A New Method for Web Service Recommendation Based on QoS Prediction

Mahmoud A. Salam

Information Technology Department, Faculty of Computers and Information, Mansoura University, Egypt

Email: Mahmoud20@mans.edu.eg

Abstract

As service-oriented architecture gains in popularity and grows in popularity, Web service recommendation and composition have become more important topics for research. Accurately predicting individualized QoS recommendations for recommending web services is a difficult task because of the inconsistency of the Internet and the scarcity of information regarding QoS history. Our team suggests a new framework for QoS values prediction and presents two methods for clustering, **User_BC**, and **Service_BC**, to support QoS prediction accuracy. Hierarchical clustering is used, based on the QoS dataset of PlanetLab1 (that) contains 200 service-user response time values, with 1,597 service values overall. In our research, we've found that our clustering-based methods beat other popular algorithms in detailed experimental comparisons and analyses.

Keywords: Recommendation; WS; PSO; QoS prediction; clustering

1. Introduction

In the design of distributed systems and applications, service-oriented architectures that feature Web services [1] with invocations via machine-to-machine interaction [2] are quickly becoming common. Web services, which are computational Web components, are specifically designed to support these kinds of interactions. On the Internet, there are 28,606 public Web services and 7,739 service providers as counted by seekda.com. It is a significant duty to evaluate and recommend multiple functional equivalents of Web services because the numbers are expanding rapidly. To address this problem, the non-functional properties of Web services (i.e., throughput, response time, failure frequency, etc.) are often described and evaluated using Quality of Service (QoS) [2]. Web service composition, sometimes abbreviated as WS composition, is a process of selecting and aggregating multiple existing services to create a new additional function service with a decent quality of service (QoS) performance and a variety of service user requirements. Yet, in service-oriented systems, the unpredictable Internet environment and where users are located dramatically affect how well they perform. While a given web service might provide the same quality to everyone, the specific experience of service users varies widely. Applications of Web services are enhanced when QoS-aware techniques are employed to assist in selecting from candidate services to give improved performance [2]. Research on QoS-guided personalized service recommendations have become quite important and compelling in recent years. There are several options for choosing a service. In general, monitoring and assessing all potential options is the simplest method. While this is feasible in theory, many web service invocations may come with expensive charges that would cost customers a great deal. Furthermore, it's a waste of time and money to make all the

DOI: https://doi.org/10.54216/JISIoT.000101

Web services execute every time because there are plenty of service options that are almost identical [3]. To resolve this difficulty, numerous different QoS prediction methods for web service recommendation have been invented, as mentioned in [4], [5], [6], and [7]. The primary methodology consists of using algorithms named collaborative filtering (CF) to simulate the similarity between the service over the web and the user to anticipate unknown values of QoS through information sharing. But these techniques strongly rely on knowledge of the Web service invocation history. There are just a few web services each user utilizes in each invocation, which makes the availability of QoS data very sparse. On top of that, certain outdated QoS values aren't updated in real-time or even historically, meaning that their data on the Internet's changing environment is inaccurate.

This research introduces a new clustering-based framework for making quality of service (QoS) predictions, using a series of known points (computers) over the Internet. This design helps combat the problem of data sparsity. Landmarks can use existing web services to continually get more accurate data about QoS performance. We employ clustering algorithms to help service users make QoS predictions with the use of real-time information of QoS about a reference of landmarks. Our real-world QoS dataset, which includes the results of 200 service users' response times for 1,597 Web services, was used for a variety of research. Experimental findings prove that our clustering-based collaborative filtering algorithm outperforms other similar ones. The following are the paper's major sections: In Section II, the associated work is presented. Section III illustrates our web service recommendation mechanism based on landmarks. Two clustering based QoS prediction techniques are described in Section IV. Section V contains discussions of the experiments and findings. Section VI, the final section, wraps up the paper.

