ITC 3/51
Information Technology and Control
Vol. 51 / No. 3 / 2022
pp. 592-605
DOI 10.5755/j01.itc.51.3.30881

C-DRM: Coalesced P-TOPSIS Entropy Technique Addressing Uncertainty in Cloud Service Selection

Received 2022/03/09 | Accepted after revision 2022/08/17

http://dx.doi.org/10.5755/j01.itc.51.3.30881

HOW TO CITE: Nivitha, K., Parameshwaran, P. (2022). C-DRM: Coalesced P-TOPSIS Entropy Technique Addressing Uncertainty in Cloud Service Selection. *Information Technology and Control*, 51(3), 592-605. http://dx.doi.org/10.5755/j01.itc.51.3.30881

C-DRM: Coalesced P-TOPSIS Entropy Technique Addressing Uncertainty in Cloud Service Selection

K. Nivitha

Department of Information Technology, Rajalakshmi Engineering College, Chennai, Tamilnadu, India;
e-mail: nivithavimal@gmail.com

Pabitha Parameshwaran

Department of Computer Technology, Madras Institute of Technology, Chennai, Tamilnadu, India;
e-mail:pabithap@gmail.com

Corresponding author: nivithavimal@gmail.com

Cloud Computing is diversified with its services exponentially and lured large number of consumers towards the technology indefinitely. It has become a highly challenging problem to satiate the user requirements. Most of the existing system ingest large search space or provide inappropriate service; hence, there is a need for the reliable and space competent service selection/ranking in the cloud environment. The proposed work introduces a Clustering - Dual Ranking Method (C-DRM) to rank the services from n services in terms of space conserving and providing reliable service quenching the user requirements as well. C-DRM is proposed focusing on the uncertainty of user preferences along with their priorities; converting it to weights with the use of Jensen-Shannon (JS) Entropy Function. The ranking of service is employed through Priority-Technique for Order of Preference by Similarity to Ideal Solution (P-TOPSIS) and space complexity is reduced by novel Utility Pruning method. The performance of the proposed work C-DRM is estimated in terms of Closeness Index (CI) and space complexity. P-TOPSIS outperforms the conventional TOPSIS method by achieving 65% reduction in space complexity.

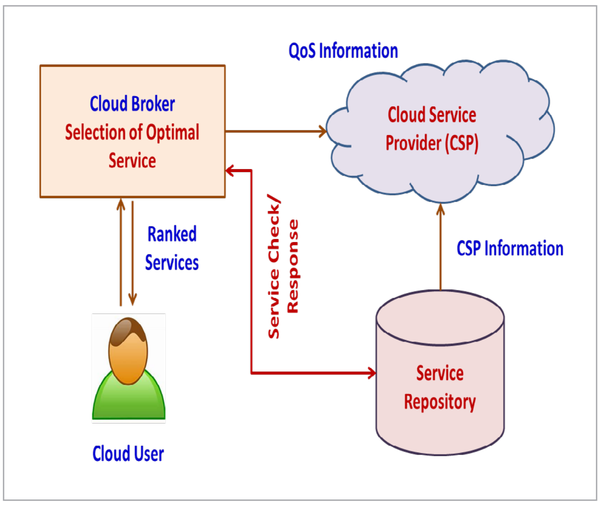
KEYWORDS: Cloud Computing, Clustering, Decision Making, Dual Ranking Method, Entropy, P-TOPSIS, Service Computing, Uncertainty.

1. Introduction

Cloud computing has become inevitable in its own way providing ample of services everyday [5, 23, 26]. This growth has given way for various Cloud Service Providers (CSP), to step into cloud market deploying variety of services, which include storage, computing, networking etc. With this development cloud computing has tossed away the need of enormous capital. The budding small and medium size Enterprises (SME's) emerges with fruitful services competing to the demand rising amongst cloud consumers and desired applications. Moreover, the pioneer technology started incorporating with trending expertise such as Big data, Internet of Things (IoT), Mobile Edge Computing, 5G [29, 30] making a pavement for huge growth in public CSP's such as Amazon, Google, Microsoft, Rackspace, etc. These CSP's provide variety of services with different range of Quality of Service (QoS) [11, 22] and cloud service pricing. This leads to a question: "How to select the right service out of many and still performing in better way?" An answer to this will be beneficial for both providers and consumers. The answer could help the consumer to pick the right service for instance, storage intensive application as one service and another service for networking intensive applications individually without any ambiguity and uncertainty. The general architecture of Cloud Service Selection (CSS) is depicted in Figure 1. The overall work has four major units: (i) Cloud User, (ii) Cloud Broker, (iii) Cloud service repository, and (iv) Cloud Service Providers. This diagram explains way of sending and receiving request from the cloud broker by the cloud user.

Figure 1

General Architecture of Cloud Service Selection



Due to the huge growth of cloud services with diversified characteristics, its tough task for the user to select appropriate cloud service satiating their requirements and large search space due to ambiguity, sometimes the similar services might conflict with one another based on the objectives [9, 23]. Moreover, with rise in demands of consumers the CSP's target to provide services with similar functions but still not trustworthy one. Hence, for a naive cloud user with less knowledge regarding the selection becomes a tricky part to handle the uncertainty in user preferences and QoS levels. To address these issues related to user preferences and huge search space, various research have attracted with notable interest to give solution separately but not on addressing all at once. However, there is still need to integrate these issues and provide a proficient cloud service selection/ranking for any cloud user on the go.

In order to address these challenges effectively, the proposed work Clustering-Dual Ranking Method (C-DRM) concentrates on the issues solving through two-phase and enhancing the trustworthy on final ranked services. Furthermore, the main concern of C-DRM is to facilitate the user with noteworthy service in a reduced search space on their actual QoS requirement through novel utility pruning method and P-TOPSIS method. The key contribution of the present research work are given as below:

1. Initially, similar services are clustered based on K-means algorithm pertaining to the user requirement.
2. To improve the space complexity and selection of optimal service, Dual Ranking Method is proposed and implemented. This model works on two-tier fashion projecting the utility pruning method which emphasis on the reduction of search space and efficiently ranking by considering the user priorities, P-TOPSIS has been implemented for the same.
3. An experiment is conducted in the form of case study to validate the performance of C-DRM. The results depicts that the proposed work is efficient

in terms of selecting the optimal cloud service and space conserving. The performance is illustrated by comparing with traditional ranking method.

The remainder of the work is organized as follows. Section 2 discusses the related work. Section 3 lines out the concept of proposed work P-TOPSIS in detail manner. The experimental analysis is presented in Section 4, various results are discussed, and finally Section 5 gives the conclusion and future work.

