SAFRank: Multi-Agent based Approach for Internet Services Selection

Imran Mujaddid Rabbani
Department of Computer Science,
University of Engineering and Technology, Pakistan
imranmrabbani@gmail.com

Muhammad Aslam
Department of Computer Science,
University of Engineering and Technology, Pakistan
maslam@uet.edu.pk

Ana Maria Martinez-Enriquez
Department of Computer Science,
CINVESTAV-IPN, Mexico
ammartin@cinvestav.mx

Abstract: *In the era of modern world, organization are preferring to adopt smart solutions for their business tasks and managing huge and complex transactions. These solutions are provided through online application infrastructures of Internet of Things (IoT), cloud, fog, and edge computing. In the presence of numerous prospects, the selection benchmark for such offers becomes vibrant, especially, when there is no supportive platform available. Prevailing approaches provide services by evaluating the quality of service parameters, K-Nearest Neighbours (KNN) classifications, k-mean clustering, assigning scores, trustworthiness and fuzzy logic techniques on customer's feedback. However, these approaches classically depend on seeker' feedback and do 'not consider interrelationship between the services. Secondly, these techniques do not follow standards derived by well-known organizations like National Institute of Standards and Technology (NIST), International Organization for Standards (ISO), and IEEE. Feedback may be self-generated or biased and leading to inappropriate recommendation to end users. To resolve the issue, we propose multi agent based approach using service association factor that computes interrelationship values among services appearing together in a package as SAFRank and evaluates it on standards along with dynamically defined quality of service parameters. It assists seekers to select the best services on their preferences from pool of IoT and internet services. The technique is tested on leading cloud vendors and results show that it meets the desires of service seekers in all service models in an efficient manner.*

Keywords: *Internet services, service selection, service association factor, IoT services.*

Received April 10, 2020; accepted September 16, 2021
https://doi.org/10.34028/iajit/19/3/2

1. Introduction

Present day innovations in computer industry are Cloud Computing (CC), Fog Computing (FC), Edge Computing (EC), and Internet of Things (IoT).

CC offers a number of advantages over traditional computing environment like strategic edge, Quality of Service (QoS), performance, fault tolerance, management, reliability, and above all cost sparing. Likewise, distributed computing clients think that it is more advantageous to get to their information from anywhere paying little heed to the machine and place, as information is placed at centre point [1, 2, 11]. CC is not handling substantial information proficiently especially at the edge of the system's network.

FC keeps information and processing near to the clients at the edge of network. FC improves the flexibility of assets, distinguishing proof of area attention to track end clients to encourage portability, and furthermore lessen information stack on the edge. FC permits shorter potential time, snappy reaction and greatest throughput, e.g., smart homes, shrewd vehicles, and perceptive networks. The decision of virtual innovation, overseeing systems, and security are some its restrictions [4, 9].

EC is a distributed computing that brings computer data storage closer to the user [7]. EC allows two approaches:

a) Push from cloud services makes computation power on the IoT devices to reduce response time and efficient processing power.
b) Pull from IoT allows to reduce huge data transmission over heavy bandwidth networks to the conventional clouds [7]. The advantage of EC is elaborated in cloudlet (bring the closer cloud data center for mobile applications) and clone cloud [21].

IoT is making the world smarter by connecting all the things together for sharing data. It refers to all objects that can be identified by an IP address and have the ability to transfer the data over a network without the human interaction. Some services of IoT are Health-care, home-automation, transportation, smart cars, environmental monitoring, and smart E-Grid stations. These technologies allow computer performance measured in terms of accuracy, efficiency, and speed of execution to bring the power of processing, information stockpiling, modelling of infrastructure, platforms, and assets services on the philosophy of pay-per-use at the central or remote end close to users' devices. The rapid growth of computing service creates a challenge in the selection process of customer's service, to entail a decision-making problem [10, 12, 15].

This is the fundamental reason that all the cloud specialist co-ops are in battle to offer the best service, in certainty, they are hustling against time to improve it gradually.

Various services are available from well-known providers, e.g. Google, Microsoft, and Amazon. The majority of these services are storage, computation power, email, platforms. But more than one vendor provides them. To face up to find the best service among all, it requires evaluation mechanism like ranking of services and providers. Besides, the services are provided in a package/group/bundle and user has to take a complete one that increases the cost, as well as other concerns. For instance, Google provides Gmail, Google Drive, and Chrome OS. When a user does not want all, he/she has to take the whole package, even if he/she requires some of them. So, the seeker demands for individual or package-less services. Services can be provided with the support of an agent.

