# Instance based Multi Criteria Decision Model for Cloud Service Selection using TOPSIS and VIKOR

Deepti Rai * and Pavan Kumar V **
*Sahyadri College of Engineering
dee.rai@gmail.com
**Sahyadri College of Engineering
pavan.cs@sahyadri.edu.in

**Abstract:** Cloud computing has been incorporated in almost every fold of customer's world today. As more challenging becomes the world of a resource consumer, so does the plethora of numerous cloud vendors. Every vendor tries to sell off their services with different attractions. Some of them are free, some as pay-as-you-go services and some at rented structures. With so many choices available to a random user, making the right choice about the type of vendor is a crucial decision. We present in this paper novel methods for cloud service selection. The methods derives from the traditional selection methods with emphasis on user criteria weights. Inherent comparisons have been conducted amongst the various methods to help in analyzing and finalizing a broker architecture for selection of the best service provider among the contending cloud vendors.

**Keywords**: cloud computing, cloud service selection, cloud user weights, TOPSIS, VIKOR.

## Introduction

Some of the best works done so far cover various Multi Criteria Decision Making methods like TOPSIS[1] , ELECTRE[14], VIKOR[3], PROMETHEE[5], AHP[6] etc. with new age additions to user preferences for fuzzy or trusted requirements[12]. Having come across various work done using these methods, it is highly imperative to understand that all these are time tested and efficient in one scenario or other. However most of these methods are applied to inherent ready set of data on the whole. The data set though complete with past values, seldom explores the individual performances of the cloud providers on an instance basis. In this paper we propose a method where we concentrate on the instance based outcomes of the service provider. To do this, we select the most accepted 2 methods – TOPSIS and VIKOR for our service selection of IaaS Clouds. We shall then categorize the inputs in two directions. First by altering the way the functional requirements are taken and secondly the way how the user weights are assigned. All these shall be realized using a simulation tool – Cloudsim[7] before concluding on the final result.

### Existing System

Existing system is a system which has been time tested and proven efficient with any or other major changes. This system may be working or just in process of being setup. In either case, an existing system comes with its own set of shortcomings and problems. Hence there is always scope of improvement to bring about better systems. For the juncture of this project, the existing system is a system where we have basically three parties. As the entire system is based on cloud computing, needless to say we have two major sides - Cloud Providers and Cloud Users. In general terms, Cloud Provider is the party which provides cloud services like hosting, storage, application setup, service setup etc. On the other hand, cloud users is the party which avails these services by either paying or by virtue of free servicing. Now as a facilitator between these 2 parties, we have a broker[18] in the middle. The broker is essentially a third party, which deals with aggregation - bringing together all the required services of the user under one banner, de-duplication - by removal of redundant data available in the cloud data sets and security - by providing a trusted environment for users to upload their applications. The brokers themselves are subjected to trust evaluation as they are third party and have their own gain as primary objective. Some Service Level Agreement negotiations are also done by brokers by drawing up a contract with both the parties. The existing system that uses brokers for all such activities is also extended for the usual service selection decision problem. The cloud service selection problem is a multi criteria problem with many decisions[19]. Hence it is done in multiple layers. While the broker goes about accumulating the metrics from the cloud service providers, the data set becomes huge in nature. Thereafter the broker applies any of the multiple criteria decision modeling methods like TOPSIS, VIKOR, PROMETHEE, AHP etc. to through its software and filters out the best cloud services. This system takes into account data metrics collected over a period of time as whole. Thereafter the decision modeling is applied to the entire data set and result is generated. This system requires maintenance of huge database which holds the criteria metrics. The averaging of the weighted criteria method implemented is basically very general in nature. It has advantages and disadvantages to its credit.

**Multi Criteria Decision Making Model**

When it comes to the cloud service selection problem, we deal with multiple vendors with multiple services and varied user preferences[2]. Hence they automatically fall into the MCDM category. As with any MCDM problem[16], we can have numerous approaches like MAUT methods, AHP, French Outranking methods and Russian ordinal methods[8]. Every method has its strong and weak areas. While some methods work well with small data set, some are very effective with large data sets like that of a cloud service provider.[13] If we were to take a generic example of cloud services along with the user wanted criteria, then with the inclusion of user preference the problem of cloud service selection becomes very difficult[11]. Also another important factor is the consideration of these aspects both in real time and also past performance[4]. When a comparison is made between clouds, it is highly preferred to have a trust factor built into the decision. This factor can be obtained by observing the performance of the said criteria over a period of time and not in an instant.[10]. For doing this, we have to micro calculate the best cloud services in parts i.e one instance of a time. This method keeps the selection more accurate in comparison to the selection from the average data set values.