2. Related Works

Cloud Service Selection, a primary concern where N number of services being deployed with many similar matching characteristic that needs to be addressed for the benefits of cloud consumers in a long run. Most perplexing task to find the reliable service from cloud server, hence cloud service recommendation is proposed [21, 16] using clustering based on trust degree computation algorithm and service suggestion is given respectively. AHP techniques ensures the weights of user requirements, many researchers implemented for ranking the services as well [1, 7, 8, 11, 12, 19]. Other method of selecting services such as by using B^{cloud}-tree [14] and Fuzzy logic [17, 18], where the computational complexity and search space is large paving way for irrelevant selection. Rough set theory combined with hypergraph fruit fly optimization [20] yields better results in terms of accuracy but failed to impress with more services that are similar and increased search space as well. Time-series analysis of CSP's provide trustworthy providers but emanates with the cost of high complexity. The dominance of Multi Criteria Decision Making (MCDM) [15] stands recognized based on recent studies (Table 1) with respect to cloud service selection.

2.1. Novelty of the Proposed Method: C-DRM

There are various MCDM approaches used by several authors for ranking the cloud services. The literature work (Table 1) claims to handle the problem of service selection in cloud, nevertheless drawbacks still exists in terms of huge searching space for the optimal service thereby diminution in performance. Another significant issue is regarding the consideration of user priority, which needs to be incorporated while recommending the optimal service. The proposed method introduces the efficient technique to select the most

Table 1
Relate Works

Authors	Techniques
Cloud Service Ranking based on MCDM Approaches	
Nivethitha et al., 2019 [20]	– Rough set theory-based hypergraph-binary fruit fly optimization – service selection
Jatoth et al., 2019 [10]	– SELCLOUD – cloud service selection framework - EGTOPSIS – AHP – weights – Grey TOPSIS – rank CSPs
Nawaz and Janjua, 2021 [19]	– Broker based approach. – Time slot weighted satisfaction score. – Best Worst Method (BWM) – rank cloud services
Krishnakumar et al., 2021 [12]	– Orthopair Fuzzy information – express preferences – Agent attitude- variance approach
Hussain et al, 2021 [7]	– Fuzzy technique for best–worst analysis (FTBWA) – Based on final score the best alternative is selected.
Abdel et al. 2018 [1]	– Neutrosophic Multi Criteria Decision Analysis (NMCDA) – Triangular neutrosophic numbers – Neutrosophic AHP – performance evaluation of CSPs
Yousef 2020 [27]	– MCDM – TOPSIS – Best Worst Method (BWM)- rank Cloud service providers
Hussain and Chun 2022 [6]	– MCDM-Modified Best Worst Method (BWM) – to compute the weights of criteria – Markov chain- summation of ranks
Other Service Selection Approaches	
Priya, and Bhuvaneswaran, 2020 [21]	– Cloud service is suggested computing the trust degree by clustering the services – Qos Parameters are considered.
Nagarajan and Thirunavukarasu, 2019 [18]	– Fuzzy logic based intelligent cloud broker – Fuzzy inferencing process identifies sevices
Lin et al., 2019 [14]	– B^{cloud}-tree – service selection algorithm
Trueman et al., 2022 [25]	– Compute partial correlation between cloud service providers – Graphical Lasso Regularization – Ranks service providers through degree centrality

optimal cloud service by fixing the existing gaps. To aggregate the C-DRM (i) tackle the space complexity by minimizing the search space through similarity ranking, (ii) priority of the user requirements are taken into account and converted into weights using entropy function (iii) provides the reliable ranking of the cloud services by handling the uncertainty in user preferences.

The primary objectives of the C-DRM pertaining to two phases (Clustering and Ranking) are as follows:

Objective 1: *To identify the similar services and clustering into three classes, which can be, considered as Class 0, Class 1 and Class 2. The class with highest similarity pertaining to user requirements are the input to the next phase.*

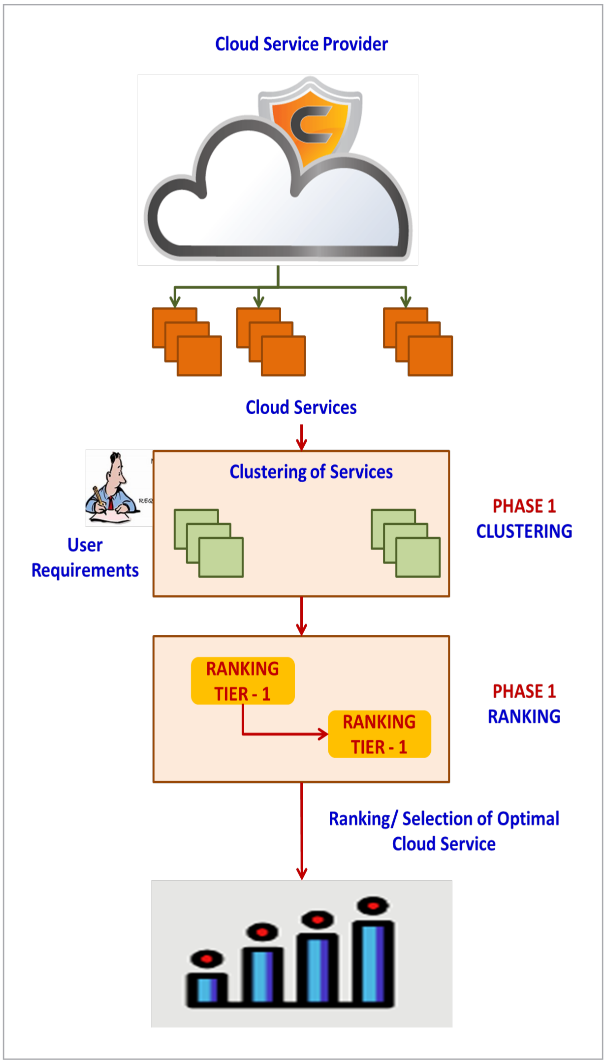
Objective 2: *To rank the cloud service from pool of Service and attaining the minimum search space Permissible to reduce the space complexity; the user Preferences variation is handled as well in tier 2 ranking.*

3. C-DRM Method: Proposed Cloud Service Selection/Ranking Algorithm

This section briefly describes about the Clustering-Ranking method based optimal cloud service selection/ranking. The entire process of C-DRM method embraces of two phases, specifically (i) Clustering and (ii) Ranking. The ranking phase is further segmented into two tier model called Dual Ranking Method with tier 1 having the novel pruning method and maximization function that defines the scalability, whereas tier 2 implements with the P-TOPSIS for ranking the service from pruned services among pool of services. Figure 2 gives detailed workflow explanation regarding the whole research work in terms of two phases. The C-DRM executes in the cloud broker, for simplicity the research work focused on single service provider in the case study as measure of validation. The phases of the research work are depicted in the Figure 2. With the minimal complexity involved in the computation of each phase. The clustered services are grouped into three classes and the class with most similar services are fed into the ranking phase. The tier 1 involves the pruning of outlier services based on the user priorities and the pruned

Figure 2

Workflow of Proposed C-DRM



services are moved to the tier 2 ranking involving the P-TOPSIS where ranking the service emphasizing on the priority of the user requirements. This flow greatly reduces the space complexity and divides the search into minimal one.

