

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Evaluating and Ranking Cloud IaaS, PaaS and SaaS Models based on Functional and Non-Functional Key Performance Indicators

FARRUKH NADEEM

Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. (e-mail: fabdullatif@kau.edu.sa)

ABSTRACT With the recent maturity of Cloud computing technology and the flexibility of the Cloud services, the offering of Cloud services has spread exponentially. Different Cloud service providers offer their services highlighting different features of their services. Because of the diversity of Cloud services and their highlighted features, the choice of the most suitable Cloud service is a complex problem for the Cloud users. Many Cloud users are unable to identify the Cloud services that best suit their needs and thus choose an unsuitable Cloud service, which results in financial losses as well as time delays. To this end, in this study, we propose an efficient three layered framework for evaluating and ranking IaaS, PaaS and SaaS Cloud services. We identified the functional and non-functional key performance indicators (KPIs) for Cloud services from 6 KPI classes. We classified these KPIs with respect to their types and criticality so that the Cloud user can easily choose according to his needs. The relative importance of the KPIs was determined using CRITIC method. We combined the KPIs values and their relative importance for an overall evaluation and ranking of the Cloud services using Vikor method. A case study is also presented for a step by step demonstration of the proposed method.

INDEX TERMS Cloud services, Cloud service KPIs, Cloud service evaluation, Cloud service ranking

I. INTRODUCTION

Cloud computing and virtualization technologies allow pooling of providers' distributed resources and offer them to remote users as a service over internet. Because of maintenance-free delivery, pay per use cost model, and ease of availability of Cloud services (CSs), the number of Cloud users (CUs) are increasing day by day. The increasing demand of Cloud services is attracting more and more Cloud service providers (CSPs) in the market. However, the CSPs differ from each other in various respects. Naturally, performance of some CSPs is better than others in some respects while others supersede in other respects. To maximize their benefits at the minimum cost, the CUs need to compare these CSs, and select the best CS that matches their needs.

Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), besides others, are among the major service models in the Cloud market. CSs can be effectively evaluated in terms of their KPIs. However, because of lack of architectural and performance

standardization of Cloud services, different CSPs provide different KPIs to differentiate their services from others, which makes the comparative evaluation of Cloud services a complex task. Common CUs are ignorant of the KPIs of CSs and thus can not comprehensively and effectively compare CSs. In addition, fuzzy nature of some KPIs makes this comparison even more difficult. Moreover, naturally, all the KPIs should not be treated equally because of different variations in their values and their supportive/conflicting behavior with each other. This behavior of KPIs makes the comparative evaluation of Cloud services even harder.

To provide a solution, in this study, we propose an efficient three-layered framework for evaluation and ranking for IaaS, PaaS and SaaS Cloud service models. The three layers address three needs of Cloud service evaluation and comparison. The layer-1 addresses the CUs' first need of identification of KPIs for evaluation of CSs. After KPIs identification, it is also important to consider the dispersion in the values of KPIs as well as their supporting/conflicting behavior of

KPIs with each other. This second need is addressed in layer-2, which calculates KPIs' weights using CRITIC method. Finally, the third need of ranking Cloud services based on KPIs' values and weights is addressed in layer-3, which uses Vikor method for overall ranking of CSs. We demonstrate the application of these layers through a step by step case study. Our proposed framework is computationally very light weight and can be easily implemented using spreadsheets software like MS Excel, etc.

The rest of the paper is organized as follows. Section II describes our framework for evaluation and ranking IaaS, PaaS and SaaS Cloud service models. The KPIs are identified in Section II-A. The Critic method for determining the weights of KPIs is described in Section II-B. The overall evaluation and ranking of the Cloud services using Vikor method is addressed in Section II-C. A step-by-step demonstration of the proposed framework is shown through a case study in Section III. The related work is described in Section IV. At last, we conclude in Section V.

II. FRAMEWORK FOR EVALUATION AND RANKING IAAS, PAAS AND SAAS CLOUD SERVICE MODELS

This section describes our three layered framework for evaluation and ranking of IaaS, PaaS and SaaS Cloud service models. Figure 1 shows our framework. In layer-1, we identify KPIs for the evaluation of IaaS, PaaS and SaaS service models. To make the selection of KPIs easy for common CUs, we classify KPIs as functional or non-functional. To further facilitate the CUs to select only some of the KPIs of their choice (for a simple and more effective comparison of CSs), we define the criticality of KPIs as high, medium and low. The criticality of KPIs is based on the functionality they provide. The KPIs targeting the core hardware/software resources and basic required features are rated as the most critical KPIs, and the KPIs for additional features are rated as medium or low. A CU may choose only highly critical KPIs or choose a combinational set of KPIs including high, medium and low criticality. It is noteworthy that to ease the CU for a simple yet effective evaluation of CSs, we do not target all performance indicators in this paper, we focus on KPIs only. In addition, we classify the KPIs as static or dynamic on the basis of whether they address static or dynamic aspects of CSs.

The layer-2 determines the relative importance of the KPIs in terms of KPI weights. The layer-3 ranks CSs considering KPIs' values and their weights. The details of these layers are described in following subsections II-A, II-B and II-C, respectively.

A. KPIs FOR IAAS, PAAS AND SAAS CLOUD SERVICE MODELS

This section provides a list of KPIs for evaluation of IaaS, PaaS and SaaS CS models. First, we present the KPIs common to these three models (referred to as general KPIs), and next, the KPIs specific to each of these models.

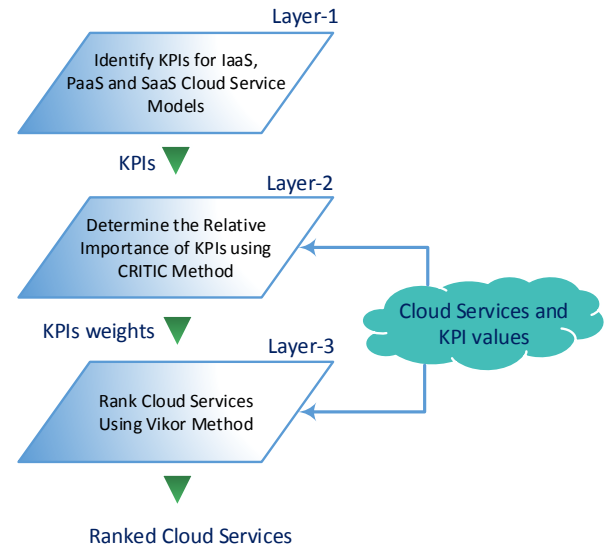


FIGURE 1. Framework for Evaluation and Ranking IaaS, PaaS and SaaS Cloud Service Models.

1) General KPI's (Common to IaaS, PaaS, SaaS)

- **Price:** Price is the first concern of common CUs while availing public CSs. Different CSPs offer different prices for different sets of features of their services. In addition to basic price of the CS, some CSPs charge extra for additional features they provide, for example, additional security features like for data storage and transfer, data encryption, firewall, etc. CUs should carefully identify their required features of the target CS and their prices from different CSPs for a better comparison.
- **Service Level Agreements (SLAs):** SLA is a contract between the CSP and the CU stating the level of QoS and associate details. The CUs want uninterrupted CSs with a certain level of performance efficiency. Otherwise, it can result in financial losses to CUs. Therefore, SLAs become very important [6], [14], [15], [4].
- **Network optimization:** The CU needs to connect to the network in order to access the CSs. For many CUs, the network performance will be a key factor for overall performance of the CS. CSPs should provide an optimized network to ensure efficiency of their services. The connection latency, bandwidth, parallel transfers, etc. are important features to evaluate a Cloud network. Some CSPs may provide special optimized links to their data centers for an extra price. Some CSPs also provide network performance monitoring and optimization tools [8], [38], [19].
- **Security:** Security is a major issue for both CUs as well as the CSPs. The CSPs must employ necessary procedures to secure CUs' information and data during and after the service use. Besides the security features provided by the CSPs, the CUs may still want to secure

their data from external attackers and internal snoopers. This problem can be mitigated by encrypting CUs' data. However, still there will be threats of data deletion and corruption by a third party. The CUs can use different data authentication technologies to overcome this issue [6], [11], [14], [15], [38], [9], [4], [26].

