models in an efficient manner.

**KEYWORDS:** Internet Services, Service Selection, Service Association Factor, Fog Agent.

## 1. Introduction

In the modern era, online Internet-based service follows different models of Cloud computing (CC), Fog computing (FC), and the Edge computing (EC). CC is

on-demand, scalable remote access to a shared pool of configurable computing resources/services and adopts utility type charging system [6, 14]. The objective is

to offer a fast-growing rental computing platform and how consumers, proprietors, service providers, and governments store, process and utilizes their information and computation powers [12, 15]. Better *security, versatility, robustness, fault tolerance* are additional benefits through web-based applications [18].

CC grows in the business sector and is getting popularity due to its fundamental characteristics, for instance, *remoteness, utility-oriented charging, and security* [11]. According to the National Institute of Standards and Technology (NIST), the three service models of CC are: a) *software as a service (SaaS)* is a pool of built-in applications, b) *platform as service (PaaS)* allows facilities for the development of new applications, and c) *infrastructure as a Service (IaaS)* is the utilization of virtual resources for their applications. Various cloud service providers (CSPs) are offering different kinds of services in the market. For instance, **Google** provides storage services (*Google Drive*), *software applications (Play Store Apps)*, and *platforms (Google Apps Engine)* [6]. Similarly, **Microsoft** provides *Azure, Intune, Cloud Platform*; **Amazon** deals with *EC2, AWS*, and many others which are listed-in [30]. The other prominent vendors are *flexiscale.com, salesforce.com Rackspace, and RightScale*. However, CC is not processing large data efficiently at the edge of the network.

Recently, **Fog** computing (FC) environment keeps data and computation close to the end users at the edge of network. FC enhances the elasticity of *resources*, identification of *location awareness* to track end user to facilitate *mobility*. FC also reduces data load on edge, allows shorter latency time, quick response, and maximum throughput in Smart applications like home, vehicle, and Smart grid. The selection of the virtual technologies, managing networks and security are some of the limitations [7, 27].

Similarly, the *Edge Computing (EC)* is a distributed computing paradigm that provides processing at the edge of the network [5], based on two approaches: a) *push* from cloud services that makes computation power on the Internet of Things (IoT) devices to diminish response time and efficient processing power; b) *pull* from IoT which reduces vast data transmissions over a substantial bandwidth networks to the conventional clouds [20]. The applications and advantages of EC can be seen in cloudlet & clonecloud [5].

These technologies allow the *power of processing, information stockpiling, infrastructure's modeling,*

*platforms, and assets services on pay-per-use philosophy*. Services are provided at central as well as at remote ends close to users' devices. Their rapid growth gives rise to a decision making in service selection for its customers [9]. Multiple cloud service providers (CSPs) improve the efficiency for offering services through the Internet [26]. For instance, Google's drive, Microsoft's sky drive, and Amazon's simple storage service are the same **storage** service. AWS, IBM, and Microsoft offer the same **Speech Recognition and different language analysis conversation services**. HP, Google, Microsoft, Amazon, Salesforce, and IBM offer the same *online education and learning services*. Giving this scenario, it is difficult to make a best decision service [3, 6, 18]. The existing platforms use a feedback from respective users in the form of company value [14], applying k-mean clustering on the feedback input by fixing the number of clusters [30], filtration of company value with quality of attributes on user-based constraints [3], and KNN-Fuzzy classification technique [8].

In the case of feedback, the broker or service provider may generate fake information or have biased data [21-25]. Any computation made on this information may not be actual results and therefore, may not present the real face of ranking and evaluation assessment of the services. The information provided to the end user should be of real nature representing actual rankings and assessments.

Quality of service (QoS) is the second issue for making comparisons and evaluation for recommending the best service. The approaches mentioned in [3, 6] defined fixed parameters that led to the discomfort of the users. The list of QoS parameters should be of dynamic nature and should be used independently of selecting these parameters. Similarly, the ranking should be on independent assessments mechanisms. In this regard, the interrelationships between the services (used by the end-users) would play a valuable asset.

To solve above-mentioned issues, we identify the interrelationship between the package services in the form of *Service Association Factor (SAF)* and filters out on a dynamically defined list of QoS parameters like *cost, performance, time and fault tolerance*, for producing efficient and accurate results.