3.1. Problem Definition

Let $S = \{s_i \,|1\leq i \leq I\}$ denote set of cloud services that provide service to cloud users. Let $U = \{u_{i,j} \in U_i| 1\leq i \leq I, 1 \leq j \leq J\}$ denote the set of cloud users claiming the cloud service, where U_i represents the set of services belonging to specific cloud service provider, u_{ij} represents the j^{th} user of the i^{th} cloud service. Let

$A = \{a_k \mid 1 \le k \le K\}$ denotes the set of QoS attributes of cloud services with similar characteristic. Let $F = \{f_i \in A \mid 1 \le i \le I\}$ denotes the scaling attributes among the total QoS attributes. Let $T = \{t_{ij}(a_k)\}$ denote Service Level Agreement (SLA) of the k^{th} QoS attribute agreed by the i^{th} service and j^{th} user.

3.2. Phase 1 – Clustering

Nowadays, clustering is considered as essential pre-processing step for many real applications. Moreover, clustering algorithms brings out the most useful information for the application by grouping according to the various data similarity metrics. The K-means clustering is carried over KSA (Knowledge, Skills and Abilities) dataset, where the service data are collected from 13 cloud nodes [2].

3.3. Phase 2: Ranking

Ranking phase is aggregated form of two tiers such as tier 1 designates the novel utility pruning method based similarity ranking and tier 2 denotes the final ranking of services based on Jensen-Shannon (JS) divergence and incorporating the priorities of users as depicted in Figure 3

3.3.1. Tier 1 – Similarity Ranking

The class pertaining to the user preferences are recognized and shifted to the tier 1 ranking module for pruning of services in order to reduce the search space complexity. The pruning of services is done in accordance with scaling factors given by user - Table 2 to bring out the maximum benefits. The CSP submits the SLA Tij(ak) of the scaling attributes maintained by the broker, denoted as Tk where maximum scaling be done by the providers.

Definition 3.1. *For a given scaling attribute f_k, if the SLA $T_{ij}(a_k)$ submitted by the user u_j is not less than the Max utility Z_k, then it is considered that the cloud service satisfy the consistency on the scaling attribute f_k.*

In a real cloud environment, the scaling attributes (f_k) can be Memory and Disk. The user has primacy to give the two scaling factor to determine the affordable utility (AF_k) that can be managed without violating the Service Level Agreement (SLA). The formal definition is as follows:

Definition 3.2. *For the given scaling attribute f_k, the cloud service satisfies the condition of pruning is: $Z_k > AF_k$. The services are pruned when the max utility is larger than the affordable utility.*

Figure 3

Schematic Diagram of C-DRM

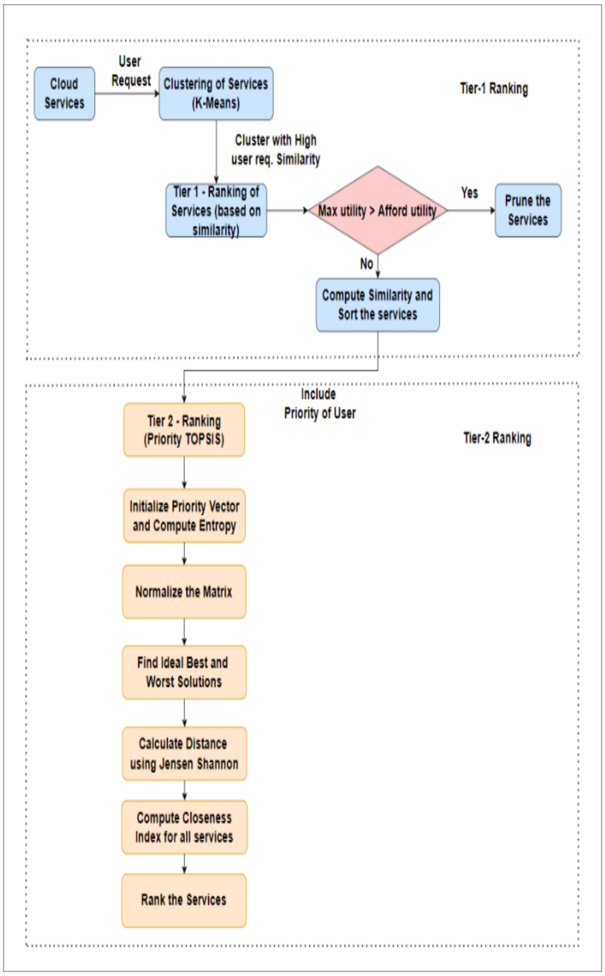


Table 2

SLA of Scaling Attributes Specified By Cloud User

Scaling attributes	Minimum	Maximum
Memory (gb)	2	20
Disk (gb)	100	1000

Based on the scaling factors (f_k) provided by user, the affordable utility (AF_k) is computed using Equation (1) and compared against the Max utility (Z_k) which is computed using knapsack optimization method using Equation (2) and compared with max utility; the services are pruned according to Equation (3). The similarity ranking algorithm is depicted in Table 3.

$$AF_k = f_{k1} f_{k2} \tag{1}$$

$$Z_k = max \sum_{k=1}^{n} f_k T_k \tag{2}$$

$$subject\ to\ \sum_{k=1}^{n} Z_k > AF_k, \tag{3}$$

where n is the No. of Services.