2. Related work

A lot of research has recently been done in recommending which web service to use. The researchers that have made their work available online have found numerous ways, such as those cited below: [8], [9], [10]. Service users are using up a lot of time and resources trying to monitor all the available Web services because Web service invocations are too expensive and there are too many to consider. And therefore, the proposal of a web service QoS prediction receives a lot of interest in the study. Web service QoS prediction has the aim of estimating the missed QoS values between services over the web and users using partial information [2]. To conclude, it is advised that a service whose QoS value is at most can be assembled for use by the user. One of the most used algorithms for making recommendations in commercial systems, called collaborative filtering, is presented to create QoS predictions for recommendations of web services. [12], [7], [6], [5], [11], [1]. The approach in Shao et al. [7] relies on the similarities between service users to forecast QoS values. To anticipate the QoS of service users by utilizing accessible QoS data from their neighbors. WSRec [6] offers a hybrid collaborative filtering based QoS prediction solution that utilizes similarities between Web service objects and users. The approach was found to yield good overall prediction accuracy. WSRec is defined as follows. Using the Pearson Correlation Coefficient (PCC) [6], first, similarities between users and service items over the web are estimated using historical QoS data. If this happens, all users will be able to locate the Top-K-like users in their friend groups, and all Websites will be able to discover Top-K-like products. Consequently, the user-based collaborative filtering methods will employ a user's (who has previously used a Web service) usage data to estimate its missed QoS values, whereas the methods of collaborative filtering will use the following equations to estimate the missed values of QoS:

$$p_u = \bar{u} + \frac{\sum_{a \in S_u} s(u_a, u)(q_{u_a i} - \bar{u}_a)}{\sum_{u_a \in S_u} s(u_a, u)}$$
$$p_i = \bar{\imath} + \frac{\sum_{i_k \in S_i} s(i_k, i)\left(q_{u i_k} - \bar{\imath}_k\right)}{\sum_{i_k \in S_i} s(i_k, i)} \qquad \textbf{(1)}$$

Where $\bar{u}$ and $\bar{\imath}$ refer to the average value of QoS of user u and the service l and. The similarity values between services and users are denoted by $s(u_a, u)$ *and* $s(i_k, i)$. **Top − K set of stuff and users *are denoted by*** S_u and $S_i$are the, and S_u and S_i's lists of similar users and items User and item QoS values of the same kind are represented by u and i. q_{ui} is a table of QoS data that include both users and items. With service similarity and user similarity, p_i and p_u are the projected QoS values WSRec plans to integrate service-based and user-based strategies to increase prediction accuracy. Finally, the equation below is used to calculate the prediction:

DOI: https://doi.org/10.54216/JISIoT.000101

$$p = w_u P_u + w_i P_i \quad (2)$$

The linear combination of $\boldsymbol{P_u}$ and $\boldsymbol{P_i}$ is the hybrid collaborative filtering method, and the coefficients are $\boldsymbol{w_u}$ and $\boldsymbol{w_i}$, where $\boldsymbol{w_i + w_u = 1}$. Collaborative filtering systems mainly rely on past QoS data, although this approach also has drawbacks. Web service QoS predictions and recommendations are potentially erroneous, as past QoS data may be limited or out of date. To solve this difficulty, we are proposing a new clustering based QoS prediction approach in this study. Our method has an algorithm for each major Web service that looks at predefined stable points (landmarks) that gauge all services' QoS and provide a static record. The related landmarks' QoS information is used to predict users' QoS values after clustering. To make forecasts, service users don't need to bring any historical QoS information with them. We have shown with our experiments that our technique is better than others.

3. WEB SERVICE RECOMMENDATION FRAMEWORK

To better accommodate restrictions in collaborative filtering, we designed a landmark based QoS prediction system, which is shown in Figure 1. It's obvious from the graphic that the web service recommendation architecture consists of the following: **1)** The landmarks are put on the Internet and make regular calls to measure the QoS information of the available Web services. **2)** Assign landmarks to groups based on their quality data. **3)** A user from each service group determines latency to landmarks and is then classified into one of three groups. **4)** Forecasts of QoS are made using landmark QoS information. **5)** The effect of prediction-based QoS-aware Web service selection and recommendation is that it enables users to be aware of their application's behavior.