The recommended systems are built under different approaches to ease the process of service selection. Some systems used feedback analysis with the help of performance factor, clustering, QoS parameter filtration. The K-Nearest Neighbours (KNN) fuzzy logics, financial base metrics, cloud workflow based constraints resolve the issue for providing best services as per user demands. Similarly, Service Measurement Index (SMI-Index), Cloud Service Provider (CSP) Index and trustworthiness are also utilized.

All these approaches work well in some scenarios, but on the same time they are not representing factual interrelationships between the services being utilized which can be a handy tool for service recommendation. The feedback can be biased, fake and self-generated [17, 22, 27] and hence so, may mislead the selection process.

We propose a novel approach named SAFRank. Service Association Factor (SAF) [18] calculates the binding force in the form of interrelationships between the services used by the customer and presents precise picture of services utilization. It is an extension of SAF in which recommendation provided on single service request using interrelationship calculation and remains silent when user wants recommendation on more than two services. SAFRank provides recommendation on ‘n’ service requests, produces SAF values among all services in the repository, and generates services’ ranks as well as vendors’ ranks accordingly. It allows user to create its own package depending upon requirements. The approach also allows new service providers to register themselves with the system and view their ranking from the perspectives of users.

Secondly, SAFRank is Multi-Agent System (MAS) technique for performing evaluation and assessments for recommending appropriate services. The goal is to provide the best quality based CSP and their services through MAS as per desires of end user. SAFRank provides better information to the new entrants as well for selecting individual service, a complete package and option to make its own package.

Further growth of IoT services are increasing day by day and in the presence of numerous smart services, it becomes hard to select the appropriate service that leads to the best suitable for one’s desires. SAFRank will help for making accurate decisions for the end users. The validation of SAFRank approach is guiding by service providers: Google, Amazon, and Microsoft.

Study’s primary motivation is to assist seekers for discovering the best service and its providers as an appropriate answer incorporating all positives features. Secondly, goal is to provide the best quality based CSP and their services through MAS as per desires of end user.

Section 2 explores the related work done so far. Selection 3 describes the concept of multiple service selection using SAF. Section 4 explains the structure of Multi Agent of SAFRank. Results are discussed in Section 5. Step by step recommendation process with an example is mentioned in section 6. Section 7 concludes and draw the future research work.

2. Related Work

Online resources provide rich information about services, but a common user can be unable to find precise information on urgent basis [23]. Many recommended systems are categorized in [32], based on users’ requirements and expectations, employed for certain scenarios where more than one option exists. Intelligent Business applications (e-business) becomes crucial to know the end user requirements. Due to the dynamicity and impulsive form of the business applications and the data load on Internet, the provision of quality is a significant challenge [30, 31]. The related work is divided into different sections as follows.

To cope with the said problems, a cloud services recommender system [32] optimizes the selection of services based on data mining technique of K-means clustering. The services are classified into different clusters. Each cluster depicts the quality rank of the services and are evaluated on the user’s feedback.

CSRecommender [33] consisting of five main components: crawler, cloud service identifier, indexer, search engine and the recommender. Crawler identifies the undiscovered Web pages to input into the cloud service identifier. It caches a copy of every Web page visited to prevent repetitive downloading of pages. By examining the home page of Cloud service identifier, it assigns to a cloud service a scoreboard to keep a check whether it is or not a service. Indexer saves a Web service into the recommender database when it is. A unique ID identifies each service. The Search engine allows user to find cloud services easily based on a search term.

Similarly, MAS [13] applies K-mean clustering for evaluation and assessment. It is useful for all types of service models of Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). The system also utilizes trust parameters for enhancing its recommendation list of services to esteemed clients.

CloudRank [25] uses adoptive quality of service management technique for mobile devices. The system performed QoS assessment on service performance monitoring through user's feedback.

The recommender system based on collaborative filtering in [29] uses fuzzy formal concept analysis. This approach based on lattice theory provides dataset of active users, similar user ratings, and top services.

Performance base ranking model suggested by Sahar [5] for selecting SaaS applications. It applies relative service ranking vector and SMICloud [8] toolkit and uses performance based quality parameters to assess the feedback gathered in tabular form. It produces service ranks for selecting best services. However, it is not covering other important QoS parameters during assessment.

A ranking-oriented prediction method [24] assists the selection of the cloud service, which has the highest customer satisfaction. This framework consists of two functions for ranking similarity estimation and a cloud service ranking prediction that consider preferences and expectations of customers. The methodology has defined an enhanced Kendall Rank Correlation Coefficient (KRCC), which reduces the influence of negative customers in the ranking similarity calculation.