Secondly, in the presence of several available services that provide runtime results on users query over large calculations, the proposed technique is developed as

an agent. The intelligent system works away from the cloud's services on a Fog structure.

The objective is to provide the best CSP and their services through an agent, close to end-user so that they can find effective services as per their desires and of the best quality. The system also maintains previous results generated by other seekers and provides better information to the new entrants as well.

We have validated our proposed methodology on *Amazon*, *Google*, and *Microsoft*. Our principal motivation is to support seekers, to find the best service providers as a desirable solution integrating all positives aspects.

Section 2 explores the existing approaches, advantages, challenges and limitations within the service selection. Section 3 describes the concept of fog agent and Section 4 presents our novel approach named Service Association Factor (SAF). Section 5 explains the structure of Fog Agent for service selection and in Section 6, process is elaborated with an example. Section 7 discusses the obtained results. Section 8 concludes and presents the future research work.

---

## 2. Related Work

Online resources provide rich information about services, but a typical user can be unable to find precise information on urgent basis [22]. Many recommended systems are categorized in [30], based on users' requirements and expectations, employed for certain scenarios where more than one option exists. Business intelligence (E-business applications) becomes crucial to know the end user requirements. Due to the dynamicity and impulsive form of the business applications and the data load on the internet, the provision of quality is a significant challenge [28, 29].

SMICloud [6] provides a framework and a mechanism to index the cloud providers on customer's needs and ranks services based on quality parameters. SMI-Cloud 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.

Simulation Program for Elastic Cloud Infrastructures (SPECI) [23] allows previous information component scaling and parameters of performance for

data centers. These metrics are useful for a recommendation, but lack of post-evaluation features.

Performance factor is an agent-based cloud service selection approach which recommends services on user's feedback on different quality attributes [18]. 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 where historical data [4] are utilized over text-based analysis and applied 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 [28, 29] works on QoS constraints with response time, cost-effectiveness. The problem is modeled by a combinatorial approach as the Multi-choice knapsack problem. Secondly, the issue is resolved with a directed acyclic graph (DAG). This algorithm produces an oversight in the user's interaction. K-mean clustering approach is applied on user's feedback and categorized them into fixed clusters [30]. 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 [24].

Similarly, Trust Evaluation Metric is a fuzzy logic based approach for the trust degree evaluation of CSP's trustworthiness [1]. In this approach, relationship among the services is missing and the degree of trust may vary from user to user. The QoS history technique ranks the services in time slots on user criteria which might lead to worst ranking if user selects incorrect time slots [9]. 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 [26].

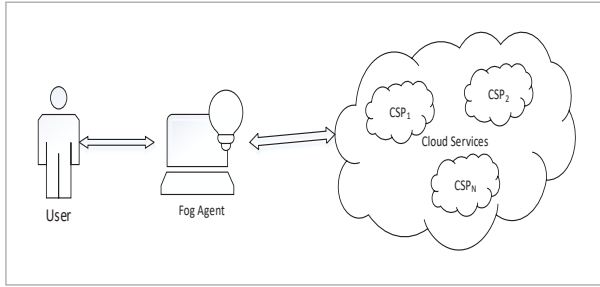
Several, already proposed algorithms are unable to find a proper service selection on user requirements using associations hidden in the services utilization.

---

## 3. Fog Computing Agent

An intelligent agent allows people to delegate their work to it, doing repetitive tasks, for instance, solves problems, computes composite data, permits learning, and enables decision making. The principal character-

**Figure 1**  
Service Selection through Fog Agent



istics of the agent are the autonomy, the communication and cooperation, pro-activity, negotiation, and re-activity. The cloud computing holds a vast scope for machine learning. For automating the process of service selection required by the seekers, an intelligent agent is the best approach to do the task. The joint venture of edge cloud and intelligent agent can play a vital role in Internet-based services-selection to the end users. A Fog Computing (FC) agent is an agent base SaaS that works on a fog environment. FC hides the technical details of clouds that how they work. The working of the FC agent shown in Figure 1 communicates with all the service providers and maintains their respective information for providing a useful service selection.