To summarize, despite many MCDM methods applied to cloud service selection, the ever changing nature of clouds and their quality of service criteria with time has not been incorporated effectively. Hence the existing approaches are not completely accurate in determining the best service provider. This paper considers the different aspects of time in past and present. While MCDM methods are the most suitable to sort out multi criteria problems[20], they themselves are inefficient in giving real time answers to user requirement. Experiments done to validate our method and outcome are an intelligent, resourceful and practical way of getting the cloud selection from user point of view.

## Proposed Model

The proposed method of this selection model is simple in terms of the components involved. To start with we have the computing environment with different cloud Datacenters[9]. These can be considered as either cloud service providers or instances of CSPs. These are simulated as an IaaS with core computing qualities. They are responsible for publishing their services and respective paradigms. This service related information is stored in a database for further use. Next we have the user group who call in the cards by specifying which criteria of the service is of importance to them. This can be realized with either assigning weights with variance method or by fuzzy weights by asking the user his/her level of importance of each criterion in relation to other. In between these 2 categories we have the main decision maker – the broker who is responsible for employing the decision making algorithm to get us the best service as the result. Traditional methods involve simple concept of taking an absolute average of all criteria values and applying MCDM to it, however very accepted, this method is not free of flaws when dealing with a huge data set of criteria values over a considerably long period. Any Cloud provider can vary with its services over a period of time. We may have a trusted cloud server to be performing poorly in the recent past or vice versa. In order to get the most unbiased measure of all the criteria, it is imperative to give more weightage to the recent past. Hence the algorithm is applied on every instance(daily) of the values received and then the overall result is computed as depicted in Fig.2. We can dig further and calculate the best provider on quarter hourly or hourly basis, however with the huge amount of data produced, and citing no major upheavals like the stock market, daily basis is sufficient enough. Once the details of the CSPs are obtained, the users are asked to weigh in the criteria as per their need. Thereafter any of the MCDM algorithms mentioned below are applied to get the best service for daily set of values. For this experiment we have chosen 2 MCDM algorithms - TOPSIS and VIKOR. Both of them have proven their efficiency and worth in the area of multi criteria decision making.

**Advantages of Proposed Model**

- The disadvantage of normal weighted average method of being sensitive to extreme values, is overcome by calculating instance based averages rather than average of the whole system. Especially when leading to dynamics of cloud, we come across extreme values which when averaged on the whole can give an inconsistent value.
- Unlike normal average method this proposed method is best suited for time series type of data. The concept of time series data comes into picture where there is diversity and volatility is involved. This type of data which is inherent to the cloud platform, is best suited to be calculated instance based rather than as a bundle.
- The proposed method works even when all values are not equally important. Especially when it comes to clouds, we have metrics of criteria ranging over a wide area. Hence Not all are equally important. The criteria have their importance marked by the user given weight. Depending on user preference, the weights are assigned to the metrics. Hence the proposed method is most suited for unequal data sets.
- Proposed method works faster in comparison to the existing method as the average is calculated on a smaller sample of data and aggregated finally in one big step. For huge databases, the minor upheavals in time can result in major performance benefit.

**Architecture**

The architecture of this selection model is simple in terms of the components involved as in Fig 1. To start with we have the computing environment with different cloud Datacenters[9]. These can be considered as either cloud service providers or instances of CSPs. These are simulated as an IaaS with core computing qualities. They are responsible for publishing their services and respective paradigms. This service related information is stored in a database for further use. Next we have the user group who call in the cards by specifying which criteria of the service is of importance to them. This can be realized with either assigning weights with variance method or by fuzzy weights by asking the user his/her level of importance of each criterion in relation to other. In between these 2 categories we have the main decision maker – the broker who is responsible for employing the decision making algorithm to get us the best service as the result.