Jahani *et al*. ARank [10] based on MAS ranks the cloud services and uses SMI-Attributes by seeking feedback from the existing users who scored each attribute.

Similarly, another MAS approach service selection uses semantic ontologies for the evluation of parameters and dynamically assessed [17]. Its basic target is to provide IaaS architectures. Based on inference rules acquires users' criteria in formal methods.

Performance factor [19] is an agent-based approach, which recommends services on user's feedback using quality attributes. The constant company value is used to compare the feedback results. The value should not be imposing from the vendor's perspective. Intermediary Service Agent Model (ISAM) [28] based on MAS for the dynamic selection of cloud services uses adaptive learning for recommending services. ISAM provides an optimized service selection using incentive and forgetting functions in processing.

A Brokerage-Based approach [6], identifies attributes to rank the cloud services. B+ index database stores the attributes. Users select the attributes and then assign weights to each attribute. Broker Cloud service selection presented in [14, 26] defines a set of attributes to analyze and assessed different cloud service brokers. The evaluation was made with multiple service models in a cloud-computing environment. The main goal behind the definition of attributes was to facilitate the service discovery for its customers to select effective services as per their requirements and desires.

The cloud service broker in [7] integrates different modules:

a) Service Discovery.
b) Matching and Ranking.
c) Decision-Support.
d) Service Normalization.
e) Service Monitoring.
f) Evaluation. The system also has a cloud service repository for the provision of effective recommendation.

SAF [18] recommends services and CSPs on rankings. It performs quality of service filtration layered on each service before recommending to the seeker. SAF presents advantage over other approaches, because it calculates the interrelationships rather than the user's feedback. SAF recommends services based on a single service request without considers multiple services for the recommendation process.

As of best of our knowledge, the proposed methodology of SAF values based on service interrelationships is first of its kind, for the evaluation of services and CSPs recommendation using ranks.

3. Multiple Service Selection based on SAF

Our problem is defined as follows:
Given:

- A set of Services $S = \{S_1, S_2, \dots S_n\}$, where the type T of S_i can be T= {CC, FC, EC, IoT},
- A set of Packages $P= \{P_1, P_2 \dots P_n\}$, where P_j can be {company, user}, and each $P_i= \{S_j, \dots S_k\}$ is a subset of S, such that $P_i \subseteq S$ where $| P_i | > 0$ and $P_i \in P_{ui}$ **V** P_{ci}
- Quality Parameters $Q = \{Q_1, Q_2, \dots, Q_n\}$,

The goal is to satisfy a user request catered in a set

$$U = \{S_1, S_2 \dots S_u\}$$

Where

$$U \subseteq S \text{ and } |U| > 0,$$

And the requested quality is

$$Qu = \{Q1, Q2 \dots Qu\},$$

Where

$$Q_u \subseteq Q$$

To achieve this goal, we calculate the Service Association Factor (SAF) which considers the interrelationship between two services (S_i, S_k) based on the frequency of occurrence of them, Equation (1).

$$A_{ik} = \sum_{k=1}^{n} A(S_i, S_k) \tag{1}$$

Where A_{ik} represents the association of S_i with S_k and $S_k = S - S_i$. SAF values are calculated with other than S_i.

Here $S_i \in P_i$ and n is the total number of packages where S_i and S_k occurred together. S_i is a set of services that allows user to request more than one service in its request and the system generates and evaluates SAFs accordingly.

The repository R_A is a matrix of all associations of services S_i and S_k;

$$R_A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{ik} \end{pmatrix} \tag{2}$$

Where A_{ik} represents the value of the association factor for service S_i, w.r.t S_k as mentioned in Equation (1). If specific requirement is provided in S_i (more than one service at a time for seeking recommendation) than only one row-based repository is calculated in the form of the following set:

$$R_i = \{S_{i1}, S_{i2}, \ldots S_{ik}\} \tag{3}$$

Where R_i is the set of all service S_k associated with S_i

- *Ranking of Service*: To attain service ranking, all the associations of service S_i are summed together with Equation (4).

$$S_{RF_i} = \sum_{k=1}^{k} A_{ik} \tag{4}$$

Where S_{RF} represents the rank factor of i^{th} service w.r.t '*k*'.

- *Ranking of Package*: is the average service's rank value of services in a given package. Equation (5) calculates it.

$$P_{RF} = \frac{\sum_{i=1}^{n} S_{RF_i}}{n} \tag{5}$$

Where P_{RF} represents the rank factor of Package and the total number of services in '*P*' is '*n*'.