Table 3

Similarity Based Ranking Algorithm

```
Algorithm 1: Similarity Ranking

Input: S services in Class (C_n) with high similarity
Output: Pruned similarity ranked services
Variables: Afford_utility – utility affordable by
             user
       Max_utiltiy – maximum scalable utility
             by provider
Begin:
for S in C_n
    AF_k = f_k1 f_k2 // Compute Afford_utility
    Z_k = max Σ_{k=1}^{n} f_k T_k // Compute Max_utility
    while (Z_k > AF_k)
             Prune the row on true
    end while
end for
   Manhattan Distance = |x1 – x2| + |y1 – y2|
       // Compute the similarity of services
       Sort the M services based on similarity
End.
```

Once after pruning N services, the residual services are listed and similarity is found using Manhattan distance using Equation (4) ,where the top M services are approved over to tier 2 ranking to find the final ranking of services, here M takes the value 10. Such carry forward shrink the search space complexity in the final ranking.

$$Manhattan\ Distance = |x1 - x2| + |y1 - y2|, \tag{4}$$

where x and y are the attributes of user requirement and the attributes of other services.

For instance, the user requirement *(U_{req})* is given as depicted in Table 3. The similar services $Sim_j\ (CS_i, U_{req})$, *where j is the no. of attributes* pertaining to user requirement and are computed using Manhattan similarity technique.

The aggregated similarity values is found between two cloud services i.e user requirement and any cloud service $Sim_j\ (CS_i, U_{req})$, as given below Equation (5):

$$\begin{aligned} Sim(CS_i, U_{req}) = {} & Sim_{\text{cpucore}}(CS_i, U_{req}) + \\ & Sim_{mem}(CS_i,U_{req}) + Sim_{Disk}(CS_i, U_{req}) \\ & + Sim_{Bandwidth}(CS_i,U_{req}) + Sim_{throughput} \\ & (CS_i,U_{req}) + Sim_{responsetime}(CS_i,U_{req}) \end{aligned} \tag{5}$$

3.3.2. Tier 2 – P-TOPSIS Based Ranking

TOPSIS first proposed in [13, 28], which is the most proficient methods to handle the MCDM problems. The core concept behind TOPSIS is by identifying the low geometric distance from the ideal positive solution, which is termed as best alternative and should be farthest distance from the negative ideal solution as well. The positive ideal solution (best) has the advantage of having the highest benefits and in contrast, the negative ideal solution (worst) has the least benefits with low advantage of being chosen. Thus, the TOPSIS is indulged with Priority to turn out to be P-TOPSIS. Following are the common step processed initially for the ranking of the services.

Step 1: Compute the Normalized Matrix N using Equation (6)

$$n_{ij} = x_{ij} \sqrt{\sum_{i=1}^{m} x_{ij}^2} \tag{6}$$

for i = 1,2 ... m and j = 1,2 ... m

x_{ij}: Score of alternative services AS_n w.r.t criterion c_j (11 attributes as mentioned in chapter 4), n_{ij}: alternative $AS_{n\_}$ normalized score w.r.t criterion c_j (attributes) and m is the no. of attributes, i.e., 11 as in this case

Main concern of tier 2 ranking is to consider the priorities and dynamic reflection in case of any change in the user preferences. The priority of the user requirements are taken into account and converted into weights using entropy function (Equations 7(a)-(b)) as a solution to address uncertainty and the values are normalized. The converted weights are multiplied with the normalized matrix to form the weighted normalized matrix. The conventional TOPSIS does not consider the priority of user requirements and it is the limitation as well. Any minor change in the user preferences (priority) can be reflected instantaneously by updating the M services from the tier 1 ranking, since the list comprises the maximum similar services pertaining to user requirements. Table 4 depicts the P-TOPSIS algorithm with the steps of initializing priority vector and conversion to weights of the same

Table 4
P-TOPSIS Algorithm for the Final Ranking of Cloud Services

```
Algorithm 2: Priority (P) - TOPSIS
Input: M services from pruned dataset
Output: Final ranks of services (r_i)
Begin:
  While (M ≠ NULL) do
   Create Decision Matrix DM;
   Initialize Priority Vector (p) from user
   Compute inter-priority matrix between
   criterions c_i
   for i in p do
     mat = [(j/i) for j in p]
   end for
   for x_ij in DM do
     n_ij = x_ij √(Σ_{i=1}^{m} x_ij^2)
 end for
entropy_i = (− 1/ln(k) ∗ (P_i ∗ ln P_i))
   for c_i in no_criterions do
     D_c = 1 − entropy(i)
   end for
     v_ij = w_ij n_ij
     // Weighted Normalized decision matrix,
      w_ij is weights of each criteria
      calculated by entropy method
     if (j ∈ J ) then
     A_j^+ = max{v_1j , v_2j, ..., v_mj}
      A_j^− = min{v_1j , v_2j,..., v_mj}
     end if
     for j ∈ c_i do
      JS(R, S_i) = 1/2 KL(R, m) + 1/2 KL(S, m)
    // Distance calculation using Jensen
      Shannon Divergence formula
     d_i^+ = JS(R, S_i)
    // Estimates separation measure of positive
     Ideal solution
     d_i^− = JS(R, S_i)
    // Estimates separation measure of negative
      ideal solution
     end for
  for each v_ij in DM do
    r_i = d_j^− / (d_i^− + d_i^+)     // Compute Closeness Index (CI)
  end for
      Rank the Alternative services based on r_i
   // Higher the CI value is the most preferred
     Optimal cloud service alternatives.
  end while
End
```

followed by normalization and weighted normalization. From Algorithm 2, the cloud user is served with most optimal service as per the requirement quoted. Whereas, the remaining services are ignored and will be recomputed for further user requests.

Step 2: The priority of the user requirements are converted into weights using entropy function using the following Equations 7(a)-(b).

$$entropy_i = \left(-\frac{1}{\ln(k)} * (P_i * \ln P_i)\right) \tag{7a}$$

$$D_c = 1 - entropy_i, \tag{7b}$$

where, k is the no. of attributes and p_i is the probability value.

Step 3: Compute the weighted normalized matrix using Equation (8):

$$v_{ij} = w_{ij}\, n_{ij}, \tag{8}$$

where v_{ij}: is the weighted normalized matrix, w_{ij}: is the weight of criterion c_j.