Traditional methods involve simple concept of taking an absolute average of all criteria values and applying MCDM to it, however very accepted, this method is not free of flaws when dealing with a huge dataset of criteria values over a considerably long period. Any Cloud provider can vary with its services over a period of time. We may have a trusted cloud server to be performing poorly in the recent past or vice versa. In order to get the most unbiased measure of all the criteria, it is imperative to give more weightage to the recent past. Hence the algorithm is applied on every instance(daily) of the values received and then the overall result is computed. We can dig further and calculate the best provider on quarter hourly or hourly bass, however with the huge amount of data produced, and citing no major upheavals like the stock market, daily basis is sufficient enough.

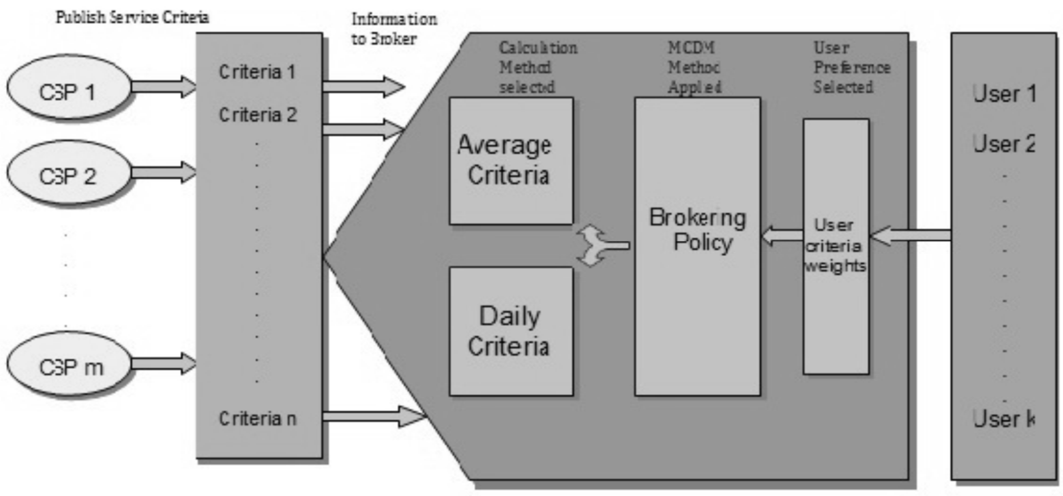


Figure 1 : Architecture of Proposed Model

**Phases of Proposed System Setup**

*Phase 1- Cloud Setup Stage*

This stage comprises of setting up an evaluation system which if not for real, atleast mimics the the original cloud environment. For the evaluation of the proposed system, a simulation environment has been devised. This has been realized using tool called Cloudsim[9]. This tool uses classic java based object structure for cloud setup. Major component of this tool is the availability of smaller classes which simulate datacentres,hosts,virtual machines etc. To start with there is a cloud information service. It is a registry which has the resources listed in cloud. After registry of all the distinct objects, individual datacenters are created. Datacenters are synonymous to clouds themselves. Hosts under each datacenter has some hosts, which has some hardware characteristics. Next there will be virtual machines which perform cloud related tasks. A broker is responsible for allocating tasks to datacenter. This is done using the brokering policy which follows any of the several algorithms available. The jobs themselves are called cloudlets, which are assigned to virtual machines/datacenter by broker.

*Phase 2 - Average Model Selection*

This stage consists of the selection of the main method of approach. As mentioned earlier, there already exists systems with average weighted based decisions. Keeping in mind their limitations, where the relevance of time is not taken into consideration, a new method has been devised. In this method, the time is of essence. The farther away we go into the past, the weightage factor reduces. This phenomenon of weight decay can be found in almost all systems which base their decisions on cumulative time based metrics. This decay can be calculated stochastically or logarithmically as per requirement. In the new method, the metrics are subjected to calculation based on the instance selected. Instance can be any