#### 4. Service Association Factor (SAF)

The frequency of occurrence of two services ( $S_i, S_k$ ) determines the interrelation connection between them. Service Association Factor (SAF) trusts the service used. The association (A) refers to the calculation of all appearances of services ( $S_i, S_k$ ) in the user's package. The process is applied on all packages where both services occurred together and counts as '1' which is irrespective of package's type, Equation (1):

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

The effective *selection* requires to analyze different parameters associated with *internal* and *external* service structure. There are two types of CSPs: a) vendor based, e.g., Google, Amazon or *Microsoft*; and b) broker/agent-based which provides a mix of vendor's services. Let's take package  $P = \{P_1, P_2 \dots P_p\}$  where each  $P_i$  has services  $S = \{S_1, S_2 \dots S_m\}$ . The cloud *client* is a user

(U) who makes requests for acquiring services:  $U = \{U_1, U_2 \dots U_x\}$ . So association **A** for  $S_k$  is calculated in Equation (2):

$$A(S_k) = \sum_{i=1}^n A(S_k, S_i), \tag{2}$$

where  $S_i \in P_i$  and n is total number of packages where  $S_i$  &  $S_k$  occurred together.

The repository  $R_A$  of all association factors between two paired services  $S_i$  &  $S_j$  is described by the following matrix (3):

$$R_A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1j} \\ A_{21} & A_{22} & \dots & A_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ A_{i1} & A_{i2} & \dots & A_{ij} \end{pmatrix}, \tag{3}$$

where  $A_{ij}$  represents the value of association factor for service  $S_i$  w.r.t  $S_j$  as mentioned in Equation (2).

For recommendation, there are two approaches: In the first approach, the previous knowledge is not existing and secondly, the knowledge is available in the form of services  $S_i$  and recommendation service  $S_k$  is made on the maximum value of SAF as defined in Equation (4):

$$A_k = \begin{cases} 0 & \text{if } S_i = \emptyset \\ \max(A_i) & \text{if } S_i = \{S_1, S_2 \dots S_i\} \end{cases} \tag{4}$$

The first option works when agent starts for the first time, or the user's request is not found in the repository, and user selects the service by its own (condition 1 in Equation (4)). However, if past information is present, the agent proposes service/package. In this case, the recommendation of best service  $S_k$  depends on the highest value of SAF (condition 2 of Equation (4)). A recommendation list  $L_R = \{A_1, A_2 \dots A_k\}$  is generated where  $A_k$  represents the service with max value of SAF w.r.t.  $S_k$ .

Further, to provide state-of-the-art services, a quality filtration layer is applied on  $L_R$ . This layer contains all quality parameters, for instance, cost effectiveness, support, response time, robustness, fault-tolerance, security, down time etc. User selects its quality

parameter from a list provided. Let's say, there are  $Q=\{Q_1, Q_2, \dots, Q_n\}$  quality parameters and user desired parameters are  $Q_{u_i}=\{Q_{u_1}, Q_{u_2}, \dots, Q_{u_i}\}$  and  $Q_{u_i} \subseteq Q$ . Each service of  $L_R$  is evaluated on each parameter  $Q_{u_i}$  and final recommendation list of best services  $L_Q$  is prepared by using a filtration function defined in Equation (5):

$$L_Q = \forall_i \forall_j \text{Quality\_Filter}(L_i, Q_j), \quad (5)$$

where *Quality\_Filter* is the quality filtration layer function,  $i$  is total number of services in  $L_R$  to be evaluated and  $j$  is the total number of quality parameters selected by the user.

The *Quality\_Filter* filters out all those services which are not fulfilling the user's quality standards and provides the best services.

The *SAF* approach also allows to recommend a complete package having best services on user's desires. It is defined by the following triplet:

$$(P_k, S_k, A_k), \quad (6)$$

where  $P_k$  is a package 'k' to be recommended having services  $S_k$  along-with association factor  $A_k$  w.r.t to all other services.

Hence, a repository of all services with *SAF* could lead to finding out finest services of one's choice and of best quality indeed.

The user information regarding its requirements is in the prescribed form.