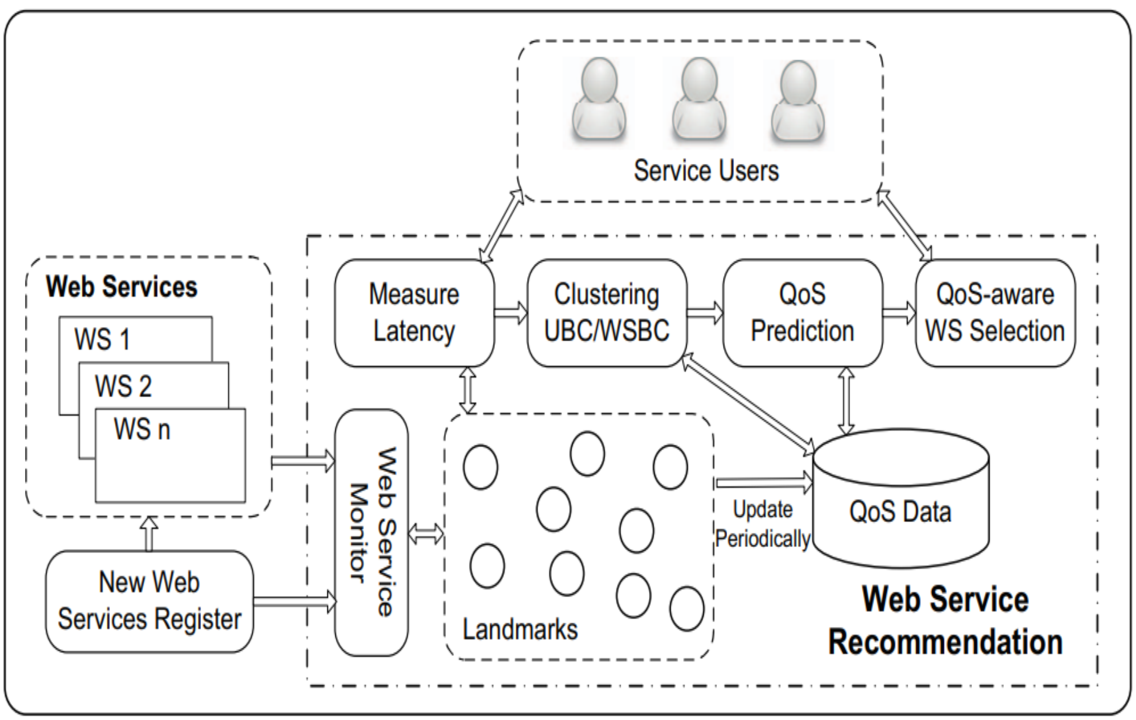

Figure 1: Framework of service recommendation

DOI: https://doi.org/10.54216/JISIoT.000101

4. QOS PREDICTION ALGORITHM

4.1 Landmarks Clustering

The set $\{\boldsymbol{X_i}$, I = 1, 2, ⋯, $\boldsymbol{N_L}\}$ is the landmark set, and we indicate it by how its landmarks are spread on the Internet. To observe the Web service and occasionally update the QoS levels, we'll rely on the landmarks. Measurement is easy since landmarks are few, and they are small in size. This provides an indicator of how well the websites' responses will perform. The equation used to explain the significance of this new data is as follows:

$$Q_L = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1W} \\ q_{21} & q_{22} & \cdots & q_{2W} \\ \vdots & \vdots & \cdots & \vdots \\ q_{N_L 1} & q_{N_L}2 & \cdots & q_{N_L W} \end{bmatrix} \quad \textbf{(3)}$$

Additionally, to get better clustering, we use the RTT matrix:

$$\boldsymbol{G_L} = \begin{bmatrix} \boldsymbol{d_{11}} & \boldsymbol{d_{12}} & \cdots & \boldsymbol{d_{1N_L}} \\ \boldsymbol{d_{21}} & \boldsymbol{d_{22}} & \cdots & \boldsymbol{d_{2N_L}} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{d_{N_L 1}} & \boldsymbol{d_{N_L}2} & \cdots & \boldsymbol{d_{N_L N_L}} \end{bmatrix} \quad \textbf{(4)}$$

1) In the "User_BC" technique:

landmarks are clustered by distance matrix GL, which shows us user location. Though the matrix makes it easy to identify landmark distances and locate landmarks, as the distances must be interpreted to establish resemblance among landmarks, the algorithm of hierarchical clustering [12] is used to assemble GN landmarks in some way into their respective groups. Clustering of Hierarchical methods attempts to construct clusters as a tree-like structure. Algorithm 1 shows our clustering algorithm in more detail.