**Flowchart for the Proposed System**

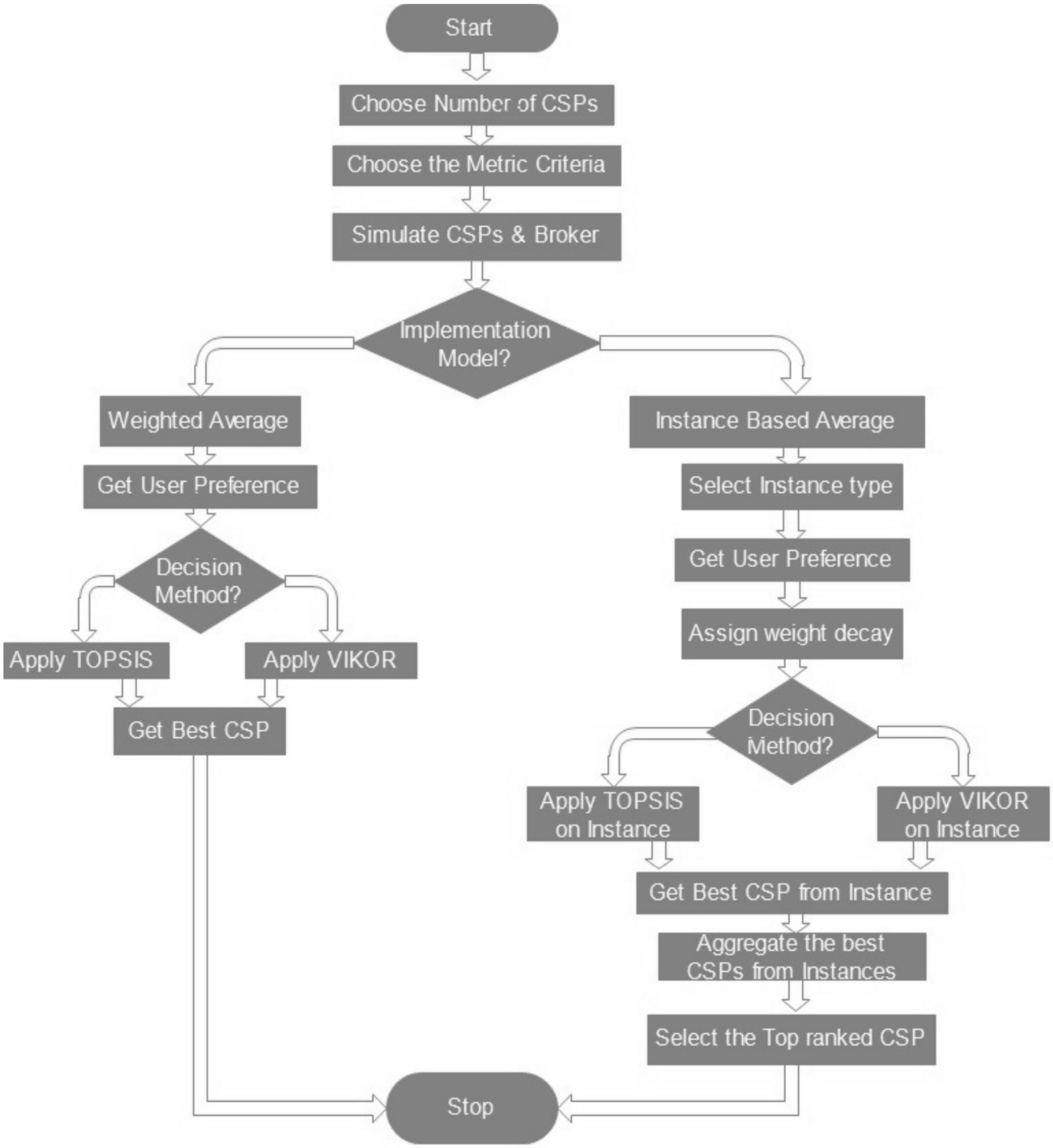


Figure 2: Flowchart for the Proposed System

single point of reference, weeks, days, hours, minutes etc. To depict the way the system works, a user is given access to the option to select any of the model required - either the average weighted or instance weighted.

*Phase 3 - Decision Algorithm Selection Stage*

After the model is selected, next step is to select the decision algorithm[17]. For the purpose of this evaluation - Topsis and Vikor have been selected. They have both proven their worth in getting the best decision out of the huge datasets of data. Two options have been given for comparison of results and data so that both the algorithms can be applied independently and evaluation be done fairly. Out of the 2 which needs to be applied is again sole decision of the user. To these algorithms we feed the cloud metric data with user preferences, weight decay etc. These perform normalization of the metrics and calculate the ranking of the cloud service providers as outcome.