### Customer Request Performa (CRP)

The prescribe form includes assessment and evaluation of the existing information in the repository. The user requires some services with different quality parameters. Therefore, a CRP is developed to seek information in a proper format. The syntax of CRP is:

---

```
(Customer_Name
[, Service_Type {Cloud_Service | Fog_Service |
Edge_Service}]
[, QoS_Array])
```

---

Where *Customer\_Name* is the end user who is requesting the online services. The *Service\_Type* is the type of service as *Cloud\_Service*, *Fog\_Service* or *Edge\_*

*Service*. The QoS parameters are gathered into *QoS\_Array* which has two types of values associated with *{Binary, Range}*. The first type contains *{Yes, No}* and the second contains a range *[Min\_value ... Max\_value]*, as we can see below:

---

```
QoS_Array
([Parameter_Name
[, Parameter_Type {Binary/Range}
[, Parameter_Value {Yes|No,
Min_value|Max_value}])
```

---

This technique allows learning the past behavior of user which package(s) or service(s) it has used. When a new user wants to use this system, with the help of *SAF*, it finds the best available services in the market. The user starts with one service, say  $S_1$ , and after the system recommends those services which have *SAF* values with  $S_1$ , in ascending order. So, one can opt the best service.

## 5. The Structure of Fog Agent

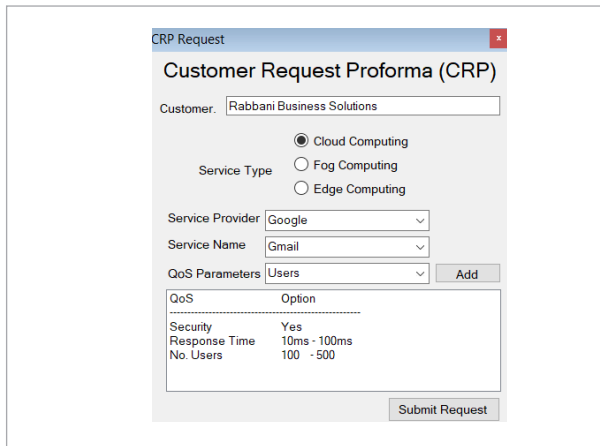
The Fog layer of computing services provides the facility to develop our proposed system. The process of service *evaluation*, *assessment*, and *recommendation* is done on the Fog agent that serves as a web service. It communicates with all CSPs and maintains their respective information in its repository for providing effective services on demand.

The user interacts with the system through its interface and submits CRP form. Services available in the repository are evaluated on said constraints defined in CRP and filtered accordingly. In the second step, *SAF* is produced for all respective services which are going to be recommended. The components of the system are *User Interface*, *Service Selector*, *Association Generator*, *Quality Filter* and *Service Recommender* (see Figure 3).

### 5.1. User Interface

The communication with the system is through the interface of the Fog agent. This module provides CRP to its users for collecting requirement. This component also allows registration to CSPs and services regarding their functionalities and adopted mechanism. The final recommended services are also provided through this interface as shown in Figure 2.

**Figure 2**  
Customer's interaction on CRP



**5.2. Service Selector**

This component is responsible for finding the services matched with user criterion. It receives requirements from user interface module and discovers services/

CSPs from the repository which are fulfilling the requirements. The association generator component gets the recommended information.