R_i in Equation (3) is the generated recommendation list. Further, quality filtration layer applied on R_i to provide the state-of-the-art services. This layer contains all quality parameters, for instance, cost effectiveness, support, response time, robustness, fault-tolerance, security, down time defined in Q_u as:

$$R_Q = \forall_i \forall_u \text{ Quality\_Filter}(R_i, Q_u) \tag{6}$$

Where *Quality_Filter* is the quality filtration layer function, *i* is the total number of services in R_i to be evaluated and *u* in R_u is the total number of quality parameters selected by the user. The *Quality_Filter* function filters out all those services, which are not meeting the user's quality standards.

So, if a repository of all services with *SAF* is maintained then one can seek services of its choice by sending request in a set S_i and SAFRank recommends services having *SAF* values with S_i, in ascending order for appropriate selection.

This technique allows learning the past behavior of user, which package or service it has used. When a new entrant wants to avail the opportunity, with the support of *SAFRank*, it finds the best available services in the market.

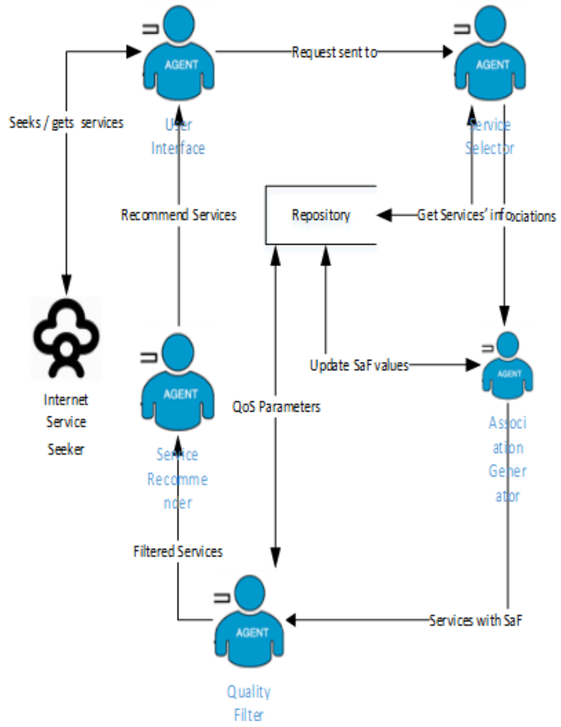

Figure 1. Multi-agent structure.

4. The Structure of Multi Agent System of SAFRank

The process of service evaluation, assessment, and recommendation is based on MAS. The user interacts with the system through its interface agent and submits its request in structured form. Services available in the repository are evaluated on said constraints and filtered accordingly. In the second step, SAFRank is produced for all respective services, which are going to be recommended. The agents of the system are User Interface Agent, Service Selector Agent, Association Generator Agent, Quality Filter Agent and Service Recommender Agent, defined as follow:

- *User Interface Agent*: allows the user's communication with the system. It provides request form to its users for collecting requirement. This agent also allows registration to different CSPs and services regarding their functionalities and the adopted mechanism. The end user can also register to seek information on time-to-time basis. This interface provides the final recommended services as shown in Figure 1.

- *Service Selector Agent*: is responsible for finding the services matched with user criterion. It receives requirements from the user interface agent and discovers services/CSPs from the repository, which are fulfilling the said requirements.
- *Association Generator Agent*: is the most important part of MAS, identifies the value of interrelationship in the form of SAF between the services available in the repository. Further, it places SAF values in ascending order so that most appropriate service is recommended first, having max SAFRank value.
- *Quality Filter Agent*: uses defined QoS parameters submitted at the time of request. This agent filters each service marked by the Association Generator Agent and sends the qualified services to the recommender agent as shown in Figure 1.
- *Service Recommender Agent*: holds the repository of usage history in user's/vendor's perspective and their respective SAFRanks along-with SAF values, named as Cloud Package's Repository. On the request made by the user, SAFRanks are shown to the users for recommendations.

5. Results and Discussions

The ten different services from Google, Microsoft and Amazon are available in the repository of the system in five different packages as shown in Table 1.

Table 1. Services' description in packages.