*Phase 4* **-** *Cloud Ranking and Selection*

The result derived from the previous step becomes the base for the user decision. The clouds are ranked in the descending manner and the one at the top becomes the best cloud. This is the final step of the entire evaluation process. The outcome may change depending on the model used or depending on the algorithm used. Also the user preferences change every now and then, resulting in variation of results.

**Algorithms**

*TOPSIS* **-** *Technique for Order of Preference by Similarity to Ideal Solution*

This method as depicted in algorithm Fig.3. works by calculating the geometric distance from the ideal solution. It takes into account the weight of each criteria, normalizing it and then determining the distance to ideal solution. Linear Normalization is performed to the metric data. The Technique for Order Preference by Similarity to Ideal Solution (Hwang and Yoon, 1981) identifies the best alternative among the one having the shortest distance from ideal solution. This method calculates the distance from the ideal alternative and the distance from negative option combining the worst performances of alternatives with respect to the single criterion. This method does not have many checks for acceptability as compared to other methods. The major advantage of this method is the admissibility of various types of criteria and various types of metric in varied ranges. TOPSIS outperforms other methods by being simple and straightforward. It uses euclidean distance method to calculate the distance, hence differentiating between individual and total satisfaction.

*VIKOR* **-** *VIseKriterijumska Optimizacija I Kompromisno Resenje*

The method Algorithm Fig.4. is that it selects the compromised solution among the other alternatives. Here the ranking is done using step by step checking where first best option is compared to second best option. It is also compared to the overall best alternative. Hence it rejects the solution if these checks are not passed. However it may give the solution of compromise if required. This method works excellently with conflicting criteria. Especially where criteria has been long drawn and selecting the best among so many different variable criteria makes this the most useful method. However since it does checking step by step, it becomes highly probable that the end result may be without solution. This method is sometimes extended to be used with other such methods like Analytic Hierarchy Process(AHP) and with fuzzy input values. The algorithm has proven efficient for large datasets where varied datatypes and criteria range are used.

**Algorithm 1** TOPSIS algorithm

**procedure** TOPSIS
  Identify the alternatives $\epsilon\mathbb{N}$
  **for** $x_{ij}\epsilon D(m,n)$ of $m*n$ **do**,
    obtain $x_{ij}$ where $i = 1,2,.,m, j = 1,2,.n$
  **end for**
  **for** $x_{ij}\epsilon D(m,n), i = 1$ to $i = m$ and $j = 1$ to $j = n$ **do**,
    Compute $N^*_{ij} = N_{ij}/\sqrt{\sum_{i=1}^{n} N_{ij}^2}$
  **end for**
  Calculate Variance of weights $V_j = (1/n)\sum_{i=1}^{n}(N^*_{ij} - (N^*_{ij})_{mean})^2$
  **for** $i = 1$ to $i = m$ **do**
    Obtain weights $W_j = V_j/\sum_{i=1}^{m} V_j$ and $\sum_{j=1}^{m} w_j = 1$
  **end for**
  **for** $i = 1$ to $i = m$ and $j = 1$ to $j = n$ **do**
    Compute Weighted normalized matrix $WV_{ij} == W_j * N_i$
  **end for**
  Determine best and worst ideal solutions as
      $A^+ = \{a_1^+, .., a_m^+\} i.e (maxWV_{ij})|(minWV_{ij})$
      $A^- = \{a_1^-, .., a_m^-\} i.e (minWV_{ij})|(maxWV_{ij})$
  Obtain separation of each alternative from $A^+$ and $A^-$ as below
      $D_i^+ = \sqrt{\sum_{j=1}^{m}(a_{ij} - a_j^+)^2}$
      $D_i^- = \sqrt{\sum_{j=1}^{m}(a_{ij} - a_j^-)^2}$
  Obtain similarity Index $C_i^* = D_i^-/(D_i^+ + D_i^-)$
  Choose best alternatives in increasing order of $C_i^*$
**end procedure**