- **Throughput:** Throughput of a service tells us how efficiently the service is working. Rate of data storage on the disks and network data transfer are two commonly used measures for throughput. A higher throughput will increase the overall performance of a CS. While selecting a CS, CUs should carefully consider its throughput and mention any specific requirements of throughput in SLA [19].
- **Elastic Scaling:** The CSs are provided on-demand and can be scaled up and down as per CUs' requirements [32]. Some CSPs provide scaling on-demand while others provide it automatically. On one hand, this scaling down allows CUs to save cost and on other hand, small businesses can scale up to massive ones. The CU should select a CS which can scale up more easily and quickly. CU can also request CSP to design personalized strategies for scaling purposes. CU should ensure the provision of elastic scaling while confirming SLA [6], [37], [11], [14], [43], [4], [27].
- **Availability / Reliability:** CUs may access CSs from different parts of the world which fall in different time zones. Therefore, the CSPs must keep their services available for both domestic as well as international CUs. The storage systems of CSs are usually backed by several data servers. These servers require maintenance every now and then. It is important that CU has access to his data during maintenance, which is usually provided by keeping a backup of CU data on different data centers. To ensure high reliability, the CSPs should ensure that they have alternative arrangements in place, for example, for internet connection, power backup, storage, and computing servers, etc. [6], [11], [14], [44], [5], [27], [26], [32].
- **Load balancing:** It is important for the CU to consider whether an automatic load balancing is provided by the CSP or not and if it is, is it free or not. The authors in [32] discuss load balancing as an architectural requirement for the CSs.
- **Upgrades:** To offer latest features to their clients, CSPs should regularly upgrade their services. The availability of the CS and CUs' data and may be affected during upgrades. While upgrading their services, CSPs should backup CUs' data and provide service from alternative sources. Any unavailability of CUs' data or service will affect the reliability and can also harm the sensitive domains like business, academia, etc. Therefore, CUs should understand these issues when they are agreeing on the term and conditions of CSP [16].
- **Fault tolerance / Disaster recovery:** The CSPs should have well defined mechanisms for fault tolerance and

be well prepared for disaster recovery. There have been several cases of failure of CSs where CUs lost their sensitive data, e.g. failure of "Linkup" [14]. Therefore, CUs' data must be redundantly stored at different data centers to recover from disasters or faults. CSPs' capability for fault tolerance adds to reliability and trust of its services. CUs with sensitive applications and data must consider this factor critically and make it a part of SLAs [31], [26].

- **Response time:** It refers to the time between the service request and provision of service to the CU. It is one of the critical KPIs of CSs especially in case of frequently used business applications. In case of critical applications hosted by the CSPs, the response time requirements should be specified in SLA [19].
- **Runtime Performance Monitoring:** CSPs pool their resources to serve multiple CUs with scalable services. This feature creates a sense of infinite immediately available resources to CUs. The resource pooling strategy includes data storage, processing and data transfer services. CSPs may allocate resources from the same pool to multiple CUs, which can cause performance problems because of resource sharing. The CSPs should monitor the runtime service performance to ensure that the CUs do not experience a degraded performance when more CUs are assigned to a resource pool [13], [16], [37], [14], [8], [43].
Many of the CUs want to monitor the runtime performance of the CSs. Some CSPs provide no performance monitoring, while others provide it to different extents. For example, some CSPs offer WebSmart monitoring [32].
- **Trust:** Trust is among the major features in the public CS. Large businesses take this parameter very seriously and always choose most trustworthy CSPs. CUs will trust the CSPs that provide efficient, reliable and fault tolerant services. Trust also matters when CU and CSPs make SLAs.
- **Customer support:** Because of variety of the available CSPs and new Cloud technologies, customer support becomes a major need of CUs, especially for the new ones. Customer support may be provided through telephone, email, live chat, and formal documentation for using CS. The CSPs should also provide a knowledge base for the CS and CU forums as resources [32].
- **Customized quality of service** A CU must take care of QoS of the CSP if there are specific QoS requirements because some CSPs offer their services with a predefined quality of service (QoS), while others offer customized QoS according to requirements of CU. QoS is ensured with the help of flexible SLAs between CUs and CSPs [32].
- **Time to Consistency:** Time to consistency is the time between the data storage in the Cloud and when it is available for reading from the storage services. This is really important for the CU as CU may require data to be

available as soon as possible after storing it. CU should make sure that consistency is there when data is read and written within same data centers [19].

- **Scaling latency:** Scaling latency is the time taken by the CSP to scale up or down. Scaling latency can directly affect the performance of CSs [19].
- **Mobile access:** Now a days a lot of tasks are performed on mobile devices, like mobile phone, tabs, etc. The CUs should also compare CSs if those can be accessed via mobile devices or not.
- **Openness of Cloud:** It is important that CU should know about the hardware, software, security level, standardization and backup mechanisms, etc. of the CSP. Different categories of cloud openness are identified in literature i.e. unknown, basic, moderate and complete. In unknown category CU has vague idea of necessary information about CSs and it can become a concern for both CU and the CSP. In the second category (basic category) CU knows about only a few features of the CS, and in moderate the CU knows about the basic and some additional features of the CS. However, in complete category CU has full information about CSs [15].
- **Value added services:** Some CSPs also offer value-added services besides basic service, e.g. end to end data encryption, etc. [32].
- **Geo Location of Data Center:** While using CSs, CUs' data is stored and managed in different data centers, which may be located in various geo locations. Based

on the geo locations of the data center, the CUs' data may be subject to different regulatory laws of the countries/states in which the data center is located. It's necessary for both CUs and CSPs to mention the geo location of the data centers in the SLAs [16], [14], [32].

A summary of these KPIs and their classifications is given in Table 1. The distribution of KPIs with respect to criticality is shown in Figure 2. We observed that 61% of KPIs are highly critical, and 22% and 17% of KPIs have medium and low criticality, respectively. We also observed that the number of functional KPIs are 33% more than non-functional KPIs. However, the number of dynamic KPIs were 62.5% more than the static KPIs. This distribution is shown in Figure 3. The distribution of KPIs in different KPI classes is shown in Figure 4. We observed that the highest number of KPIs (30%) fall in performance class. Whereas, the lowest number of KPIs fall in agility class (3%).

2) KPI's for Evaluating Infrastructure as a Service (IaaS)

The KPIs for evaluating IaaS are already listed under general KPIs in Section II-A1. We do not list any KPIs specific to IaaS.

3) KPI's for Evaluating Platform as a Service (PaaS)

In addition to the general KPIs described in Section II-A1, following are the KPIs specifically used for PaaS.

- **Programming frameworks.** The users of PaaS may have requirements of specific programming frameworks to run their applications. For example, Python, Java

TABLE 1. Classification, criticality and types of common KPIs for IaaS, PaaS, SaaS.