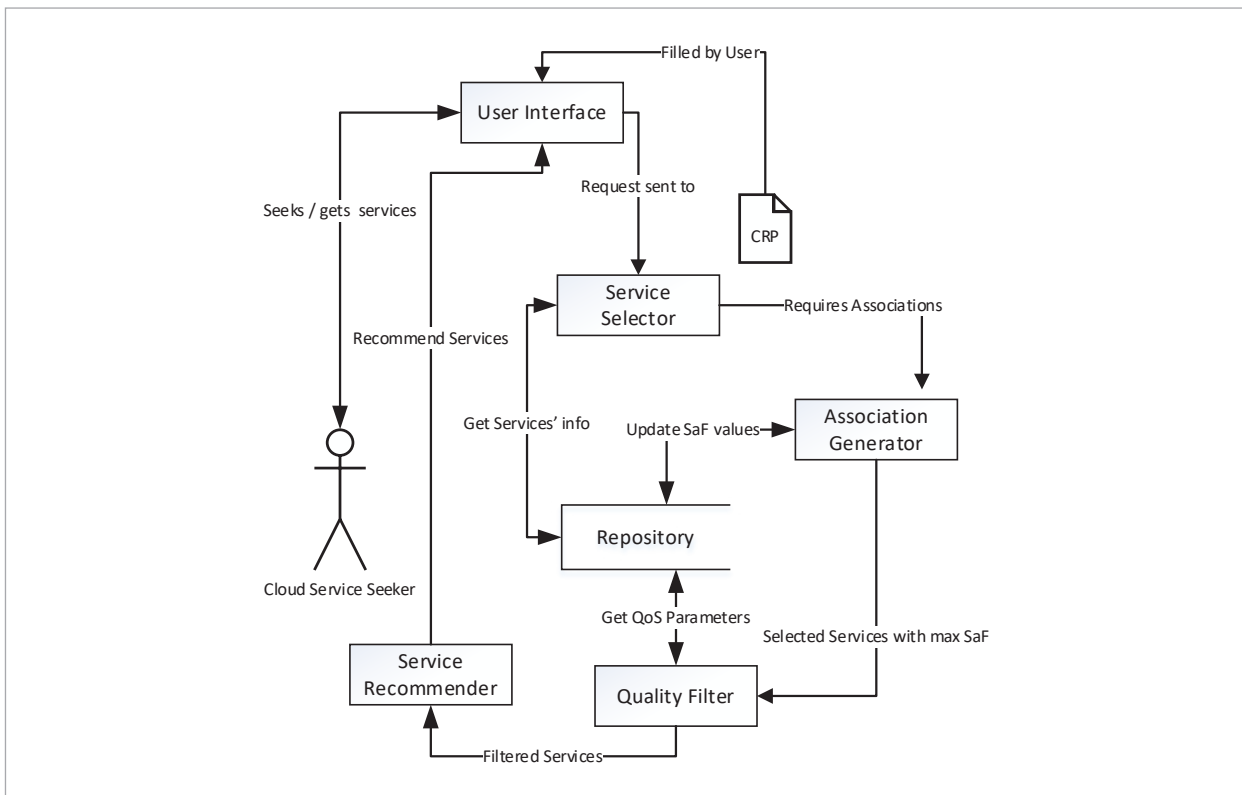
**5.3. Association Generator**

The most important part of the agent that identifies the value of interrelationship in the form of *SAF* between the services available in the repository is association generator. In the second step, it placed *SAF* values in ascending order so that most appropriate service is recommended first, having max *SAF* value. The higher *SAF*'s services are sent to the next unit of the quality filter where further evaluation will be processed.

**5.4. Quality Filter**

The service evaluation of quality parameters is the task of the Quality filter defined in *QoS\_Array* of CRP at the time of submission. *QoS\_Array* filters every service marked by the Association Generator and qualified services are sent to next unit of service recommender as shown in Figure 3.

**Figure 3**  
Process of service selection using SAF



## 5.5. Service Recommender

The system holds the repository of usage history in user's/vendor's perspective, named *Cloud Package's Repository*. On the request made by the user, *SAF* calculates, stores, and submits all related services for the recommendation.

## 6. A SAF Calculation Example

Google, Microsoft & Amazon are the three well known CSPs offering different services. There are five packages, used by the seekers and the information is available in the Fog's repository as shown in Table 1.

Giving  $P_i$  package, all mutual occurrences of  $S_i$  w.r.t other  $S_j$  are added into the cell  $(S_i, S_j)$  according to Equation. (4) (See Table 2). This process is repeated for all 'i' from 1 to the maximum number of services in the repository. For instance, the association between  $S_5$  and  $S_8$  is counted as 4 and represented in the cell (5, 7), and similarly,  $S_3$  and  $S_5$  is also 4 in the cell (3, 5) where Cell (x, y) represents as:

x = row number and y = column number

**Table 1**

Services' Description in Packages

Service	Description	Packages
$S_1$	Google search	$P_1, P_4$
$S_2$	Windows Azure	$P_1, P_3, P_5$
$S_3$	Gmail	$P_1, P_2, P_3, P_4$
$S_4$	Google Drive	$P_1$
$S_5$	Amazon Elastic Compute Cloud	$P_1, P_2, P_3, P_4, P_5$
$S_7$	Amazon Cloud Search	$P_2$
$S_8$	Amazon Simple Storage Service	$P_2, P_3, P_4, P_5$
$S_9$	Amazon SES	$P_5$
$S_{10}$	Bing	$P_3$
$S_{13}$	Chrome OS	$P_4$
$S_{15}$	Send Grid Email Service	$P_5$