Figure 3: TOPSIS Algorithm

**Algorithm 2** VIKOR algorithm

**procedure** VIKOR
  Identify the alternatives $\epsilon\mathbb{N}$
  **for** $x_{ij}\epsilon D(m,n)$ of $m*n$ **do**,
    obtain $x_{ij}$ where $i=1,2,.,m,j=1,2,.n$
  **end for**
  **for** $x_{ij}\epsilon D(m,n)$,$i=1$ to $i=m$ and $j=1$ to $j=n$ **do**,
    Compute $N^*_{ij}=N_{ij}/\sqrt{\sum_{i=1}^{n}N_{ij}^2}$
  **end for**
  Calculate Variance of weights $V_j=(1/n)\sum_{i=1}^{n}(N^*_{ij}-(N^*_{ij})_{mean})^2$
  **for** $i=1$ to $i=m$ **do**
    Obtain weights $W_j=V_j/\sum_{i=1}^{m}V_j$ and $\sum_{j=1}^{m}w_j=1$
  **end for**
  **for** $i=1$ to $i=m$ and $j=1$ to $j=n$ **do**
    Compute Weighted normalized matrix $WV_{ij}==W_j*N_i$
  **end for**
  Obtain Maximum Criterion Weight and Minimum Criteria Weight as below:
      $F^+_{ij}=\max(N_{ij})$
      $F^-_{ij}=\min(N_{ij})$
  Compute Utility Measure
      $U_i=\sum_{i=1}^{m}W_j(F^+_j-F_{ij})/(F^+_j-F^-_j)$
  Compute Regret Measure
      $R_i=max[W_j(F^+_j-F_{ij})/(F^+_j-F^-_j)]$
  Calculate the Vikor Index
      $VI_i=v((U_i-U^-)/(U^+-U^-))+(1-v)((R_i-R^-)/(R^+-R^-))$
  Choose best ranking alternative in increasing order of $VI^*_i$
**end procedure**

Figure 4: VIKOR Algorithm

## Results

### Discussion

The idea basically deals with 2 different models - one being with the average method and other being Instance Method. The implementation and comparison of the first method has been completed at this juncture. The cloud creation has been completed using cloudsim toolkit. The system developed can create cloud structures with 3 criteria mark values for RAM, Bandwidth and Storage. Number of clouds to be created is kept with the admin of the system. Below referenced Table 1, Table 2, Table 3, Table 4, Table 5, Table 6 show the results of the experiment.[15]. Along with this a broker is also created. Once the clouds are created with the metrics, they are then stored in the sql database for future use. Immediately following this, the traditional method for cloud selection has been established which compares TOPSIS and VIKOR in terms of evaluation time for arriving at the best cloud. \par While evaluating both Topsis and Vikor, it was found that evaluation with a collection of 10 clouds randomly created by the cloudsim, vikor performed better than topsis in terms of evaluation time. Also when it came to memory vikor was better than topsis. As the data is collected over a period of 30 days, the dataset is sufficient to establish the fact that though both the algorithms are excellent choices for cloud service selection, vikor outperforms topsis. Next would be development of the Instance based model using real cloud data value sets and instances of days. This would be to apply averages on individual days rather than entire set, giving a microscopic and accurate result.

Table 1: Average Criteria Values

| Cloud | RAM | Bandwidth | Storage |
|---|---|---|---|
| Cloud1 | 1433 | 18000 | 360000 |
| Cloud2 | 1638 | 22000 | 160000 |
| Cloud3 | 1843 | 22000 | 240000 |
| Cloud4 | 1433 | 18000 | 360000 |
| Cloud5 | 1843 | 21000 | 480000 |
| Cloud6 | 1433 | 20000 | 320000 |
| Cloud7 | 1024 | 12000 | 300000 |
| Cloud8 | 1433 | 23000 | 360000 |
| Cloud9 | 1740 | 18000 | 340000 |
| Cloud10 | 1331 | 28000 | 360000 |