#	KPI	KPI Class	Criticality	Type	Functionality
1	Price	Management	High	Static	Non-Functional
2	SLAs	Accountability	High	Static	Non-Functional
3	Network optimization	Management	High	Static	Functional
4	Security	Accountability	High	Dynamic	Functional
5	Throughput	Performance	High	Dynamic	Functional
6	Elastic scaling	Agility	High	Dynamic	Functional
7	Availability / Reliability	QoS	High	Dynamic	Functional
8	Load balancing	Performance	High	Dynamic	Functional
9	Upgrades	Performance	High	Static	Functional
10	Fault tolerance	Performance	High	Dynamic	Non-Functional
11	Response time	Performance	High	Dynamic	Functional
12	Runtime performance monitoring	QoS	High	Dynamic	Non-functional
13	Trust	Accountability	Medium	Static	Non-functional
14	Customer support	Management	Medium	Static	Non-Functional
15	Customized QoS	QoS	Medium	Dynamic	Non-functional
16	Time to consistency	Performance	Medium	Dynamic	Functional
17	Scaling latency	Performance	Medium	Dynamic	Functional
18	Mobile access	Usability	Low	Static	Non-Functional
19	Openness of Cloud	Accountability	Low	Dynamic	Functional
20	Value added services	QoS	Low	Dynamic	Functional
21	Geo location of data center	Accountability	Low	Static	Non-Functionall

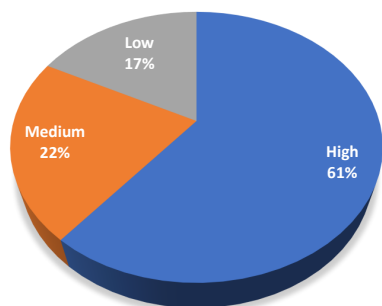


FIGURE 2. Distribution of general KPIs with respect to their criticality.

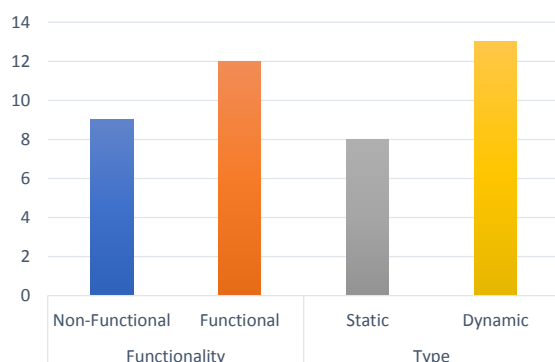


FIGURE 3. Distribution of general KPIs with respect to their type and functionality.

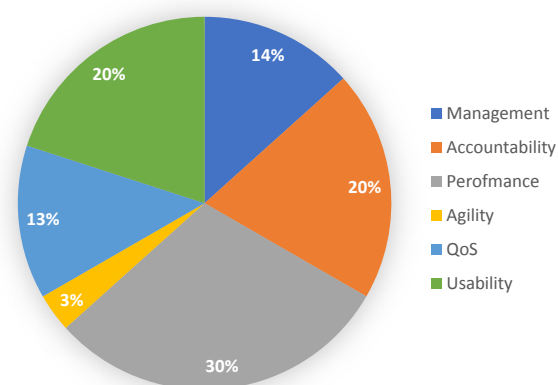


FIGURE 4. General KPIs distribution in different KPIs classes

or .NET etc. The CUs must carefully identify whether their selected CSs provide their required programming frameworks or not.

- **Root access:** A CU may require root access (administrative rights) for his applications while using PaaS. Therefore, the CU should carefully identify whether the administrative rights are provided by the CSPs or not.

- **Support for Integration of legacy applications:** Many businesses are still using some legacy applications, which are working fine for them. Due to expansion and other reasons, they want to run these applications in the Cloud. However, this requires necessary customized support from CSPs. This can be a decisive factor for the selection of a CS as unavailability of such support from a CSP will divert CU to other CSPs.
- **Provider Licenses:** The software licensing system restricts the usage of software on different computers. Many CSPs rely on both open and licensed software. Amazon and Google's PaaS both were built using open source programming languages, i.e., Xen for Amazon and Python for Google's PaaS. However, this is not the case for all. The CUs should take care of the license type of the software when using PaaS because they may have to pay extra charges for the use of licensed software [14], [15], [26], [32].
- **Interoperability:** A CS is interoperable with other services if it could be made a part of a bigger system easily. For the CUs who use CSs as a part of bigger systems, interoperability is very important [14], [27].
- **Lock-in for Data Migration:** In lock-in, a CU is restricted to move his data from one CSP to other CSPs. Every CSP is using its own format for storing data. Hence, it has become difficult to move data to other CSPs. Lock-in is one of the major restrictions in wide adoption of CSs. Most businesses are not putting their data on clouds because of "Lock-in". A CU should carefully consider this factor before choosing a CSP.

A summary of these KPIs and their classifications is given in Table 2.

4) KPI's for Evaluating Software as a Service (SaaS)

In addition to general KPIs described in Section II-A1, following are the KPIs used specifically for SaaS.

- **User Interface:** A wide range of users use CSs. These include technical experts, business professionals, laymen and even disabled. While designing their services, the CSPs should consider the accessibility of users of various types. The CSP should design easy to access interfaces of their services so that various types of CUs can understand and use them [6]. Other than that, mobile interfaces should be made available by CSPs as large number of CUs have shifted to handhelds and mobiles [37], [9], [27]. The CUs should also compare the CSs for the method or tool of access of CSs, for example, web browsers, command line tools or Application Programming Interface (APIs) [32].
- **Ownership:** When the data of CU is on CSP's infrastructure, the ownership rights of CU may be compromised. CSP can access the data and can use it for its own purpose. So, it is the responsibility of CSP to identify the rights of CUs over software, properties, and data. When making legal agreements, the CU and CSP should

TABLE 2. Classification, criticality and types of KPIs for PaaS.

#	KPI	KPI Class	Criticality	Type	Functionality
1	Programming frameworks	Usability	High	Static	Functional
2	Root access	Performance	High	Static	Functional
3	Legacy support/ Integration support	Management	Medium	Dynamic	Functional
4	Provider License	Security	Medium	Static	Non-Functional
5	Interoperability	Usability	Medium	Dynamic	Non-Functional
6	Lock-in	Usability	Low	Static	Non-Functional

TABLE 3. Classification, criticality and types of KPIs for SaaS.

#	KPI	KPI Class	Criticality	Type	Functionality
1	User Interfaces	Usability	High	Static	Functional
2	Ownership	Accountability	Low	Static	Non-Functional
3	Transparency	Usability	Low	Dynamic	Functional

agree upon these terms and conditions. When selecting or evaluating CSs, the CUs should consider this as a major issue and should add proper legal statements in SLA [6].

- **Transparency:** Transparency of different aspects of CSs, especially about security aspects, is an important demand from the CUs to CSPs. The higher transparency about CSs will help CUs to choose better services and will lead to more satisfied clients. On the other hand, less transparent CSs may lead to unbearable consequences [6], [27].

A summary of these KPIs and their classifications is given in Table 3.