<table>
<tr><th>Service</th><th>Description</th><th>Packages</th></tr>
<tr><td>S_1</td><td>Google search</td><td rowspan="10">$P_1 = \{S_1, S_2, S_3, S_4, S_5\}$
$P_2 = \{S_3, S_5, S_6, S_7\}$
$P_3 = \{S_2, S_3, S_5, S_7, S_{10}\}$
$P_4 = \{S_1, S_3, S_5, S_7, S_9\}$
$P_5 = \{S_2, S_5, S_7, S_8\}$</td></tr>
<tr><td>S_2</td><td>Windows Azure</td></tr>
<tr><td>S_3</td><td>Gmail</td></tr>
<tr><td>S_4</td><td>Google Drive</td></tr>
<tr><td>S_5</td><td>Amazon Elastic Compute Cloud</td></tr>
<tr><td>S_6</td><td>Amazon Cloud Search</td></tr>
<tr><td>S_7</td><td>Amazon Simple Storage Service</td></tr>
<tr><td>S_8</td><td>Amazon SES</td></tr>
<tr><td>S_9</td><td>Bing</td></tr>
<tr><td>S_{10}</td><td>Chrome OS</td></tr>
</table>

Giving P_i package, all mutual occurrences of S_i w.r.t other S_j are summed into the cell (S_i, S_j) according to Equation (1) (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 2 shows SAF of all services coupled in above mentioned packages. It is mentioned here that self-associations are not calculated and are shown as '-' in diagonal cells of Table 2. It is also seen that service S_3 and S_5 have the maximum association value of '4' and similarly S_5 and S_7 have. The service S_4, S_6, S_9, and S_{10} have minimum associations with respect to others.

Table 2. Calculation of SAF value.

Services	S_1	S_2	S_3	S_4	S_5	S_7	S_8	S_9	S_{10}	S_{13}
S_1	-	1	2	1	2	0	1	0	1	0
S_2	1	-	2	1	3	0	2	1	0	1
S_3	2	2	-	1	**4**	1	3	0	1	1
S_4	1	1	1	-	1	0	0	0	0	0
S_5	2	3	**4**	1	-	1	**4**	1	1	1
S_6	0	0	1	0	1	-	1	0	0	0
S_7	1	2	3	0	**4**	1	-	1	1	1
S_8	0	1	0	0	1	0	1	-	0	0
S_9	1	0	1	0	1	0	1	0	-	0
S_{10}	0	1	1	0	1	0	1	0	0	-

The Service and Package's Ranking is performed using Equations (4) and (5) and shown in Figures 2 and 3 respectively. It is seen that package P_2 gets the highest ranking and package P_1 gets the lowest ranking with respect to rest of the packages.

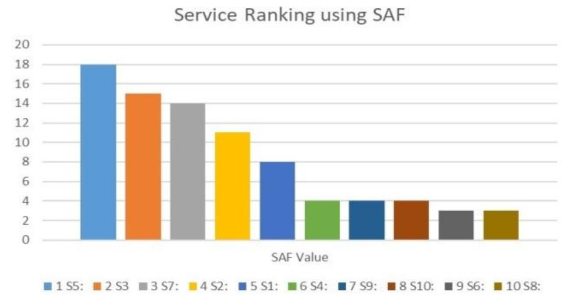

Figure 2. Service ranking using SAFRank.

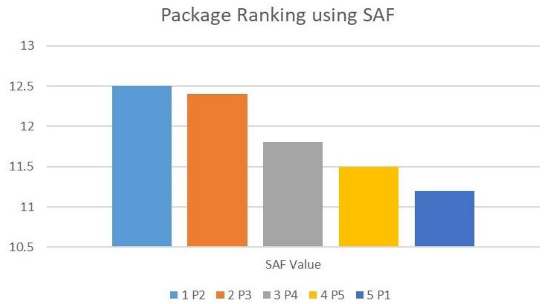

Figure 3. Package ranking using SAFRank.

6. Recommendation of Services Using SAFRank

It is an iterative process; the approach selects one service in one iteration for recommendation and continue until user wants. To recommend the services on user's requirements, let's select top two services from ranking list as shown in Figure 3. And group them in set $S_i=\{S_3, S_5\}$. The SAF values are produced on S_i shown in Table 3.

Here, only two iterations of R_i are performed, on given packages P_1–P_5, in the first iteration, the results are shown for $S_i=\{S_3, S_5\}$ and in the second, results are for $S_i=\{S_3, S_5, S_7\}$. Tables 3 and 4 shows the *SAFRanks* of each iteration respectively.

The iterations can be more, depending upon the availability of the services. It is also possible that we can restrict the iterations on user's request by applying iterative deepening search technique. It is a learning based system to evaluate and recommend services as and when required. It populates its repository with creation of every new package, registered by the new vendor or create by the end user itself.

Table 3. Calculations of SAFRank based on user desires ‘S_i’.

Iteration	**I^st for R_i**	**2^nd for R_i**
S_i	$R_i = \{S_3, S_5\}$	$R_i = \{S_3, S_5, S_7\}$
S_1	2	1
S_2	2	1
S_4	1	0
S_6	1	1
S_7	3	X
S_8	0	0
S_9	1	1
S_{10}	1	1

Table 4. Ranking of services.