**Table 2**

Determination of Service Association Factors among services

Services	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{13}$	$S_{15}$
$S_1$	-	1	2	1	2	0	1	0	0	1	1
$S_2$	1	-	2	1	3	0	2	1	1	1	0
$S_3$	2	2	-	1	4	1	3	1	1	1	1
$S_4$	1	1	1	-	1	0	0	0	0	0	0
$S_5$	2	3	4	1	-	1	4	2	1	2	1
$S_7$	0	0	1	0	1	-	1	1	0	0	0
$S_8$	1	2	3	0	4	1	-	2	1	2	1
$S_9$	0	1	1	0	2	1	2	-	0	1	0
$S_{10}$	0	1	1	0	1	0	1	0	-	0	0
$S_{13}$	1	1	1	0	2	0	2	1	0	-	1
$S_{15}$	1	0	1	0	1	0	1	0	0	1	-

## 7. A Comparison and Performance Evaluation

In this section, we evaluate the strength of our proposed SAF approach with others, based on feedback, service scores, and rankings. Table 3 shows the key evaluation parameters which are necessary to assess the service selection policies available. Zain et al. [30], Saurabh et al. [6], and Abid et al. [13] are using feedback datasets which may have biasness and does not generate accurate results. Similarly, Sahar et al. [19] and Abid et al. [13] have used specific QoS parameters for their evaluation and missing other parameters. Arezoo et al. [10], Saurabh et al. [6], Parhi et al. [17], and Sahar et al. [19] provide selection for one type of

services only. Zain et al. [30], Abid et al. [13], and Ibrahim et al. [2], all are providing selection for all types of services but have different limitations as mentioned in Table 3. Therefore, in view of the evaluation assessment, we infer that the proposed technique is more efficient and reliable in terms of feedback biasness, fixation of parameters and types of services. SAF is independent of these issues and represents real face of rankings.

From example, as mentioned in previous section, it is derived that the Amazon Elastic Compute Cloud ser-

**Table 3**

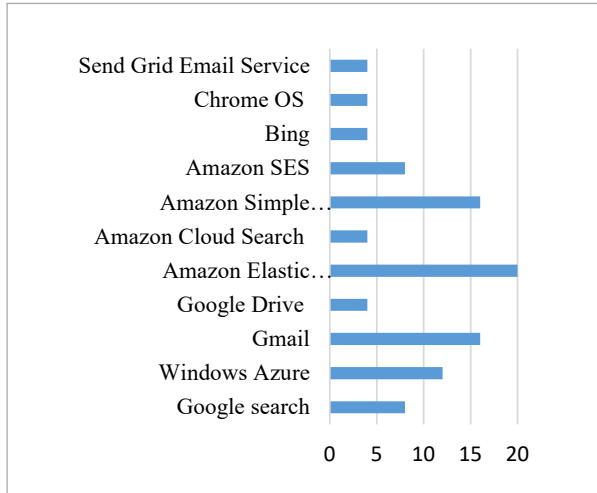
A comparison of approaches on the basis of evaluation parameters

Approaches / Evaluation Parameters	Users' Criteria	QoS Parameters' Evaluation	Data Set for Evaluation	Technology	Evaluation Scope	Constraints
Zain et al. [30]	Tabular Form	Fixed	Feedback	Clustering	SaaS, PaaS, IaaS	Fix Number of Clusters
Arezoo et al. [10]	Tabular Form	SMI-Attributes	Feedback	Multi-Agent System	SaaS	User given scores
Saurabh et al. [6]	Yes	Fixed	Feedback in Vectors	AHP	IaaS	Missing inter-relationships
Parhi et al. [17]	Formal	Dynamic	Semantic ontology	Inference Rules	IaaS	Not cover all types of services
Sahar et al. [19]	Tabular Form	Performance base parameters	Service Ranks	Relative Service Ranking Vector & SMICloud Toolkit	SaaS	Not cover other important QoS parameters
Abid et al. [13]	Structured in GUI	Trust Parameters	Feedback & Past Experience	MAS + K-mean Clustering	SaaS, PaaS, IaaS	Lack inter-relationships between the services
Ibrahim et al. [2]	Tabular Form	Fixed	Score matrix	MCDM	SaaS, PaaS, IaaS	Rank depends on scores and lack of important quality parameters
SAF	Structured in the Form of CRP	Dynamic	Historical Data / Service Utilization	Multi-agents System	SaaS, PaaS, IaaS	Independent of fixed, type defined values