B. DETERMINE THE RELATIVE IMPORTANCE OF KPIS USING CRITIC METHOD

Naturally, the KPIs will have different dispersion in their values. Moreover, some KPIs will support/conflict with other KPIs. Therefore, it is of critical importance to consider these behaviors of KPIs in the evaluation framework. We model these behaviors of KPIs in terms of KPI weights, which are determined using CRITIC (Criteria Importance through Inter-Criteria Correlation) method [7]. The Critic method determines KPIs' weights considering their conflicting behavior and degree of contrast. We applied CRITIC method for determining KPI's weights through the following seven steps. In step-1, we determined the decision matrix of values of n KPIs $\{K_1, K_2, K_3, \dots, K_n\}$ for the set of m CSs $\{CS_1, CS_2, CS_3, \dots, CS_m\}$ the CU want to evaluate and rank, as shown below.

$$\begin{array}{c|ccccc}
 & K_1 & K_2 & K_3 & \dots & K_n \\
 \hline
 CS_1 & \nu_{11} & \nu_{12} & \nu_{13} & \dots & \nu_{1n} \\
 CS_2 & \nu_{21} & \nu_{22} & \nu_{23} & \dots & \nu_{2n} \\
 CS_3 & \nu_{31} & \nu_{32} & \nu_{33} & \dots & \nu_{3n} \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 CS_m & \nu_{m1} & \nu_{m2} & \nu_{m3} & \dots & \nu_{mn}
 \end{array} \quad (1)$$

Here, ν_{ij} represents value of KPI j for CS i .

The scales of KPIs vary. To bring them to a uniform scale, in step-2, we normalized the above decision matrix using the equation 2 below.

$$\bar{\nu}_{ij} = \frac{\nu_{ij} - \nu_j^{worst}}{\nu_j^{best} - \nu_j^{worst}} \quad (2)$$

Here, $\bar{\nu}_{ij}$, ν_j^{best} , and ν_j^{worst} represent normalized value of ν_{ij} , best value of K_j and worst value of K_j , respectively. The best and worst values of K_j for the KPIs whose higher values are desired by a CU (e.g., throughput, etc.) correspond to the maximum and the minimum values of K_j , respectively. In contrast, the best and worst values of K_j for the KPIs whose lower values are desired by a CU, (e.g. cost, etc.) correspond to the minimum and the maximum values of K_j , respectively.

$$\begin{array}{c|ccccc}
 & K_1 & K_2 & K_3 & \dots & K_n \\
 \hline
 CS_1 & \bar{\nu}_{11} & \bar{\nu}_{12} & \bar{\nu}_{13} & \dots & \bar{\nu}_{1n} \\
 CS_2 & \bar{\nu}_{21} & \bar{\nu}_{22} & \bar{\nu}_{23} & \dots & \bar{\nu}_{2n} \\
 CS_3 & \bar{\nu}_{31} & \bar{\nu}_{32} & \bar{\nu}_{33} & \dots & \bar{\nu}_{3n} \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 CS_m & \bar{\nu}_{m1} & \bar{\nu}_{m2} & \bar{\nu}_{m3} & \dots & \bar{\nu}_{mn}
 \end{array}$$

The normalization of KPI values also allows us aggregation of different KPIs. The normalized decision matrix is shown below. In step-3, we calculate the standard deviation σ_j of normalized values of each K_j . Then, we have

$$\begin{array}{cccccc} K_1 & K_2 & K_3 & \dots & K_n \\ \sigma_1 & \sigma_2 & \sigma_3 & \dots & \sigma_n \end{array}$$

In step-4, we calculate an $n \times n$ symmetric matrix of the linear correlation coefficients γ_{ij} among KPIs.

	K_1	K_2	K_3	\dots	K_n
K_1	γ_{11}	γ_{12}	γ_{13}	\dots	γ_{1n}
K_2	γ_{21}	γ_{22}	γ_{23}	\dots	γ_{2n}
K_3	γ_{31}	γ_{32}	γ_{33}	\dots	γ_{3n}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
K_n	γ_{n1}	γ_{n2}	γ_{n3}	\dots	γ_{nn}

Here, γ_{ij} represents linear correlation coefficient between K_i and K_j .

In step-5, we calculate the total conflict of each KPI with the rest of the KPIs. For K_j , the total conflict ζ_j is calculated as

$$\zeta_j = \sum_{i=1}^n (1 - \gamma_{ij}) \quad (3)$$

After step-5, we have standard deviation and total conflict of all KPIs, as shown below.

K_1	K_2	K_3	\dots	K_n
σ_1	σ_2	σ_3	\dots	σ_n
ζ_1	ζ_2	ζ_3	\dots	ζ_n

In step-6, the quantity of information ϕ related to each KPI is determined as

$$\phi_j = \sigma_j * \zeta_j \quad (4)$$

Here, ϕ_j represent quantity of information related to K_j . After step-6, we have the standard deviation, total conflict, and quantity of information corresponding to all KPIs.

K_1	K_2	K_3	\dots	K_n
σ_1	σ_2	σ_3	\dots	σ_n
ζ_1	ζ_2	ζ_3	\dots	ζ_n
ϕ_1	ϕ_2	ϕ_3	\dots	ϕ_n

Finally, in step-7, the objective weights w_j of the KPIs are determined as

$$w_j = \frac{\phi_j}{\sum_{i=1}^n \phi_i} \quad (5)$$

The weights of KPIs are shown as below.

K_1	K_2	K_3	\dots	K_n
σ_1	σ_2	σ_3	\dots	σ_n
ζ_1	ζ_2	ζ_3	\dots	ζ_n
ϕ_1	ϕ_2	ϕ_3	\dots	ϕ_n
w_1	w_2	w_3	\dots	w_n

These weights will be used in overall evaluation and ranking of CSs using Vikor method, as described in the following section II-C. If a CU has his preferences about any KPIs, he should multiply the weight of his preferences with the

weights determined above. For example, the CU may want to give more preference to KPIs with high criticality, etc.

C. RANK CLOUD SERVICES USING VIKOR METHOD

For overall evaluation and ranking of CSs, we use Vikor (VIsekriterijumsko KOMpromisno Rangiranje) method [25]. The Vikor method is suitable for complex multicriteria decision making problems where there are a large number of criteria. The detailed steps of Vikor method are described as following.

In the first step of the Vikor method, we begin with the decision matrix as shown in matrix 1. In step-2 of Vikor, the normalized values of KPIs are calculated as

$$\hat{\nu}_{ij} = \frac{\nu_j^{best} - \nu_{ij}}{\nu_j^{best} - \nu_j^{worst}} \quad (6)$$

Here, $\hat{\nu}_{ij}$ represents the normalized value of ν_{ij} .

The normalized decision matrix is shown below.

	K_1	K_2	K_3	\dots	K_n
CS_1	$\hat{\nu}_{11}$	$\hat{\nu}_{12}$	$\hat{\nu}_{13}$	\dots	$\hat{\nu}_{1n}$
CS_2	$\hat{\nu}_{21}$	$\hat{\nu}_{22}$	$\hat{\nu}_{23}$	\dots	$\hat{\nu}_{2n}$
CS_3	$\hat{\nu}_{31}$	$\hat{\nu}_{32}$	$\hat{\nu}_{33}$	\dots	$\hat{\nu}_{3n}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
CS_m	$\hat{\nu}_{m1}$	$\hat{\nu}_{m2}$	$\hat{\nu}_{m3}$	\dots	$\hat{\nu}_{mn}$

In step-3, the weighted normalized values of the KPIs are calculated as $w_j * \hat{\nu}_{ij}$. The weighted normalized values are KPIs are shown below.