Rank	**Service**	**SAF Value**	**Service**	**SAF Value**
on $S_i = \{S_3, S_5\}$ (1st Iteration)			$S_i = \{S_3, S_5, S_7\}$ (2nd Iteration)	
1	S7	3	S_1	1
2	S1	2	S_2	1
3	S2	2	S_6	1
4	S4	1	S_9	1
5	S6	1	S_{10}	1
6	S9	1	S_4	0
7	S10	1	S_8	0
8	S8	0		

Figure 4 shows the recommendation process of a targeted service. Where after choosing the service, all the associated services are ordered ascendingly with respective SAF values. User starts with selecting service S_1, the system calculates SAF values and after performing quality filter check, recommends services which have association with S_1, in ascending order. On top, services are listed with higher SAF value and helping in choosing the best service accordingly. In the next step, if a user selects S_3, then query of $\{S_1, S_3\}$ will be sent to system for further recommendations. Similarly, in step 3, user selects S_2 and query will be $\{S_1, S_3, S_2\}$. In step 4, if user does not prefer to select top one and goes for 3rd rank services in the list, the next query will be $\{S_1, S_3, S_2, S_6\}$ and so on.

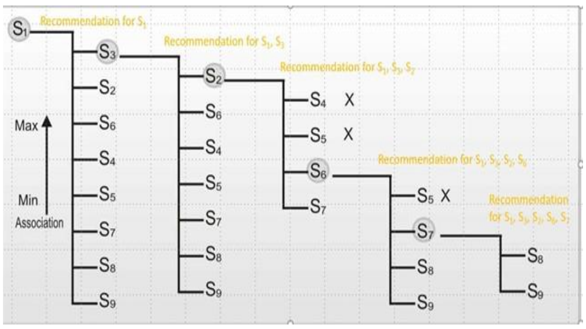

Figure 4. Service recommendation process using SAFRank.

In order to make user defined package, the above mentioned process is helpful and allows to select services on his/her own desires and that best suited to their deemed requirements. Here, user is creating its package P_u as follows:

$$P_u = \{S_1, S_3, S_2, S_6, S_7\}$$

Therefore, if database of all services, their associations and relevant packages are maintained, then one can find its best services of its choice. This technique allows learning from the past behaviour of users that which package(s) or service(s) has used. When a new user wants to use this system, with the help of association factor, it proceeds towards the best available services in the market.

The study explored several approaches and summarized them over important features for the provision of quality-based services to its seekers equipping their desires. The most successful approach on these parameters was presented in [2] but it is not allowing dynamics of quality parameters and not used the association of services with each other. This approach also covering two type of cloud services in its scope (i-e., SaaS and PaaS).

The study observed that to provide/recommend state-of-the-art services on one’s desires in all respects, the selection process should have evaluation criterion that includes:

- *Dynamic QoS*: parameters which should be dynamic in nature and one can prepare its own list/group for evaluation of services available in the pool.
- *Feedback*: selected service should be independent of any type of user’s feedbacks because due to its biasness and self-generation, a true ranking cannot be achieved.
- *Scope*: it’s important to know whether policy covers all types of internet services (i-e., SaaS, PaaS, IaaS or DaaS along with internet of things (IoT).
- *Constant Values*: policy should free from any type of value fixations.
- *Learning*: to perform rich evaluation on past experiences of users/CSPs, system should have learning mechanism, for making recommendations.
- *MAS*: self-organized system to work autonomously in all types of environments.

These aspects are elaborated in Table 5. These well-known approaches are examined on dynamic QoS attributes, feedback, scope of services as SaaS (S), PaaS (P) and IaaS (I), learning mechanism, fixed values and agent based approach. The ‘√’ shows to which scope of service is targeted.

Table 5. Summary of cloud service selection approaches.

Approaches	Dynamic QoS	Feedback	Scope S, P, I	Constant Values	Learning	MAS
AHP [8]	×	√	√	√	×	×
Historical Usage Data [3]	×	√	√	√	×	×
Performance Factor [19]	×	√	√√√	√	×	√
Multi-Agent System Arank [10]	×	√	√	√	×	√
Inference Rules [17]	√	×	√	×	√	×
Relative Service Ranking Vector and SMICloud Toolkit [5]	×	×	√	√	×	×
MAS+K-mean Clustering [13]	×	√	√√√	√	×	√
Group Decision Making Method (MCDM) [2]	×	×	√-√	√	×	×
SAF[18]	√	×	√√√	×	√	√

7. Conclusions

Cloud and IoT service selection policies have constraints of feedback biasness and self-generation, fixed QoS parameters, fixed values, coupled and single type services. In addition, they performed evaluation on single services and lacking recommendations when more than one service is set as input. The innovative approach SAFRank resolves these issues. It recommends services on a set of services mentioned as requirements. New seekers get best facilities as per their needs. The results show that technique works well in said scenarios. The Multi-agent system is helpful for online community/QoS brokers to have effective online resources. SAFRank is useful for identifying best online education courses, business applications, medical assistance, hotel reservations etc.