Table 2: Evaluation Matrix

| Cloud | RAM | Bandwidth | Storage |
|---|---|---|---|
| Cloud1 | 0.059 | 0.083 | 0.168 |
| Cloud2 | 0.068 | 0.101 | 0.075 |
| Cloud3 | 0.076 | 0.101 | 0.112 |
| Cloud4 | 0.059 | 0.083 | 0.168 |
| Cloud5 | 0.076 | 0.097 | 0.224 |
| Cloud6 | 0.059 | 0.092 | 0.15 |
| Cloud7 | 0.042 | 0.055 | 0.14 |
| Cloud8 | 0.059 | 0.106 | 0.168 |
| Cloud9 | 0.072 | 0.083 | 0.159 |
| Cloud10 | 0.055 | 0.129 | 0.168 |

Table 3: Similarity Index using TOPSIS

| Cloud | Similarity Index |
|---|---|
| Cloud1 | 0.446 |
| Cloud2 | 0.449 |
| Cloud3 | 0.530 |
| Cloud4 | 0.446 |
| Cloud5 | 0.433 |
| Cloud6 | 0.473 |
| Cloud7 | 0.432 |
| Cloud8 | 0.485 |
| Cloud9 | 0.431 |
| Cloud10 | 0.535 |

Table 4: Vikor Index using VIKOR

| Cloud | Vikor Index |
|---|---|
| Cloud1 | 0.341 |
| Cloud2 | 0.909 |
| Cloud3 | 0.605 |
| Cloud4 | 0.341 |
| Cloud5 | 0.0 |
| Cloud6 | 0.445 |
| Cloud7 | 0.729 |
| Cloud8 | 0.269 |
| Cloud9 | 0.35 |
| Cloud10 | 0.216 |

Table 5: Proposed Daily Method

| Days | TOPSIS | VIKOR |
|---|---|---|
| 1 | cloud6 | cloud9 |
| 2 | cloud5 | cloud6 |
| 3 | cloud2 | cloud2 |
| 4 | cloud9 | cloud6 |
| 5 | cloud8 | cloud2 |
| 6 | cloud2 | cloud2 |
| 7 | cloud1 | cloud7 |
| 8 | cloud2 | cloud8 |
| 9 | cloud6 | cloud2 |
| 10 | cloud2 | cloud6 |
| 11 | cloud1 | cloud8 |
| 12 | cloud9 | cloud2 |
| 13 | cloud2 | cloud8 |
| 14 | cloud5 | cloud6 |
| 15 | cloud2 | cloud2 |
| 16 | cloud5 | cloud10 |
| 17 | cloud2 | cloud2 |
| 18 | cloud9 | cloud2 |
| 19 | cloud7 | cloud5 |
| 20 | cloud3 | cloud6 |
| 21 | cloud2 | cloud2 |
| 22 | cloud8 | cloud2 |
| 23 | cloud2 | cloud1 |
| 24 | cloud7 | cloud8 |
| 25 | cloud1 | cloud2 |
| 26 | cloud10 | cloud7 |
| 27 | cloud2 | cloud2 |
| 28 | cloud9 | cloud9 |
| 29 | cloud2 | cloud2 |
| 30 | cloud2 | cloud2 |

Table 6: Result Comparison

| Average Method | | Daily Method | |
|---|---|---|---|
| TOSIS | VIKOR | TOPSIS | VIKOR |
| cloud10 | cloud2 | cloud2 | cloud2 |

## Conclusion

In conclusion , we assert that the method proposed by us is efficient and practical in terms of real time and changing cloud environments. This method keeps into account the instance based calculation of metrics and hence gives a more focused result than the assumptions based on entire average of metrics in whole. As we have shown the results for a small group of data, this method can be tested against a fairly large dataset without issues. Enhancement of this method could be to include fuzzy weights and future predictions of criteria change. Also some important parameters like the cost, vendor lock in, data disruption etc. have not been considered, which can be included in this model.

## References

[1] Serafim Opricovic, Gwo-Hshiung Tzeng, “Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS”, European Journal of Operational Research, Elsevier, Volume 156, Issue 2, July 2004, Pages 445–455.

[2] R.Buyya, C.S.Yeo, S. Venugopal,J. Broberg,I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision,Hype,and Reality for Delivering Computing as the 5th Utility” , Future General Computer System, IEEE, Vol. 25, June 2009, pp. 599–616.