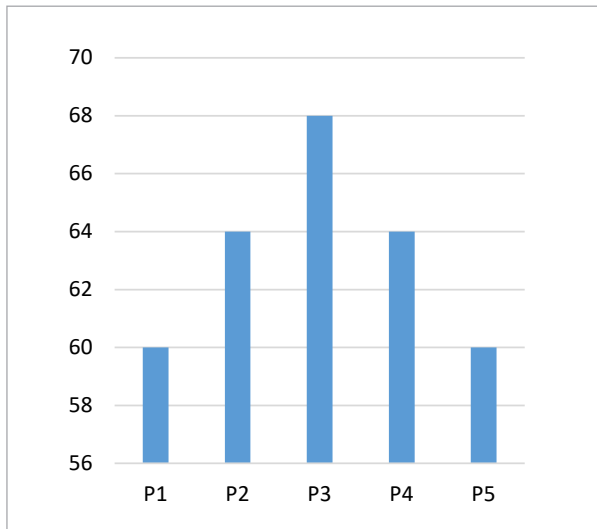
vice gets the highest value of SAF (sum of values) and shows that it is the most liking service with profit of 20 (Figure 4(a)).

**Figure 4(a)**  
Services' Score using SAF



In addition, the SAF approach identifies the best package (sum of all SAF values) as described in Figure 4(b) where package P<sub>3</sub> gets the maximum value of SAF and stands first on user's desires with profit of 68. Thus, the ranking of online services, as well as their providers can also be obtained using SAF.

**Figure 4(b)**  
Packages' Score using SAF



Further, we tested SAF with two other approaches on three different CSPs with their services for 20-user data set as mentioned in Table 4. The results are also shown in Table 4 as well.

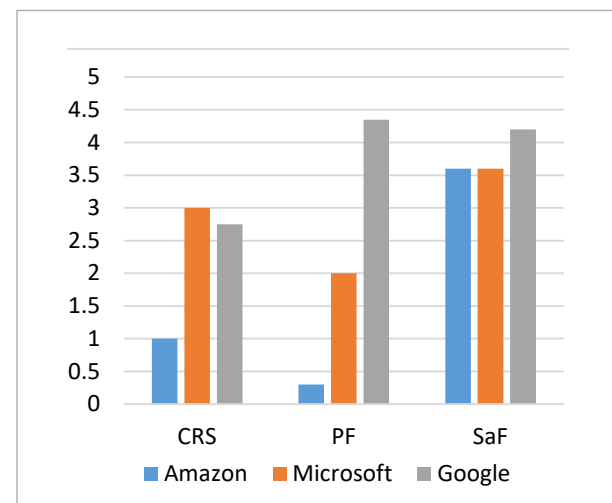
**Table 4**  
Comparison of Approaches for Rankings

Methods	Amazon	Microsoft	Google
Service Association Factor (SAF)	3.6	3.6	4.2
Performance Factor (PF) [18]	0.38	2	4.35
Cloud Recommender System (CRS) [30]	1	3	2.75

The following conclusions are drawn.

- The techniques in [18] and [30] are dependent on the user's feedback while SAF is independent of all fixed and defined values.
- Using standard deviation, Amazon is at lower point as compared to other two vendors as shown in Figure 4(c).
- Google is best among the rest in all three approaches with higher ranking value and,
- SAF is the most consistent among all providers with the least standard deviation. The inconsistency in other two techniques might be due to the biasness and self-generated feedback collected by the vendors.