Step 4: Acquire the Ideal solutions Positive (A_j^+) and Negative (A_j^-) using Equations (9)-(10):

$$A_j^+ = \max\{v_{1j}, v_{2j}, \ldots, v_{mj}\}. \tag{9}$$

$$A_j^- = \min\{v_{1j}, v_{2j}, \ldots, v_{mj}\}. \tag{10}$$

Step 5: Thereafter the normalization, ideal best and worst solution are computed and the ranking is calculated on the basis of closeness index, which is found using JS divergence formula as given in Equation (11).

$$JS(R, S_i) = \frac{1}{2} KL(R, m) + \frac{1}{2} KL(S, m) \tag{11}$$

$$m = \frac{1}{2}(R + S) m = \frac{1}{2}(R + S) \tag{11a}$$

$$KL(R, S_i) = -\sum_i log\, R \frac{S_i}{R}\; KL(R, S_i) = -\sum_i R\, log \frac{S_i}{R}, \tag{11b}$$

where, R is the given user requirement, S_i is the similar service to measure the distance and KL is the Kullback-Leibler divergence.

Step 6: Calculate the Closeness Index (r_i) for each alternative services to the ideal solution using Equation (12):

$$r_i = \frac{d_j^-}{d_i^- + d_i^+}. \tag{12}$$

Step 6: Rank R using Equation (13):

$$R = [r_1, r_2 \ldots r_m], \tag{13}$$

where R is the vector of all closeness Index, the highest closeness index is the best alternative as per Equation (13).

4. Experimental and Output Analysis

The dataset used in this research work is from the KSA (Knowledge, Skills and Abilities) Ministry of Finance [4], comprises of 28,147 instances from 13 cloud nodes out of which 2425 instances are considered after data preprocessing. The dataset contains 11 parameters such as Network Bandwidth in Kbps, Memory utilization, CPU utilization, Number of Jobs in a Minute, Number of Jobs in 5 min, Memory Capacity, Disk Capacity, Number of CPU Cores, CPU Speed per Core, Number of Jobs in 15 min, and response time in milliseconds. These are the parameters related to performance of a cloud service. The service selection is performed on Intel Pentium machine with python language in google colab.

4.1. Case Study

The performance and efficiency of proposed methodology is observed through the KSA dataset, where the information are collected from 13 cloud nodes as its performance related data. This work considered 10 services of same functionality for this case study and the services are evaluated based on 11 criteria's i.e., Number of Jobs in a Minute (C1), Number of Jobs in 5 min (C2), Number of Jobs in 15 min (C3), Memory Capacity (C4), Disk Capacity (C5), Number of CPU Cores (C6), CPU Speed per Core (C7), Network Bandwidth in Kbps (C8), Memory utilization (C9), CPU utilization (C10) and Response time (11). Table 5 Depicts the general outline of how the services been selected on user requirement basis, where S1, S2 and S3 are sample services. The services undergo Tier-1 and Tier-2 ranking and best optimal service is selected on the basis on Closeness Index measure.

4.1.1. Decision Matrix Establishment

For assumption the requirements are submitted by the cloud user to the cloud broker for providing the best service. The *M* candidate services i.e. eligible cloud Alternative Services (AS_i) are found by the cloud broker, where i = 1...m.

The Alternative services are evaluated based on the criteria's C_j, where j = 1....n, hence establishing the Decision Matrix (DM = $(x_{ij})_{m*n}$) of m cloud Alternative services with their n criteria's where all the service are under eligible category. The DM is established as follows:

$$DM = \begin{array}{c} AS_1 \\ AS_2 \\ \vdots \\ \\ AS_m \end{array} \begin{pmatrix} C_1 & C_2 & \dots & C_n \\ x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

Table 5
Example of Services

QoS Attributes		S1	S2	S3	User Requirement
Capacity	CPU Cores	8	8	8	6
	Memory (gb)	8	12	12	8
	Disk (gb)	128	128	128	128
	Bandwidth (kbps)	24	13	13	13
Throughput (%)		83	100	79	99
Response Time (sec)		60	60	70	60-120

4.1.2 Conversion of Priority to Weights Using Entropy Function

The user priority (preferences) are taken into consideration owing to conversion of criteria weights as mentioned in earlier chapter. The priorities are observed from user, here for case study the high priority is given to the criteria C11 (response time) and the least priority is given to the criteria C5 (Disk capacity). Taking these into account the alternative services are ranked through P-TOPSIS and the following steps takes place accordingly.

Step 1: The priority vector are converted to interpriority matrix i.e. initial steps of Algorithm 2.

Step 2: The entropy values are computed from interpriority matrix and converted to weights after normalizing the entropy values by computing Equation (7)(a-b), Tables 6(a)-(b) depicts the values of Normalized entropy and corresponding converted weights of each criterions.

4.1.3 Ranking the Services

Step 3: Weighted normalized matrix (v_{ij}) is determines using Equation (8), i.e., the weights of criteri-

Table 6(a)
Normalized Entropy Values of Each Criterion

Criterions	Normalized Entropy Values
C1	0.3186
C2	20.8687
C3	1.5098
C4	5.0664
C5	3.1290
C6	12.2415
C7	7.2573
C8	9.6591
C9	14.9817
C10	17.8621
C11	0.2136

Table 6(b)
Converting Entropy into Weights

Criterions	Weights
C1	0.0034
C2	0.2252
C3	0.0163
C4	0.0547
C5	0.0338
C6	0.1321
C7	0.0783
C8	0.1042
C9	0.1617
C10	0.1927
C11	0.0023

ons (w_{ij}) and normalized decision matrix (n_{ij}) are multiplied as depicted in Tables 7(a)-(b).

The criterions C4, C5 and C6 carries the same values since the top M services are most similar in values regarding the memory, disk and CPU cores.

Step 4: The ideal best (A_j^+) and worst (A_j^-) solutions are computed using Equations (9)-(10), thereby determining the best alternative service distances (d_i^+) and (d_i^-) from the ideal best and worst solutions through Jensen Shannon method using Equation

Table 7(a)
Weighted Normalized Matrix

Alternative Services	C1	C2	C3	C4	C5
AS1	0.0010	0.0705	0.0060	0.0173	0.0107
AS2	0.0010	0.0691	0.0055	0.0173	0.0107
AS3	0.0009	0.0635	0.0056	0.0173	0.0107
AS4	0.0012	0.0768	0.0053	0.0173	0.0107
AS5	0.0008	0.0549	0.0038	0.0173	0.0107
AS6	0.0011	0.0718	0.0047	0.0173	0.0107
AS7	0.0014	0.0894	0.0063	0.0173	0.0107
AS8	0.0013	0.0759	0.0052	0.0173	0.0107
AS9	0.0012	0.0802	0.0057	0.0173	0.0107
AS10	0.0008	0.0516	0.0036	0.0173	0.0107

Table 7(b)
Weighted Normalized Matrix

Alternative Services	C6	C7	C8	C9	C10	C11
AS1	0.0417	0.0247	0.0344	0.0529	0.0332	0.0006
AS2	0.0417	0.0247	0.0344	0.0529	0.0389	0.0008
AS3	0.0417	0.0247	0.0344	0.0529	0.0333	0.0006
AS4	0.0417	0.0247	0.0344	0.0529	0.0314	0.0008
AS5	0.0417	0.0247	0.0263	0.0435	0.0362	0.0006
AS6	0.0417	0.0247	0.0344	0.0529	0.0927	0.0006
AS7	0.0417	0.0247	0.0344	0.0529	0.0283	0.0007
AS8	0.0417	0.0247	0.0344	0.0529	0.0977	0.0010
AS9	0.0417	0.0247	0.0344	0.0529	0.0948	0.0007
AS10	0.0417	0.0247	0.0263	0.0435	0.0567	0.0008

(11). The services are ranked based on CI values computed through Equation (12) and depicted via Table 8.