	K_1	K_2	K_3	\dots	K_n
CS_1	$w_1 * \hat{\nu}_{11}$	$w_2 * \hat{\nu}_{12}$	$w_3 * \hat{\nu}_{13}$	\dots	$w_n * \hat{\nu}_{1n}$
CS_2	$w_1 * \hat{\nu}_{21}$	$w_2 * \hat{\nu}_{22}$	$w_3 * \hat{\nu}_{23}$	\dots	$w_n * \hat{\nu}_{2n}$
CS_3	$w_1 * \hat{\nu}_{31}$	$w_2 * \hat{\nu}_{32}$	$w_3 * \hat{\nu}_{33}$	\dots	$w_n * \hat{\nu}_{3n}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
CS_m	$w_1 * \hat{\nu}_{m1}$	$w_2 * \hat{\nu}_{m2}$	$w_3 * \hat{\nu}_{m3}$	\dots	$w_n * \hat{\nu}_{mn}$

In step-4, the unity value U_i for each CS is obtained as a sum of weighted normalized values of KPIs obtained in step-3.

$$U_i = \sum_{j=1}^n (w_j * \hat{\nu}_{ij}) \quad (7)$$

After step-4, we have utility values of all CSs, as shown below.

CS_1	U_1
CS_2	U_2
CS_3	U_3
\vdots	\vdots
CS_m	U_m

In step-5, the regret value G_i of each CS is determined as

the maximum value of its weighted normalized KPIs.

$$G_i = \max_j (w_j * \hat{v}_{ij}) \quad (8)$$

After step-5, we have

$$\begin{array}{c|cc} CS_1 & U_1 & G_1 \\ CS_2 & U_2 & G_2 \\ CS_3 & U_3 & G_3 \\ \vdots & \vdots & \vdots \\ CS_m & U_m & G_m \end{array}$$

In step-6, the values of the minimum utility (U^-), the maximum utility (U^+), the minimum regret (G^-) and the maximum regret (G^+) are found as below.

$$U^- = \min_i U_i \quad (9)$$

$$U^+ = \max_i U_i \quad (10)$$

$$G^- = \min_i G_i \quad (11)$$

$$G^+ = \max_i G_i \quad (12)$$

Finally, in step-7, the Vikor rank Ω_i for each CS is calculated as

$$\Omega_i = \delta * \frac{U_i - U^-}{U^+ - U^-} + (1 - \delta) * \frac{G_i - G^-}{G^+ - G^-} \quad (13)$$

Here, δ represents the weight for the strategy of maximum group utility. After step-7, we have

$$\begin{array}{c|ccc} CS_1 & U_1 & G_1 & \Omega_1 \\ CS_2 & U_2 & G_2 & \Omega_2 \\ CS_3 & U_3 & G_3 & \Omega_3 \\ \vdots & \vdots & \vdots & \vdots \\ CS_m & U_m & G_m & \Omega_m \end{array}$$

The lower the value of Ω_i , the higher is the overall rank of the CS. The CU should select the highest overall ranked CS. The ultimate objective of this study is to rank CS based on the KPIs. Therefore, we limit our analysis to this step-of Vikor. However, if CU is interested in a compromised solution, he may further extend this analysis to find a compromised solution.

III. CASE STUDY

To demonstrate the application of the proposed framework, we present here a case study of ranking of five Cloud services: CS_1, CS_2, CS_3, CS_4 and CS_5 . For the sake of simplicity of the case, we will evaluate these CSs for five KPIs: K_1, K_2, K_3, K_4 and K_5 , where these KPIs represent price, response time, performance monitoring, security and customer support, respectively. The values of these KPIs for the selected CSs are shown below.

	K_1	K_2	K_3	K_4	K_5
CS_1	\$223	4.5s	5	4	3
CS_2	\$230	5s	4	3	5
CS_3	\$217	4s	1	5	3
CS_4	\$245	2.5s	3	4	5
CS_5	\$251	3s	4	5	4

(14)

Here K_1 is measured in dollars, K_2 in seconds, and K_3, K_4 , and K_5 are measured using a five point scale, where 1, 2, 3, 4 and 5 represent low, below average, average, good and excellent scales, respectively. It is noteworthy that the measurements of the KPIs and suitability of their measurement scale is out of scope of this paper. Several methods for the measurement of the KPIs and their scales have been proposed in literature. The user may choose any suitable method for measurement of KPIs. Here, we demonstrate a step-by-step application of layer-2 and layer-3 of our proposed framework in Section III-A and Section III-B, respectively.

A. DETERMINE THE RELATIVE IMPORTANCE OF KPIS USING CRITIC METHOD

In this section, we will demonstrate the application of CRITIC method of layer-1 (See Section II-B). Step-1 of the CRITIC method is determining the decision matrix of five available CSs for selected KPIs, which is given in matrix 14.

In step-2, the KPIs are normalized to adjust the difference of their scales using equation 2. For normalization, we need to find ν^{best} and ν^{worst} for each KPI. These are shown below.

	K_1	K_2	K_3	K_4	K_5
ν^{best}	\$217	2.5s	5	5	5
ν^{worst}	\$251	5s	1	1	1

(15)

The normalized value of K_1 for CS_1 is calculated using equation 2 as shown below.

$$\begin{aligned} \bar{\nu}_{11} &= \frac{223 - 251}{217 - 251} \\ &= 0.82 \end{aligned}$$

Similarly, the normalized values of remaining KPIs and other CSs are calculated. The normalized decision matrix is shown below.

	K_1	K_2	K_3	K_4	K_5
CS_1	0.82	0.2	1	0.75	0.5
CS_2	0.62	0	0.75	0.5	1
CS_3	1	0.4	0	1	0.5
CS_4	0.18	1	0.5	0.75	1
CS_5	0	0.8	0.75	1	0.75

In step-3, the standard deviation in the normalized values of each KPI is calculated, which is shown below.

	K_1	K_2	K_3	K_4	K_5
σ_i	0.42	0.41	0.38	0.21	0.25

In step-4, an $n \times n$ symmetric matrix of the linear correlation coefficients among the KPIs is calculated. This is shown below.

	K_1	K_2	K_3	K_4	K_5
K_1	1	-0.73	-0.30	-0.10	-0.61
K_2	-0.73	1	-0.22	0.52	0.24
K_3	-0.30	-0.22	1	-0.47	0.16
K_4	-0.10	0.52	-0.47	1	-0.60
K_5	-0.61	0.24	0.16	-0.60	1

In step-5 of the critic method, the conflict of each KPI with the rest of the KPIs is calculated as $1 - \gamma$. These conflict values are shown in the matrix below.

	K_1	K_2	K_3	K_4	K_5
K_1	0.00	1.73	1.30	1.10	1.61
K_2	1.73	0.00	1.22	0.48	0.76
K_3	1.30	1.22	0.00	1.47	0.84
K_4	1.10	0.48	1.47	0.00	1.60
K_5	1.61	0.76	0.84	1.60	0.00

The total conflict of each KPI with the rest of the KPIs is calculated using equation 3. The calculation of total conflict (ζ_1) for KPI K_1 are shown below.

$$\begin{aligned}\zeta_1 &= 0 + 1 - (-0.73) + 1 - (-0.30) + 1 - (-0.10) + \\ &\quad 1 - (-0.61) \\ &= 5.74\end{aligned}$$

Similarly, other values of ζ_i are calculated. The final values of ζ_i are shown below.

	K_1	K_2	K_3	K_4	K_5
σ_i	0.42	0.41	0.38	0.21	0.25
ζ_i	5.74	4.19	4.83	4.65	4.80

In step-6, the quantity of information (ϕ_i) related to each KPI is determined using equation 4. The calculation for ϕ_1 is shown below.

$$\begin{aligned}\phi_1 &= 0.42 * 5.74 \\ &= 2.44\end{aligned}$$

Similarly other values of ϕ_i are calculated and are shown below.