Algorithm 1: Hierarchal Cluster (GL, GN)

```
Set distance matrix GL, set cluster number GN as input.
The GN clusters and their members in specific as output.
proximity matrix G = GL;
divide each landmark into its own cluster;
G = GL;
while G ≠ GN do
      Connect the two clusters that are closest together.
      the distance between the clusters d_{ci,cj} = min d_{ij}
end
```

2) Web Service-based Clustering (Service BC):

Here, we rely on the NL-by-W QoS matrix QL to cluster the landmarks. A feature is a characteristic found in each row. Obviously, the feature is a rectangle that has homogenous properties. For determining the similarity of the two users, we utilize the Pearson Correlation Coefficient (PCC). The PCC may measure how two users are connected, and that can be shown as follows:

DOI: https://doi.org/10.54216/JISIoT.000101

$$s_{ij} = \frac{\sum_{m=1}^{W}(q_{im}-\bar{q}_i)(q_{jm}-\bar{q}_j)}{\left[\sum_{m=1}^{W}(q_{im}-\bar{q}_i)^2\sum_{m=1}^{W}(q_{jm}-\bar{q}_j)^2\right]^{1/2}} \quad (5)$$

For $\bar{q}_i$ denotes the average value of QoS q_{im} and sij refers to the result of similarity. The PCC ranges from -1 to 1, with 1 meaning the most resemblance and -1 meaning the most and so our formula is:

$$d'_{ij} = 1 - s_{ij}$$

it is known as the distance between major landmarks on land Thus, we can use the hierarchical clustering approach to extract the distance matrix NL and cluster the landmarks (GL, GN).

4.2 Prediction of QoS

With this information, we will develop forecasts on QoS levels for people who are trying to determine which Web service to use. Due to the constantly changing Internet environment, we do not rely on historical QoS data in our QoS prediction system. Nonetheless, each user calculates the time to landmark locations and, thus, they are clustered into one of the GN clusters, where QoS information about landmarks inside that cluster is gathered. The following will describe the NU -by-NL latency matrix for the number of service users being NU :

$$D_{UL} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1N_L} \\ d_{21} & d_{22} & \cdots & d_{2N_L} \\ \vdots & \vdots & \cdots & \vdots \\ d_{N_U1} & d_{N_U}2 & \cdots & d_{N_UN_L} \end{bmatrix} \quad (6)$$

It is necessary to know the distance between each user and each cluster to allocate service users to certain clusters. This can be done using hierarchical algorithms, which provide the minimum distance to the landmarks in each cluster. Each client can be put into a single cluster. The landmarks that are in the same group or cluster mainly affect the user's QoS prediction, where the service cluster is cluster u, L is the well-known site in Cluster u, known as u and l. The tiny distance d_{ul} signifies a high level of similarity between u and l, so that u and l can be said to be equal and so have $s_{ul} = 1/d_{ul}$. With the equation below, the values of QoS of user u, who is unknown, can be predicted:

$$p_{uw} = \frac{\sum_{l\in \text{Cluster } u} s_{ul}q_{lw}}{\sum_{l\in Cluster_u} s_{ul}}, w = 1, 2, \cdots, W \quad (7)$$

The forecast result, p_{uw}, shows how many QoS values will be saved (p_{uw}) when the Web service w is invoked by the landmark l. To boost the quality of their service level prediction, this QoS prediction system improves the accuracy by using a database of Internet landmarks and tracking web services over time. According to recent data about the quality of service (QoS), a new QoS prediction technique would likely lead to improvements in prediction accuracy when recommending web services, as well as being able to rapidly update QoS prediction results.

5. Experiments:

We put this area to use by conducting an experiment and a performance comparison between our proposed clustering based QoS prediction approaches and similar existing prediction methods, with the goal of understanding the advantages and disadvantages of both.