Table 8
P-TOPSIS Analysis Results for Cloud Services

Alternative Services (AS_n)	Distance from Ideal Best (d_i^+)	Distance from Ideal Worst (d_i^-)	Closeness Index (CI)	Rank (r_i)
AS1	0.1394	0.0475	0.2540	**8**
AS2	0.1217	0.0480	0.2830	**6**
AS3	0.1404	0.0390	0.2170	**9**
AS4	0.1449	0.0558	0.2788	**7**
AS5	0.1263	0.0284	0.1837	**10**
AS6	0.0323	0.1454	0.8181	**3**
AS7	0.1586	0.0802	0.4530	**5**
AS8	0.0249	0.1529	0.8610	**2**
AS9	0.0191	0.1488	0.8857	**1**
AS10	0.0846	0.0881	0.5101	**4**

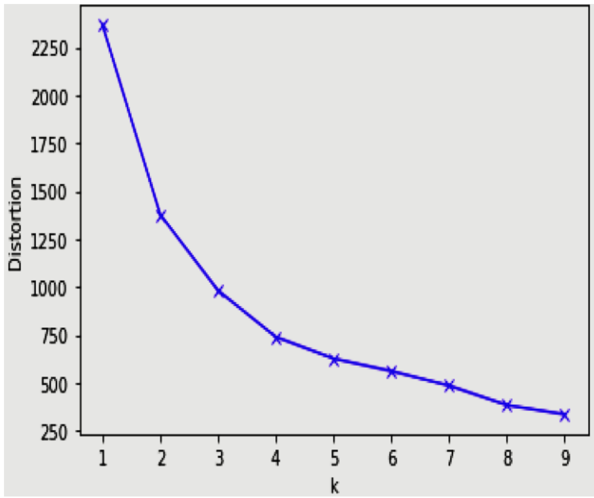
4.2. Clustering Experimental Analysis

As the proposed method involves two segments, clustering being the first phase. The KSA dataset is clustered based on K-means techniques which in turn made use of the elbow method to determine the count of clusters as indicated in Figure 4. Thus, the finest quantity of clusters is chosen as 3. The number of instances in each clusters has been depicted in Table 9. The most similar cluster (i.e., the cluster 2 as in this work) is submitted to the ranking phase for the further processing with similarity ranking and priority based ranking to obtain the final service pertaining to user requirements and priority given.

Table 9
Number of Instances in each Clusters

Clusters	No. of Instances (Services)
0	1225
1	999
2	201

Figure 4
Elbow Method with Optimal K



4.3. Rank Conformance Analysis

The proposed method (P-TOPSIS) has been compared with existing method (TOPSIS) to analyze its conformity. The same KSA dataset is considered for both the methods to prove their performance. Figure 5 shows the ranking of two MCDM techniques. Both the method showed full consensus on several ranks of alternative service i.e. 4th, 6th, 8th and 10th ranks. The result proves the proposed method is consistent with existing MCDM method as well.

Figure 5
Ranking of Cloud Service with Different Methods

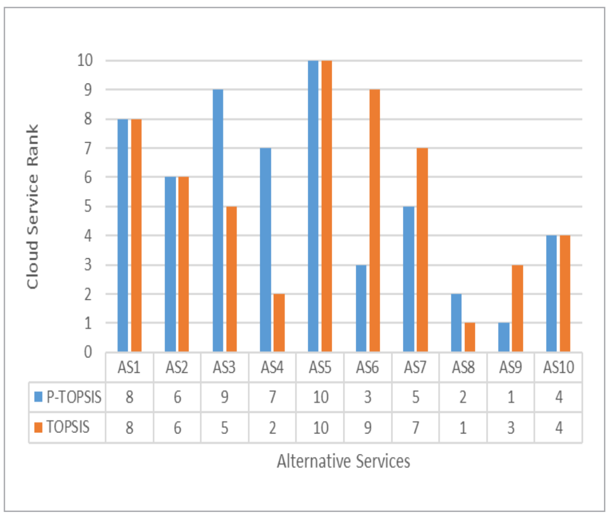


Figure 6 depicts the services for the first 5 ranks based on closeness index for both the MCDM techniques. Higher the CI higher the similarity between user requirement and chosen service. The ideal best and ideal worst values along with the CI values are shown in Table 8.

Figure 6

Ranking of Services Based on Closeness Index (CI)

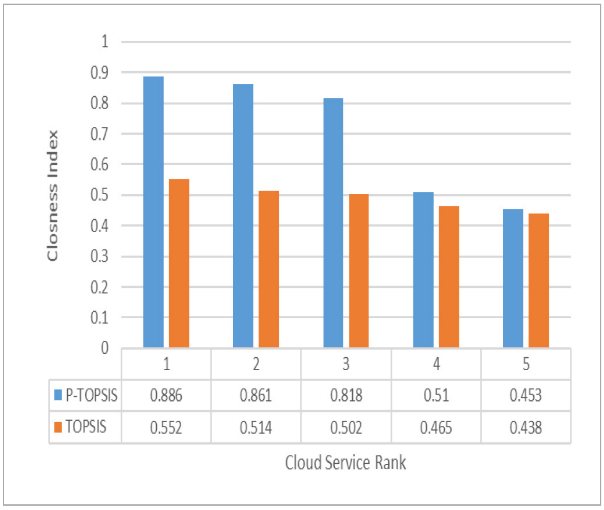
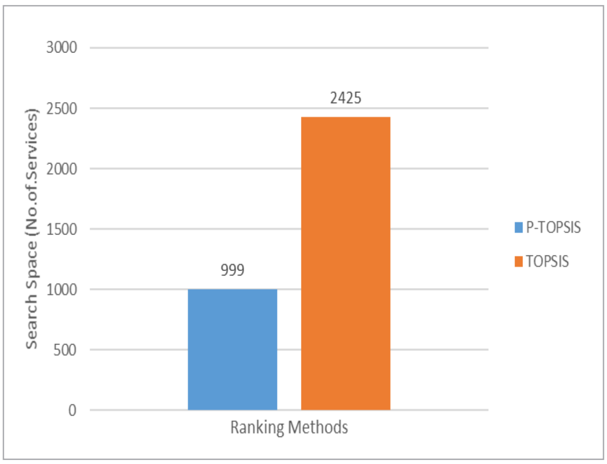