	K_1	K_2	K_3	K_4	K_5
σ_i	0.42	0.41	0.38	0.21	0.25
ζ_i	5.74	4.19	4.83	4.65	4.80
ϕ_i	2.44	1.74	1.83	0.97	1.20

Finally, in step-7, the objective weights of the KPIs are determined using equation 5. The calculation of w_1 is shown below.

$$\begin{aligned}w_1 &= \frac{2.44}{2.44 + 1.74 + 1.83 + 0.97 + 1.20} \\ &= 0.30\end{aligned}$$

Similarly, other values of w_i are calculated and are shown below.

	K_1	K_2	K_3	K_4	K_5
σ_i	0.42	0.41	0.38	0.21	0.25
ζ_i	5.74	4.19	4.83	4.65	4.80
ϕ_i	2.44	1.74	1.83	0.97	1.20
w_i	0.30	0.21	0.22	0.12	0.15

The CU may include his preferences about KPIs at this step. To do so, the CU will multiply the weights of his preferences with w_i . For the sake of simplicity, in this case study, we assume no preferences (or equal preferences) for KPIs and thus the above weights remain the same.

After determining the KPI weights, our method proceeds to the layer-2, where we evaluate and rank the selected Cloud services using Vikor method, which is described in the Section III-B below.

B. RANK CLOUD SERVICES USING VIKOR METHOD

In step-1 of the Vikor method, we begin with the decision matrix as shown in matrix 14. In step-2, this decision matrix is normalized using equation 6. The ν^{best} and ν^{worst} of the KPIs are shown in matrix 15. The normalized value of CS_1 for K_1 , i.e. $\hat{\nu}_{11}$, is calculated using equation 6 and is shown below.

$$\begin{aligned}\hat{\nu}_{11} &= \frac{217 - 223}{217 - 251} \\ &= 0.176\end{aligned}$$

Similarly, the remaining normalized values are calculated. The normalized decision matrix is shown below.

	K_1	K_2	K_3	K_4	K_5
CS_1	0.176	0.8	0	0.250	0.5
CS_2	0.382	1	0.25	0.5	0
CS_3	0	0.6	1	0	0.5
CS_4	0.824	0	0.5	0.25	0
CS_5	1	0.2	0.25	0	0.25

In step-3 of Vikor method, weighted normalized matrix is calculated by multiplying the KPI values by their weights as determined in matrix 16. The weighted normalized value of K_1 for CS_1 is calculated as $w_1 * \hat{v}_{11} = 0.30 * 0.176 = 0.053$.

Similarly, the weighted normalized values of remaining KPIs and other CSs are calculated. The weighted normalized matrix is shown below.

	K_1	K_2	K_3	K_4	K_5
CS_1	0.053	0.170	0	0.030	0.073
CS_2	0.114	0.213	0.056	0.059	0
CS_3	0	0.128	0.224	0	0.073
CS_4	0.245	0	0.112	0.030	0
CS_5	0.298	0.043	0.056	0	0.037

In step-4, the unity values U_i are calculated for each CS using equation 7. The calculations for U_1 are shown below.

$$U_1 = 0.298 * 0.176 + 0.213 * 0.800 + 0.224 * 0 + 0.119 * 0.25 + 0.147 * 0.5 = 0.326$$

Similarly, the other values of U_i are calculated. These unity values are shown below.

	U_i
CS_1	0.326
CS_2	0.442
CS_3	0.425
CS_4	0.387
CS_5	0.433

In step-5, the regret values G_i are calculated for each CS using equation 8. The regret values of CS_1 is calculated as

$$G_1 = \max(0.053, 0.170, 0, 0.030, 0.073) = 0.170$$

Similarly, the regret values for other CSs are calculated and are shown below.

	U_i	G_i
CS_1	0.326	0.170
CS_2	0.442	0.213
CS_3	0.425	0.224
CS_4	0.387	0.245
CS_5	0.433	0.298

In step-6, the minimum utility (U^-), the maximum utility (U^+), the minimum regret (G^-) and the maximum regret (G^+) are calculated using equations 9, 10, 11, 12, respectively. Their values are shown below

$$\begin{aligned} U^- &= 0.326 \\ U^+ &= 0.442 \\ G^- &= 0.170 \\ G^+ &= 0.298 \end{aligned}$$

Finally, in step-7, Vikor ranks Ω_i are calculated for all CSs using equation 13. For our case study, we set the value of $\delta = 0.5$. The calculations for Ω_1 are shown below.

$$\Omega_1 = 0.5 * \frac{0.326 - 0.326}{0.442 - 0.326} + (1 - 0.5) * \frac{0.170 - 0.170}{0.298 - 0.170} = 0$$

Similarly, other Ω_i are calculated. All values of all Ω_i are shown below.

	U_i	G_i	Ω_i
CS_1	0.326	0.170	0
CS_2	0.442	0.213	0.666
CS_3	0.425	0.224	0.638
CS_4	0.387	0.245	0.558
CS_5	0.433	0.298	0.961

The overall ranks are determined from values of Ω_i . The lower the value of Ω_i , the higher is the overall rank. The overall ranks of CS are shown below.

	U_i	G_i	Ω_i	Overall Rank
CS_1	0.326	0.170	0	1
CS_2	0.442	0.213	0.666	4
CS_3	0.425	0.224	0.638	3
CS_4	0.387	0.245	0.558	2
CS_5	0.433	0.298	0.961	5

IV. RELATED WORK

Because of multiplicity of the available CSs, their performance indicators of CSs and varying importance of performance indicators for different CUs, ranking and selection of CSs has become a hard problem [30], [10]. Several multi-criteria decision making techniques and meta-heuristic methods have been applied to evaluate and rank CSs. Many of these use *hybrid* methods for ranking CSs.

The authors in [24] proposed a hybrid broker for ranking CSs. First, the authors used Markov chain to model varying user preferences. Then, they employed best worst method for ranking CSs. Kumar et al. [18] also used a hybrid method employing AHP ("analytical hierarchical process") and TOPSIS ("Technique for order preference by similarity to ideal solution") to find the best CS. The AHP was employed to find weights of CS performance indicators and TOPSIS was used for ranking the CSs. The authors in [17] developed a

hybrid framework to rank the CSs based on their quality of service (QoS). They also employed best worst method to determine the user priorities for CS performance indicators and then used TOPSIS to finally rank the CSs. Another hybrid method based for clustering and ranking of the CSs was presented in [1] by Al-Faifi et. al. The authors used k-means clustering algorithm to cluster CSs based on their similarities and then used DEMATEL-ANP for ranking the CSP clusters. The study by Youssef [42] proposed a hybrid method based on TOPSIS and best worst method for ranking CSs. The study in [36] developed a framework based on TOPSIS and Gaussian distribution for ranking the CSs. The authors also analyzed the sensitivity of their results. The author in [12] also employed a hybrid method using artificial immune network and fuzzy theory for selection of CSs.