**Figure 4(c)**  
Impact of recommendation approaches on service providers



## 8. Conclusion

Existing techniques for cloud service selection have certain limitations like feedback biasness and self-generation, QoS parameters, fixed values, coupled, and single type services. The novel approach of Fog agent by implementing *SAF* resolves these issues. It calculates occurrences of service's interrelationship in a package, and is independent of fixation type. The system assesses and evaluates the services on quality parameters, and criteria demand in CRP. Buyers get best facilities as per their needs. The results

show that the proposed system and technique works well in said scenarios. The system is helpful for on-line community/QoS brokers to make effective online decisions. These decisions are useful to find online education courses, business applications, medical assistance, hotel reservations etc.

In future, we intend to implement *SAF* technique on vendors' side helping them in redesigning their services group for capturing better 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. Al-Jabri, Ibrahim, M., Mustafa, I., Sadiq S. M. A Group Decision-Making Method for Selecting Cloud Computing Service Model. *International Journal of Advanced Computer Science and Applications*, 2018, 9(1), 449-456. <https://doi.org/10.14569/IJACSA.2018.090162>
3. Aqsa, A., Imran, M. R., Martinez, E., Aslam, M. A User-Centered Approach for Cloud Service Selection and Recommendation. *Research in Computing Science*, 2016, 130(2016), 99-110.
4. 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>
5. Francis, T., Madhiajagan M. A Comparison of Cloud Execution Mechanisms: Fog, Edge and Clone Cloud Computing. *Proceeding of the Electrical Engineering Computer Science and Informatics*, 2017, 4, 446-450.
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. Hao, Z., Novak, E., Yi, S., Li, Q. Challenges and Software Architecture for Fog Computing. *IEEE Internet Computing*, 2017, 21(2), 44-53. <https://doi.org/10.1109/MIC.2017.26>
8. Humaira, N., Imran, M. R., Aslam, M., Martinez, E. KNN-Fuzzy Classification for Cloud Service Selection. *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems – ICFNDS18*, 2018. <https://doi.org/10.1145/3231053.3231133>
9. 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>
10. Jahani, A., Derakhshan, F., Leyli, M. K. ARank: A Multi-Agent Based Approach for Ranking of Cloud Computing Services. *Scalable Computing: Practice and Experience*, 2017, 18(2), 105-116. <https://doi.org/10.12694/scpe.v18i2.1283>
11. 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>
12. 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>
13. Mahmood, A., Shoaib, U., Shahzad, 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>
14. 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>
15. 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.
16. Online Reviews and Endorsements. (2015). Report on the CMA's call for information. [online] Competitions & Markets Authority (CMA). Available at: [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/436238/Online\\_reviews\\_and\\_endorsements.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/436238/Online_reviews_and_endorsements.pdf) [Accessed 11 Oct. 2018].
  17. Parhi, M., Pattanayak, B. K., Patra, M. R. A Multi-Agent-Based Framework for Cloud Service Discovery and Selection Using Ontology. *Service Oriented Computing and Applications*, 2017, 12(2), 137-154. <https://doi.org/10.1007/s11761-017-0224-y>
  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](https://doi.org/10.1007/978-3-642-37371-8_14)
  19. Sahar, A. E., Yousif, A., Bashir, M. B. Performance Based Ranking Model for Cloud SaaS Services. *International Journal of Information Technology and Computer Science (IJITCS)*, 2017, 9(1), 65-71. <https://doi.org/10.5815/ijitcs.2017.01.08>
  20. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 2016, 3(5), 637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
  21. Shojae, S., Azman, A., Murad, M., Sharef, N., Sulaiman, N. A Framework for Fake Review Annotation. *Proceedings of IEEE 17th UKSIM-AMSS International Conference on Modelling and Simulation*, 2015, 153-158.
  22. 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>
  23. 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](https://doi.org/10.1007/978-3-642-10665-1_35)
  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. Valant, J. The Case of Misleading or Fake Reviews. *Online Consumer Reviews. European Parliamentary Research Service*, 2015. PE 571.301
  26. Yeh, K.-H. An efficient Resource Allocation Framework for Cloud Federations. *Information Technology and Control*, 2015, 44(1), 64-76. <https://doi.org/10.5755/j01.itc.44.1.6875>
  27. Yi, S., Hao, Z., Qin, Z., Li, Q. Fog Computing: Platform and Applications. *Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 2015, 73-78. <https://doi.org/10.1109/HotWeb.2015.22>
  28. 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>
  29. 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). <https://doi.org/10.1145/1232722.1232728>
  30. 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>