Figure 7

Search Space Redcution Graph of P-TOPSIS and TOPSIS



As the Figure 7 depicts the space complexity, can analyse that P-TOPSIS has reduced the space complexity of about 65% when compared to the conventional TOPSIS method, showing 2425 are the no. of services (search space) for existing method where as merely 999 services undergo the ranking phase for proposed method. Thus, the proposed method outperforms the traditional method significantly.

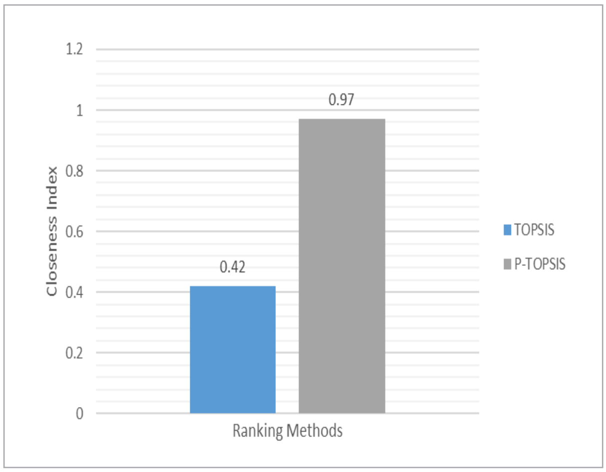
4.4. Performance Analysis on Amazon Data

Proposed C-DRM is validated through case study, data are collected from Amazon website. Due to the unavailability of variety of attributes, small analysis in done with pre-dominate attributes such as CPU, Memory and Bandwidth and the proposed method exhibits the flexiblility to accomadate these attributes. 50 service instances were collected and gone through each phases of C-DRM, undoubtedly the P-TOPSIS showed substantial improvement in terms of finding similar service to the user requirement.

The performance comparison (Figure 8) is computed with Closeness Index (CI) to verify the most similar service pertaining to the user requirement.

Figure 8

Performance Comparison on Amazon Data with P-TOPSIS and TOPSIS



4.5. Rank Reversal

The phenomenon associated with change in alternative rank order is termed as Rank Reversal [2, 3, 24], when new alternative is added/removed in entire decision making process. The only issue of using TOPSIS method is the rank reversal which can be solved by making the normalization a constant value hence the CI values doesn't change when a service is added or removed. This issue is considered as special case of the proposed method. The authentic normalization, Equation (1), implemented for solving the issue of rank reversal and the normalization can be converted into Equation (14) as follows:

$$n_{ij} = x_{ij} / Min_i(x_{ij}). \tag{14}$$

The following Table 10 provides the details of rank reversal before and after the change of normalization, for both cases the removal of an alternative and addition of an alternative as well. The original CI value is checked against the rank reversal for the alternative service AS_7.

Table 10

Rank Reversal Analysis

Ranking Methods	Original CI value	CI Value upon Addition	CI Value upon Removal
TOPSIS	0.26	0.62	0.74
P-TOPSIS	0.45	0.45	0.45

5. Conclusion and Future Work

The dawn of cloud computing and its impression on various other business works has paved way for spike in abundant cloud service providers posing plenty of similar services with diversification of features. The rapid increase with trending technologies, the number of cloud services is skyrocketing with plunge in the identification of appropriate service for the cloud user. The persistent struggles by the research community had driven the service selection in cloud as a conspicuous key to the issue of cloud service selection. To be precise in addressing the user prefernces and the chore of colossal search space.

In such way, the proposed work Clustering-Ranking based cloud service selection flagged way to address the challenge in an fitting manner. The entire worklfow of C-DRM comprises of two phases, namely (i) Clustering phase: grouping the similar services and (ii) Ranking phase: tier 1 – to condense the space complexity and tier 2 – to inculcate the user priority and to address the variation in user preference change at concluding stage. The experimental analysis proves the performance of C-DRM over the existing approach. For the future work addition of user feedback along with several service providers and corresponding information can be considered for selection and malicious records need to be addressed efficiently.

References

1. Abdel-Basset, M., Mohamed, M., Chang, V. NMCDA: A Framework for Evaluating Cloud Computing Services. Future Generation Computer Systems, 2018, 86, 12-29. https://doi.org/10.1016/j.future.2018.03.014
2. Aires, R. F. D. F., Ferreira, L. The Rank Reversal Problem in Multi-Criteria Decision Making: A Literature Review. Pesquisa Operacional, 2018, 38(2), 331-362. https://doi.org/10.1590/0101-7438.2018.038.02.0331
3. Aires, R. F. D. F., Ferreira, L. The Rank Reversal Problem in Multi-Criteria Decision Making: A Literature Review. Pesquisa Operacional, 38, 2018, 331-362. https://doi.org/10.1590/0101-7438.2018.038.02.0331
4. Al-Faifi, A. M., Song, B., Hassan, M. M., Alamri, A., Gumaei, A. Data on Performance Prediction for Cloud Service Selection. Data in Brief, 2018, 20, 1039-1043. https://doi.org/10.1016/j.dib.2018.08.108
5. Distefano S, Puliafito, A. Cloud@Home: Toward a Volunteer Cloud. IT Professional, 2012, 14(1), 27-31. https://doi.org/10.1109/MITP.2011.111
6. Hussain, A., Chun, J. Cloud Service Scrutinization and Selection Framework (C3SF): A Novel Unified Approach to Cloud Service Selection with Consensus. Information Sciences, 586, 2022, 155-175. https://doi.org/10.1016/j.ins.2021.11.024
7. Hussain, A., Chun, J., Khan, M. A Novel Multicriteria Decision Making (MCDM) Approach for Precise Decision Making Under a Fuzzy Environment. Soft Computing, 2021, 25(7), 5645-5661https://doi.org/10.1007/s00500-020-05561-9
8. Jagatheswari, S., Ramalingam, P., Chandra Priya, J. Improved Grey Relational Analysis-based TOPSIS Method for Cooperation Enforcing Scheme to Guarantee Quality of Service in MANETs. International Journal of Information Technology, 2022, 14, 887-897. https://doi.org/10.1007/s41870-022-00865-5
9. Jatoth, C., Gangadharan, G. R., Buyya, R. Computational Intelligence Based QoS-aware Web Service Composition: A Systematic Literature Review. IEEE Tran-

sactions on Services Computing, 2015, 10(3), 475-492. https://doi.org/10.1109/TSC.2015.2473840