Some authors evaluated *QoS* of CSs and used it for CS selection. The author in [23] proposed a comprehensive CS measurement index and a flexible framework for ranking CSs based on user preferences for QoS and usability criteria. The author identified 65 KPIs to evaluate CSs and used MAGIC method for ranking the CSs. Li et. al. [21] also proposed a hybrid CS selection method based on heterogeneous QoS parameters. The heterogeneity and fuzziness of the QoS parameters was addressed through entropy weights and GRA-ELECTRE III method was used for overall evaluation of CSs. The work in [2] also employed fuzzy rough sets and weighted Euclidean distance to rank CSs. The work in [29] proposed a method employing matter element extension for ranking CSs. Sun et. al. [35] measured the relationship between CS selection criteria. Then, the authors estimated the criteria interaction and their importance. Next, they proposed a priority based method to select CSs. The work in [41] developed a QoS based method to recommend CSs. The authors predicted QoS from the nearest neighbors using NearestGraph.

Some studies proposed to use *trustworthiness* of CSPs to select the most suitable CS. These studies used various methods to model the trustworthiness of CSPs. Wu et. al. [40] evaluated and compared CSs for customized recommendations about their trustworthiness. The authors used fuzzy processing to handle imprecise user preferences and recurrent neural network to adapt the evaluation to the user preferences. Mujawar et. al. [22] modeled trustworthiness of CSP based on CSP's behavior and CU's feedback. The authors considered different QoS and SLA parameters to compute the trust. The authors in [39] also evaluated the trust of CSs. Different aspects of the trust were modeled using correlation analysis, rough set theory, and AHP. The work in [34] evaluated trustworthiness of CSPs based on compliance values of CSPs, which were further processed using a variant of TOPSIS. The study in [33] compared three MCDM approaches (PROMETHEE, TOPSIS, and AHP) for determining multi-perspective trustworthiness of CSPs.

In some other efforts, the authors modeled the *reliability* of Cloud data centers. The authors in [20] evaluated the reliability of Cloud data centers using HCGS petri nets and Monte Carlo Simulation. Their evaluation criteria included

overall performance of data center, connectivity of IT infrastructure and runtime delivery of service. Zhou et. al. [46] proposed a model for reliability evaluation of CSs using their subjective as well as objective attributes. Their model is based on hierarchy variable weight and classified statistics. Table 4 summarizes different related works.

Most of the existing methods are not comprehensive to address all KPIs or too complex to apply for CU. In contrast, our study, first of all, identifies important functional and non-functional KPIs for evaluating IaaS, PaaS, and SaaS models. Next, we propose a hybrid decision-making tool that is flexible to consider all or some functional as well as non-functional KPIs of the CS. For a more meaningful and effective evaluation and comparison of CSs, we calculate KPIs' weights employing critic method [7] that considers KPIs conflicting behavior as well quantity of information in KPI values. Next, to rank the CSs efficiently and consistently we use Vikor method. The Vikor method can effectively address complex ranking of various available CSs using with multiple KPIs.

V. CONCLUSION

With the increasing demand of Cloud services, the number of CSP are also increasing. Because of lack of standards for CSs, the CSP highlight different performance indicators of their services. This makes the choice of the best Cloud service difficult for the CUs. To address the needs of CUs for effective comparative evaluation and ranking of CSs, this study presents an efficient three layered framework. We highlight several KPIs for comprehensive evaluation of Cloud IaaS, PaaS and SaaS service models, and classify them as functional or non-functional and static or dynamic, which makes it easy for the CUs to choose KPIs of their choice in their evaluation. The second layer of our framework determines KPIs' weights based on variations in KPIs values as well as their conflicting behaviors. The third layer effectively ranks the CSs considering KPIs' weights. A step by step demonstration of the proposed framework is shown through a case study. Our framework is very easy to understand and light weight to implement, and can be easily implemented using spreadsheets.

ACKNOWLEDGEMENTS

This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (). The authors, therefore, acknowledge with thanks DSR technical and financial support.

REFERENCES

- [1] A. Al-Faifi, B. Song, M. Hassan, A. Alamri, and A. Gumaei. A hybrid multi criteria decision method for cloud service selection from smart data. *Future Gener. Comput. Syst.*, 93:43–57, 2019.
- [2] P. S. Anjana, P. Badiwal, R. Wankar, S. Kallakuri, and C. R. Rao. Cloud service provider evaluation system using fuzzy rough set technique. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 187–18709, 2019.
- [3] R. Buyya, R. Ranjan, and R. N. Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Proceedings of the 10th International Conference on Algo-*

TABLE 4. A summary of evaluation parameters recommended by different works.

Serial #	Authors	Service type	Static params.	Dynamic params.	Total params.	Func. params.	Non-func. params.	Work type	Services compared
1	Popovic et al. [28]	IaaS	1	2	3	2	1	Applied	3
2	Peng et al. [27]	IaaS	2	5	7	7	0	Applied	4
3	Pedro et al. [6]	SaaS, PaaS, IaaS	12	18	30	24	6	Applied	2
4	Chris Haddad [13]	PaaS	2	2	4	4	0	Suggestive	0
5	Simon et al. [26]	IaaS	0	2	2	2	0	Applied	1
6	Minqi et al. [45]	SaaS, PaaS, IaaS	2	3	5	4	1	Applied	27
7	Keith Combs [16]	IaaS	1	4	5	5	0	Suggestive	0
8	Cloudinfinet [4]	SaaS, PaaS, IaaS	5	1	6	5	1	Applied	5
9	Vaquero et al. [37]	SaaS, PaaS, IaaS	5	7	12	9	3	Applied	0
10	Rimal et al. [31]	SaaS, PaaS, IaaS	1	7	8	8	0	Applied	14
11	Godse et al. [11]	SaaS	3	2	5	4	1	Suggestive	0
12	Hilley et al. [14]	SaaS, PaaS, IaaS	6	13	19	15	4	Applied	15
13	Dillon et al. [8]	SaaS, PaaS, IaaS	3	3	6	5	1	Applied	0
14	Zhang et al. [44]	SaaS, PaaS, IaaS	2	5	7	7	0	Applied	4
15	Zhang et al. [43]	SaaS, PaaS, IaaS	2	5	7	7	0	Applied	3
16	Rajkumar et al. [3]	SaaS, PaaS, IaaS	0	3	3	3	0	Applied	5
17	Hoefer et al. [15]	SaaS, PaaS, IaaS	7	8	15	10	5	Applied	2
18	Voras et al. [38]	SaaS, PaaS, IaaS	0	6	6	6	0	Applied	10
19	Garg et al. [9]	SaaS, PaaS, IaaS	3	4	7	2	4	Applied	0
20	Li et al. [19]	PaaS, IaaS	2	5	7	7	0	Applied	3
21	Nasuni Corp. [5]	IaaS	3	0	3	3	0	Applied	5
22	Current work	SaaS, PaaS, IaaS	16	25	41	21	20	Applied	5

rithms and Architectures for Parallel Processing, Busan, Korea, pages 13–31, 2010.

- [4] Compare cloud providers, 2018. http://www.Cloudinfinet.com/company_compare; Last accessed: 06/02/2018.
- [5] N. Corporation. The state of cloud storage: A benchmark comparison of speed, availability and scalability. Technical Report 2015 Industry Report, White paper, 2105.
- [6] P. Costa, J. P. Santos, and M. M. d. Silva. Evaluation criteria for cloud services. In 2013 IEEE Sixth International Conference on Cloud Computing, pages 598–605, June 2013.
- [7] D. Diakoulaki, G. Mavrotas, and L. Papayannakis. Determining objective weights in multiple criteria problems: The critic method. Computers & OR, 22:763–770, 08 1995.
- [8] T. Dillon, C. Wu, and E. Chang. Cloud computing: issues and challenges. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, pages 27–33. IEEE, 2010.
- [9] S. K. Garg, S. Versteeg, and R. Buyya. Smicloud: A framework for comparing and ranking cloud services. In Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing, pages 210–218,

Washington, DC, USA, 2011. IEEE Computer Society.