In future, we will improve SAFRank technique on backup services and suggest domain based QoS rankings to provide effective and better recommendation.

References

[1] Alhamad M., Dillon T., and Chang E., "A Trust-Evaluation Metric for Cloud Applications," *International Journal of Machine Learning and Computing*, vol. 1, no. 4, pp. 416-421, 2011.

[2] Al-Jabri B., Mustafa I., and Sadiq S., "A Group Decision-Making Method for Selecting Cloud Computing Service Model," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, pp. 449-456, 2018.

[3] Chan N., Gaaloul W., and Tata S., "A Recommender System Based on Historical Usage Data for Web Service Discovery," *Service Oriented Computing and Applications*, vol. 6, no. 1, pp. 51-63, 2012.

[4] Elazhary H., "User-Centric Cloud Service Broker," *International Journal of Computer Applications*, vol. 154, no. 7, pp. 28-35, 2016.

[5] Elmubarak S., Yousif A., and Bashir M., "Performance Based Ranking Model for Cloud SaaS Services," *International Journal of Information Technology and Computer Science*, vol. 9, no. 1, pp. 65-71, 2017.

[6] Fowley F., Pahl C., and Zhang L., *Handbook of Research on Architectural Trends in Service-Driven Computing*, IGI Global, 2016.

[7] Francis T. and Madhiajagan M., "A Comparison of Cloud Execution Mechanisms: Fog, Edge and Clone Cloud Computing," *in Proceeding of the Electrical Engineering Computer Science and Informatics*, vol. 4, pp. 446-450, 2017.

[8] Garg S., Versteeg S., and Buyya R., "SMICloud: A Framework for Comparing and Ranking Cloud Services," *in Proceeding of 4th IEEE International Conference on Utility and Cloud Computing*, Melbourne, pp. 210-218, 2011.

[9] Hao Z., Novak E., Yi S., and Li Q., "Challenges and Software Architecture for Fog Computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 44-53, 2017.

[10] Jahani A., Derakhshan F., and Khanli L., "A Multi-Agent Based Approach for Ranking of Cloud Computing Services, *Scalable Computing*," *Practice and Experience*, vol. 18, no. 2, pp. 105-116, 2017.

[11] Kavin B. and Ganapathy S., "A New Digital Signature Algorithm for Ensuring the Data Integrity in Cloud Using Elliptic Curves," *The International Arab Journal of Information Technology*, vol. 18, no. 2, pp. 180-190, 2021.

[12] Kornevs M., Minkevica V., and Holm M., "Cloud Computing Evaluation Based on Financial Metrics," *Information Technology and Management Science*, vol. 15, no. 1, pp. 87-92, 2012.

[13] Mahmood A., Shoaib U., and Sarfraz M., "A Recommendation System for Cloud Services Selection Based on Intelligent Agents," *Indian Journal of Science and Technology*, vol. 11, no. 9, pp. 1-6, 2018.

[14] Mezni H. and Abdeljaoued T., "A Cloud Services Recommendation System Based on Fuzzy Formal Concept Analysis," *Data and Knowledge Engineering*, vol. 116, pp. 100-123, 2018.

[15] Natesan G. and Chokkalingam A., "An Improved Grey Wolf Optimization Algorithm Based Task Scheduling in Cloud Computing Environment," *The International Arab Journal*

of Information Technology, vol. 17, no. 1, pp. 73-81, 2020.

[16] Online Reviews and Endorsements, Report on the CMA's call for information, [online] Competitions and Markets Authority (CMA). Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/436238/Online_reviews_and_endorsements.pdf, Last Visited, 2018.

[17] Parhi M., Pattanayak B., and Patra M., "A Multi-Agent-Based Framework for Cloud Service Discovery and Selection Using Ontology," *Service Oriented Computing and Applications*, vol. 12, no. 2, pp. 137-154, 2017.

[18] Rabbani I. and Muhammad A., "Internet Service Selection using Service Association Factor (SAF)," *Information Technology and Control*, vol. 48, no. 1, pp. 104-114, 2019.