10. Jatoth, C., Gangadharan, G. R., Fiore, U., Buyya, R. SELCLOUD: A Hybrid Multi-criteria Decision-making Model for Selection of Cloud Services. Soft Computing, 2019, 23(13), 4701-4715. https://doi.org/10.1007/s00500-018-3120-2
11. Jin, H., Lv, S., Yang, Z., Liu, Y. Eagle Strategy Using Uniform Mutation and Modified Whale Optimization Algorithm for QoS-aware Cloud Service Composition. Applied Soft Computing, 2022, 114, 108053. https://doi.org/10.1016/j.asoc.2021.108053
12. Krishankumar, R., Arun, K., Pamucar, D., Ravichandran, K. S. A Decision Aiding Framework for Two-Way Selection of Cloud Vendors in Medical Centers With Generalized Orthopair Fuzzy Information. IEEE Transactions on Engineering Management, 2021. https://doi.org/10.1109/TEM.2021.3097139
13. Lai, Y. J., Liu, T. Y., Hwang, C. L. Topsis for MODM. European Journal of Operational Research, 1994, 76(3), 486-500. https://doi.org/10.1016/0377-2217(94)90282-8
14. Lin, D., A. C. Squicciarini, V. N. Dondapati, Sundareswaran, S. A Cloud Brokerage Architecture for Efficient Cloud Service Selection. IEEE Transactions on Services Computing, 2019, 12(1), 144-157. https://doi.org/10.1109/TSC.2016.2592903
15. Liu, L., Lu, C., Xiao, F., Liu, R., Xiong, N. N. A Practical, Integrated Multi-criteria Decision-making Scheme for Choosing Cloud Services in Cloud Systems. IEEE Access, 2021, 9, 88391-88404. https://doi.org/10.1109/ACCESS.2021.3089991
16. Malouche, H., Halima, Y. B., Ghezala, H. B. Trust Level Estimation for Cloud Service Composition with Inter-service Constraints. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(12), 4881-4899. https://doi.org/10.1007/s12652-019-01182-9
17. Nagarajan, R., Thirunavukarasu, R., Shanmugam, S. A Fuzzy-based Intelligent Cloud Broker with Mapreduce Framework to Evaluate the Trust Level of Cloud Services Using Customer Feedback. International Journal of Fuzzy Systems, 2018, 20(1), 339-347. https://doi.org/10.1007/s40815-017-0347-5
18. Nagarajan, R., Thirunavukarasu, R. A Fuzzy-based Decision-making Broker for Effective Identification and Selection of Cloud Infrastructure Services. Soft Computing, 2019, 23(19), 9669-9683. https://doi.org/10.1007/s00500-018-3534-x
19. Nawaz, F., Janjua, N. K. Dynamic QoS-aware Cloud Service Selection Using Best-worst Method and Timeslot Weighted Satisfaction Scores. The Computer Journal, 2021, 64(9), 1326-1342. https://doi.org/10.1093/comjnl/bxaa039
20. Nivethitha, S., Raman, M. G., Gireesha, O., Kannan, K. Sriram, V. S. An Improved Rough Set Approach for Optimal Trust Measure Parameter Selection in Cloud Environments. Soft Computing, 2019, 23(22), 11979-11999. https://doi.org/10.1007/s00500-018-03753-y
21. Priya, A. S. B., Bhuvaneswaran, R. S. Cloud Service Recommendation System Based on Clustering Trust Measures in Multi-cloud Environment. Journal of Ambient Intelligence and Humanized Computing, 2020, 1-10. dhttps://doi.org/10.1007/s12652-020-02368-2
22. Seghir, F., Khababa, G. Fuzzy Teaching Learning Based Optimization Approach for Solving the Qos-aware Web Service Selection Problem in Uncertain Environments. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(12), 10667-10697. https://doi.org/10.1007/s12652-020-02879-y
23. Singh S., Chana, I., Singh, M. The Journey of QoS-aware Autonomic Cloud Computing. IT Professional, 2017, 19(2),42-49. https://doi.org/10.1109/MITP.2017.26
24. Totsenko, V. G. On Problem of Reversal of Alternatives Ranks while Multicriteria Estimating. Journal of Automation and Information Sciences, 2006, 38(6), 1-11. doi.org/10.1615/J Automat Inf Scien.v38.i6.10
25. Trueman, T. E., Narayanasamy, P., Ashok Kumar, J. A Graph-based Method for Ranking of Cloud Service Providers. The Journal of Supercomputing, 2022, 78(5), 7260-7277. https://doi.org/10.1007/s11227-021-04156-x
26. Varghese, B., Buyya, R. Next Generation Cloud Computing: New Trends and Research Directions. Future Generation Computer Systems, 2018, 79, 849-861. https://doi.org/10.1016/j.future.2017.09.020
27. Youssef, A. E. An Integrated MCDM Approach for Cloud Service Selection Based on TOPSIS and BWM. IEEE Access, 2020, 8, 71851-71865. https://doi.org/10.1109/ACCESS.2020.2987111
28. Zeng, S., Xiao, Y. A Method Based on TOPSIS and Distance Measures for Hesitant Fuzzy Multiple Attribute Decision Making. Technological and Economic Development of Economy, 2018, 24(3), 969-983. https://doi.org/10.3846/20294913.2016.1216472

29. Zhang, N., Yang, P., Zhang, S., Chen, D., Zhuang, W., Liang, B., Shen, X. S. Software Defined Networking Enabled Wireless Network Virtualization: Challenges and Solutions. IEEE Network, 2017, 31(5), 42-49. https://doi.org/10.1109/MNET.2017.1600248

30. Zhang, N., Yang, P., Ren, J., Chen, D., Yu, L., Shen, X. Synergy of Big Data and 5g Wireless Networks: Opportunities, Approaches, and Challenges. IEEE Wireless Communications, 2018, 25(1), 12-18. https://doi.org/10.1109/MWC.2018.1700193