[3] Lisa Y. Chen, Tien-Chin Wang “Optimizing partners’ choice in IS/IT outsourcing projects: The strategic decision of fuzzy VIKOR”,International Journal of Production Economics,Volume 120, Issue 1, July 2009, Pages 233–242.

[4] Yager, R.R., “On ordered weighted averaging aggregation operators in multicriteria decision making”, IEEE Transactions on Systems, Man and Cybernetics, Volume:18 ,Issue: 1, 1998, pp .183 - 190.

[5] Toinard, C., Ravier, T., Cerin, C., Ngoko, Y., “The Promethee Method for Cloud Brokering with Trust and Assurance Criteria”, IEEE, Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015, pp 1109 - 1118.

[6] Rajanpreet Kaur Chahal , Sarbjeet Singh, “AHP-Based Ranking of Cloud-Service Providers” , Springer, Advances in Intelligent Systems and Computing, Vol 443, 2016, pp 491-499.

[7] Buyya, R., Ranjan, R., Calheiros, R.N., “Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities”, IEEE, Conference proceedings on High Performance Computing and Simulation, 2009, pp 1-11.

[8] S. Sindhu, Saswati Mukherjee, “Efficient Task Scheduling Algorithms for Cloud Computing Environment”, Springer, High Performance Architecture and Grid Computing, Vol 169, 2011, pp 79-83.

[9] Wang Long, Lan Yuqing , Xia Qingxin, “Using CloudSim to Model and Simulate Cloud Computing Environment”, IEEE, Conference proceedings of Computational Intelligence and Security (CIS), 2013, pp 323 - 328.

[10] Treadgold, N.K., Gedeon, T.D. “Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm”, IEEE Transactions on Neural Networks, Vol:9, n.4, 1998, pp 662 – 668.

[11] Johan Tordssona, Rubén S. Monterob, Rafael Moreno-Vozmedianob, Ignacio M. Llorenteb, “Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers”, Elsevier, Future Generation Computer Systems, Vol 28, n.2, February 2012 pp 358–367.

[12] Wang, T.-C., Lee, H.-D., Chang, M.-S. “A fuzzy TOPSIS approach with entropy measure for decision-making problem”, International Conference on Industrial Engineering and Engineering Management, IEEE, 2007, pp. 124-128.

[13] Shangguang Wang , Zhipiao Liu , Qibo Sun , Hua Zou , Fangchun Yang, “Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing”, Journal of Intelligent Manufacturing, Vol.25 n.2, April 2014, pp.283-291.

[14] Luis Del Vasto-Terrientesa, Aida Vallsa, Roman Slowinskib c, Piotr Zielniewiczb, “ELECTRE-III-H: An outranking-based decision aiding method for hierarchically structured criteria”, Elsevier, Expert Systems with Applications, Vol 42, n. 11, 2015, Pages 4910–4926.

[15] Deepti Rai, Pavan Kumar V, “Instance based multi criteria decision-model for cloud service selection using TOPSIS and VIKOR”, IAEME, International Journal of Computer Engineering and Technology , Vol 7 , n.2, 2016, pp 78-87.

[16] Gang Kou , Yong Shi , Shouyang Wang, “Multiple criteria decision making and decision support systems”, Decision Support Systems, v.51 n.2, May 2011, p.247-249.

[17] Tao Yu, Kwei-Jay Lin, “Service selection algorithms for Web services with end-to-end QoS constraints”, Springer, Information Systems and e-Business Management, Vol 3, n.2, July 2005, pp 103-126.

[18] S Sundareswaran., A.quicciarini, D.Lin, “A Brokerage-Based Approach for Cloud Service Selection”,IEEE,Conference proceedings from Cloud Computing , June 2012, pp 558-565.

[19] Saurabh Kumar Garga, Steve Versteegb, Rajkumar Buyyaa, “A framework for ranking of cloud computing services”, Future Generation Computer Systems, Vol 29, Issue 4, June 2013, Pages 1012–1023.

[20] Md Whaiduzzaman, Abdullah Gani, Nor Badrul Anuar, Muhammad Shiraz, Mohammad Nazmul Haque, Israat Tanzeena Haque, “Cloud Service Selection Using Multicriteria Decision Analysis”, Hindawi, The Scientific World Journal, 2014.