[19] Rabbani I., Muhammad A., and Enriquez M., "Intelligent Cloud Service Selection Using Agents," *Advances in Intelligent Systems and Computing*, pp. 105-114, 2013.

[20] Rehman Z., Hussain O., and Hussain F., "Multi-Criteria IaaS Service Selection Based on QoS History," *in Proceeding of IEEE 27th International Conference on Advanced Information Networking and* Applications, Barcelona, pp. 1129-1135, 2013.

[21] Shi W., Cao J., Zhang Q., Li Y., and Xu L., "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.

[22] Shojaee S., Azman A., Murad M., Sharef N., and Sulaiman N., "A Framework for Fake Review Annotation," *in Proceedings of IEEE 17th UKSIM-AMSS International Conference on Modelling and Simulation*, England, pp. 153-158, 2015.

[23] Shrivastava A. and Rajawat A., "A Review on Web Recommendation System," *International Journal of Computer Applications*, vol. 83, no. 17, pp. 14-17, 2013.

[24] Tang L., Dong J., Peng T., and Tsai W., "Modeling Enterprise Service-Oriented Architectural Styles," Service Oriented Computing and Applications, vol. 4, no. 2, pp. 81-107, 2010.

[25] Tang W. and Yan Z., "Cloudrec: A Mobile Cloud Service Recommender System Based on Adaptive Qos Management," *in Proceedings of IEEE Trustcom/BigDataSE/ISPA*, Helsinki, pp. 9-16, 2015.

[26] Vakili M., Jahangiri N., and Sharifi M., "Cloud Service Selection Using Cloud Service Brokers: Approaches and Challenges," *Frontiers of Computer Science*, vol. 13, no. 1, pp. 599-617, 2018.

[27] Valant J., "The Case of Misleading or Fake Reviews," Online Consumer Reviews. European Parliamentary Research Service, PE 571. 301. Available at: https://www.europarl.europa.eu/RegData/etudes/BRIE/2015/571301/EPRS_BRI(2015)571301_EN.pdf, Last Visited, 2018.

[28] Wang X., Cao J., and Wang J., "A Dynamic Cloud Service Selection Strategy Using Adaptive Learning Agents," *International Journal of High Performance Computing and Networking*, vol. 9, no. 1/2, pp. 70-81, 2016.

[29] Wheal J. and Yang Y., "Csrecommender: A Cloud Service Searching and Recommendation System," *Journal of Computer and Communications*, vol. 3, no. 06, pp. 65-73, 2015.

[30] Yeh K., "An Efficient Resource Allocation Framework for Cloud Federations," *Information Technology and Control*, vol. 44, no. 1, pp. 64-76, 2015.

[31] Yi S., Hao Z., Qin Z., and Li Q., "Fog Computing: Platform and Applications," *in Proceedings of 3rd IEEE Workshop on Hot Topics in Web Systems and* Technologies, Washington, pp. 73-78, 2015.

[32] Yu T. and Lin K., "Service Selection Algorithms for Web Services with End-to-End QoS Constraints," *Information Systems and e-Business Management*, vol. 3, no. 2, pp. 129-136, 2004.

[33] Yu T., Zhang Y., and Lin K., "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," *ACM Transactions on the Web*, vol. 1, no. 1, 2007.

Imran Mujaddid Rabbani completed his PhD degree in computer science from the dept. of computer science, UET, Lahore, Pakistan in the year 2020. He completed his MSCS degree from Govt. College University, Lahore. He is presently working as an Instructor IT in Punjab Judicial Academy, Lahore, Pakistan. His main research areas are AI based Cloud Computing, Algorithms, Internet Based service selection and expert judicial systems.

Muhammad Aslam received the Ph.D. degree in computer science from the CINVESTAV-IPN, Mexico. He is currently serving as a Professor in the Dept. of CS, UET, Lahore, Pakistan. He has authored over 50 publications in different journals and conference of national and international repute. His current research interests include artificial intelligence, human–computer interface, image and speech processing, NLP, and computational intelligence.

Ana Maria Martinez-Enriquez is a full researcher at the Dept. of CS, CINVESTAV-IPN, D.F. Mexico, since October 1985. She received her Ph.D. in Computer Science from the University of Paris VI, France in March 1985, for the thesis entitled Automatic Archiving of Knowledge from Textual Documents (in French). Also, she received her Master in CS from University of Paris VI. Following her Ph.D. research activities, she focused on foundations, methods, and tools for modeling, building and validating applications based on KBS, natural language processing, machine learning, and data mining. Later, her research focuses on DAI, CSCW, deductive and adaptive collaborative interface, and group awareness. Her publications include several journals in JCR, in specialized journal in Computer Science, and International Conference.