- [10] Y. L. Ge Zhang, Weijie Wang. Understanding the complex adoption behavior of cloud services by smes based on complexity theory: A fuzzy sets qualitative comparative analysis (fsqca). Complexity, 2021.
- [11] M. Godse and S. Mulik. An approach for selecting software-as-a-service (saas) product. In 2009 IEEE International Conference on Cloud Computing, pages 155–158, Sept 2009.
- [12] W. Ha. Cloud service selection with fuzzy c-means artificial immune network memory classifier. In 2018 14th International Conference on Computational Intelligence and Security (CIS), pages 264–268, 2018.
- [13] C. Haddad. Selecting a cloud platform : A platform as a service scorecard. White paper, WSO2 : The Open Source Technology for Digital Business, December 12 2011.
- [14] D. Hilley. Cloud computing: A taxonomy of platform and infrastructure-level offerings. Technical Report git-cercs-09-13, Georgia Institute of Technology, 2009.
- [15] C. N. Höfer and G. Karagiannis. Cloud computing services: taxonomy and comparison. Journal of Internet Services and Applications, 2(2):81–94, Sep 2011.

- [16] Microsoft technet: Building iaaS cloud platforms, 2018. <https://blogs.technet.microsoft.com/privateCloud/2012/04/04/what-is-infrastructure-as-a-service-iaas/>; Last accessed: 06/02/2018.
- [17] R. R. Kumar, B. Kumari, and C. Kumar. Ccs-ossr: a framework based on hybrid mcdm for optimal service selection and ranking of cloud computing services. *Cluster Computing*, 24(2):867–883, 2021.
- [18] R. R. Kumar, S. Mishra, and C. Kumar. A novel framework for cloud service evaluation and selection using hybrid mcdm methods. *Arabian Journal for Science and Engineering*, 43(12):7015–7030, 2018.
- [19] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10*, pages 1–14, New York, NY, USA, 2010. ACM.
- [20] X.-Y. Li, Y. Liu, Y.-H. Lin, L.-H. Xiao, E. Zio, and R. Kang. A generalized petri net-based modeling framework for service reliability evaluation and management of cloud data centers. *Reliability Engineering & System Safety*, 207:107381, 2021.
- [21] M. Liu, Y. Shao, C. Yu, and J. Yu. A heterogeneous qos-based cloud service selection approach using entropy weight and gra-electre iii. *Mathematical Problems in Engineering*, 2020:1–17, 2020.
- [22] T. N. Mujawar and L. B. Bhajantri. Behavior and feedback based trust computation in cloud environment. *Journal of King Saud University - Computer and Information Sciences*, 2020.
- [23] F. Nadeem. A unified framework for user-preferred multi-level ranking of cloud computing services based on usability and quality of service evaluation. *IEEE Access*, 8:180054–180066, 2020.
- [24] F. Nawaz, M. R. Asadabadi, N. K. Janjua, O. K. Hussain, E. Chang, and M. Saberi. An mcdm method for cloud service selection using a markov chain and the best-worst method. *Knowledge-Based Systems*, 159:120–131, 2018.
- [25] S. Opricovic. *Multicriteria Optimization of Civil Engineering Systems*. PhD thesis, Faculty of Civil Engineering, Belgrade, 1998.
- [26] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. A performance analysis of ec2 cloud computing services for scientific computing. In D. R. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel, editors, *Cloud Computing*, pages 115–131, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [27] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li. Comparison of several cloud computing platforms. In *Proceedings of the 2009 Second International Symposium on Information Science and Engineering, ISISE '09*, pages 23–27, Washington, DC, USA, 2009. IEEE Computer Society.
- [28] O. Popovic, Z. Jovanovic, N. Jovanovic, and R. Popovic. A comparison and security analysis of the cloud computing software platforms. In *2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS)*, volume 2, pages 632–634, Oct 2011.
- [29] C. Radulescu, M. Radulescu, R. Boncea, I. Petre, I. Sandu, and M. Dumitrache. A multicriteria framework for cloud service providers selection based on the matter element extension method. *Studies in Informatics and Control*, 30:77–87, 2021.
- [30] M. Rahimi, N. Jafari Navimipour, M. Hosseinzadeh, M. H. Moattar, and A. a. Darwesh. Toward the efficient service selection approaches in cloud computing. *Kybernetes*, 2021.
- [31] B. P. Rimal, E. Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, NCM '09*, pages 44–51, Washington, DC, USA, 2009. IEEE Computer Society.
- [32] B. P. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven. Architectural requirements for cloud computing systems: an enterprise cloud approach. *Journal of Grid Computing*, 9(1):3–26, 2011.
- [33] J. Sidhu and S. Singh. Design and comparative analysis of mcdm-based multi-dimensional trust evaluation schemes for determining trustworthiness of cloud service providers. *Journal of Grid Computing*, 15(2):197–218, 2017.
- [34] J. Sidhu and S. Singh. Improved topsis method based trust evaluation framework for determining trustworthiness of cloud service providers. *Journal of Grid Computing*, 15(1):81–105, 2017.
- [35] L. Sun, H. Dong, O. K. Hussain, F. K. Hussain, and A. X. Liu. A framework of cloud service selection with criteria interactions. *Future Generation Computer Systems*, 94:749–764, 2019.
- [36] R. Tiwari and R. Kumar. G-topsis: a cloud service selection framework using gaussian topsis for rank reversal problem. *The Journal of Supercomputing*, 77:523–562, 2020.
- [37] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: Towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55, Dec. 2008.
- [38] I. Voras, B. Mihaljević, and M. Orlić. Criteria for evaluation of open source cloud computing solutions. In *Information Technology Interfaces (ITI), Proceedings of the ITI 2011 33rd International Conference on*, pages 137–142. IEEE, June 2011.
- [39] Y. Wang, J. Wen, X. Wang, B. Tao, W. Zhou, and J. Hu. A cloud service trust evaluation model based on combining weights and gray correlation analysis. *Sec. and Commun. Netw.*, 2019, Jan. 2019.
- [40] Z. Wu and Y. Zhou. Customized cloud service trustworthiness evaluation and comparison using fuzzy neural networks. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 433–442, 2016.
- [41] S. A. Yiqi Fu, Ding Ding. Using nearestgraph qos prediction method for service recommendation in the cloud. *Wireless Communications and Mobile Computing*, 2018, 2018.
- [42] A. E. Youssef. An integrated mcdm approach for cloud service selection based on topsis and bwm. *IEEE Access*, 8:71851–71865, 2020.
- [43] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, May 2010.
- [44] S. Zhang, S. Zhang, X. Chen, and X. Huo. Cloud computing research and development trend. In *Proceedings of the 2010 Second International Conference on Future Networks, ICFN '10*, pages 93–97, Washington, DC, USA, 2010. IEEE Computer Society.
- [45] M. Zhou, R. Zhang, D. Zeng, and W. Qian. Services in the cloud computing era: A survey. In *2010 4th International Universal Communication Symposium*, pages 40–46, Oct 2010.
- [46] P. Zhou, L.-M. Meng, X.-S. Qiu, Z.-S. Wang, Z.-P. Wang, and Z.-F. Chen. Evaluation of cloud service reliability based on classified statistics and hierarchy variable weight. *Journal of Signal Processing Systems*, 91(10):1115–1126, 2019.

...