5.1 collection of data:

To access a database of research findings, we can utilize the PlanetLab system, which provides several systems, including data storage, map-making tools, and peer-to-peer programs. Today, PlanetLab, which has 1,093 nodes, is dispersed among 530 sites around the globe. Most of the nodes are in North America and Europe. The initial step in our plan was to create a list of the active nodes of PlanetLab, approximately 578 in number, which excluded inactive nodes. These inactive nodes may have simply gone offline or become disconnected from the Internet. If the performance of various nodes is comparable and they are all in the

DOI: https://doi.org/10.54216/JISIoT.000101

same lab, then the QoS prediction accuracy will rise but at the expense of realistic conditions, therefore we eliminate the duplicates to acquire a list of 288 nodes. In addition, around 10,000 URLs from the WSDream release dataset came from information from the Internet regarding Web services, gathered by using internet crawlers. On the other hand, ICMP ping does not work in all web services. For our experiment, we check and save over 2,000 web services. We transmit ICMP ping packets every 10 seconds, which carry 32 bytes of data. Then, we gather all replies with RTT as our response time. We ping ICMP to gauge a response time mostly because all web services we test are identical function-equivalent candidates, so their service-running times are the same. We are just discussing latency time prediction in this study, and we do so exclusively with regard to a dynamic Internet environment. Two data matrices were obtained after gathering information from 288 PlanetLab nodes and discarding failure results. one matrix has RTT times for each of the 200 PlanetLab nodes from any other 200 PlanetLab nodes, and the other matrix contains RTT times for each of the 1,597 web services from any of the 200 PlanetLab nodes.

5.2 metrics of evaluation:

To measure the QoS prediction performance, we use various metrics such as Root Mean Square Error RMSE [3] and Mean Absolute Error MAE [6], and Median Relative Error MRE [4] in our tests. One popular measure of how accurate predictions are is called mean absolute error (MAE), and is described as follows:

$$MAE = \frac{\sum_{i,j} |p_{ij} - q_{ij}|}{N} \tag{8}$$

Using q_{ij} for measured QoS value and p_{ij} for anticipated QoS value, a measure of the variability of the prediction error is mainly given by Root Mean Square Error (RMSE) which is given by:

$$RMSE = \sqrt{\frac{\sum_{i,j} (p_{ij} - q_{ij})^2}{N}} \tag{9}$$

It is also in use to assess the negative effect of Internet latency, as a way to uncover errors in predictions. The MRE value is calculated to acquire the average accuracy among all the values.

$$MRE = \text{Median}\left(RE_{ij}\right) = \text{Median}\left(\frac{|p_{ij} - q_{ij}|}{q_{ij}}\right) \tag{10}$$

Therefore, 50% of the overall prediction errors are smaller than MRE.

Table 1: Comparison of performance

Metrics	RMSE	MRE	MAE
UBC	**21.252**	**0.073**	**6.502**
IPCC	24.313	0.097	8.101
WSBC	22.148	0.091	7.252
UPCC	23.978	0.211	9.210
WSRec	22.368	0.094	8.103

5.3 Comparison of performance

To evaluate how well our approaches, predict customer behavior, we examine them by evaluating them against the WSRec algorithm (described in [6]), which is the well-known hybrid recommendation algorithm. WSRec is built around hybrid recommendation approaches that merge user-based and item-based recommendation algorithms. The selection of landmarks for our experiment is explained in a subsequent Section. In addition, NC is set to 50, which represents the number of clusters. Table I displays the expected prediction accuracy for several strategies, as found in [6]. In this table, UPCC, IPCC, and WSRec refer to three collaborative filtering methods suggested there. User-based CF method (UPCC) and item-based CF method (IPCC) are codes for the user-based and item-based CF methods, respectively. UPCC, WSRec , IPCC, and findings are acquired with 49% density of matrix. We can see from the findings in the table that the User BC and Web Service BC approaches both have decreased MAE, RMSE, and MRE values. This

DOI: https://doi.org/10.54216/JISIoT.000101

means that they do better than the collaborative filtering methods, and User BC is doing much better. The effectiveness of collaborative filtering-based methods relies entirely on having the right Top-K comparable neighbors; however, this is prone to error if dissimilar neighbors wind up in the Top-K. On top of that, we have no way of knowing if the predictions regarding QoS (which are built on top of previous QoS data) are currently available in real time. Since the Internet environment is always changing, the projections are not confident. Our approaches, on the other hand, are more accurate since the size of each cluster has a variable number of similar neighbors, which is modifiable for QoS prediction. User BC, which benefits from having access to more latency information, notably the "latency matrix GL," provides greater performance than Web Service_BC.

5.4 The impact of GN:

In clustering based QoS prediction methods, the importance of cluster number GN is crucial. To research the effect of GN, we experiment with different GN levels (10 to 100), and we give our results in Figure 2. We can see that GN affects prediction accuracy greatly. The MAE reaches its lowest point when GN = 50, and the MRE has its lowest point when GN = 90. In contrast, the optimum results are obtained for both MAE and MRE when GN = 92 for Web Service BC. basically, as GN increases, the prediction errors first reduce and eventually don't change much. Because when GN is little, the cluster is oversized. Even though certain landmarks exist in the same group as the user, they may not be close enough to the user to have any similarities. While we cannot identify adequate landmarks within one cluster of clients to assist in the accurate quality of service (QoS) forecasts, there are many more than expected. That means that the creation of a more effective GN will improve prediction accuracy greatly.

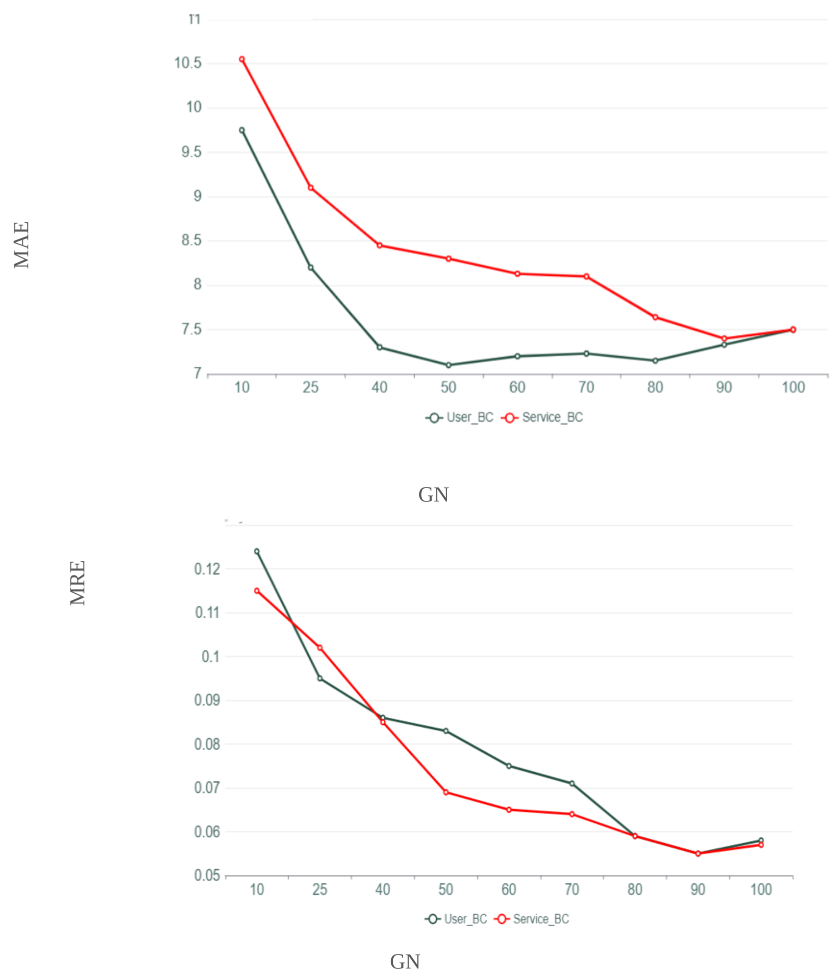

Figure 2: The impact of GN

DOI: https://doi.org/10.54216/JISIoT.000101

5.5 The impact of the selection of landmarks

The placement and number of landmarks in the real world are critical for referencing service users. An incorrect reference point might have a considerable negative impact on QoS prediction performance. While doing an assignment, it is essential to completely isolate any notable landmarks in order to learn more about the network structure. We present two selection approaches, random and clustering-based, to analyze the influence of selected landmarks. We randomly select landmarks by sampling them all evenly, meaning that every node has an equal chance of being chosen.

- **Clustering**: We cluster all candidate nodes into NL clusters in order to segregate them well and then select one reference number for each cluster. In this plan, all landmarks are spaced widely apart.

We use our data, which appears in Figure 3, to show the performance of our tests, which were conducted to compare the performance of different schemes. Based on the experiments, the method that utilizes clustering for landmark selection regularly outperforms the random method by a wide margin, which proves that landmarks with large separation can yield much more information. When landmarks grow, the MAE accuracy improves. Based on this finding, it appears that additional landmarks for the service customers' reference will improve the performance of the predictions.

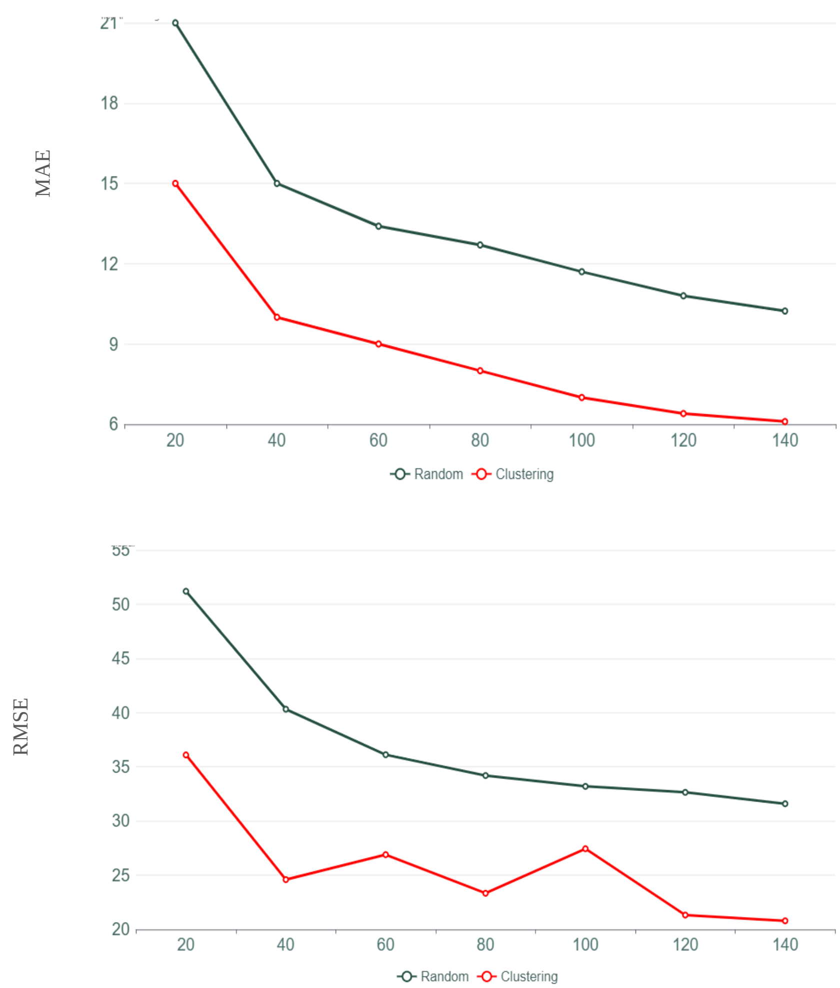

Figure 3: The impact of Landmarks selection

DOI: https://doi.org/10.54216/JISIoT.000101

6. Conclusion:

In this research, we have created a new QoS prediction framework based on landmarks, as well as a recommendation for cluster-based recommendations of Web services. An algorithm deployed on the Internet relies on a system of predefined landmarks to measure and update QoS information, allowing customers to know where their QoS levels are headed. We conducted experiments to prove that our methods are more successful at collaborative filtering and service recommendation than existing ones. This should boost confidence in the quality-of-service predictions.

References

[1] L. Zhang, J. Zhang, and H. Cai, Services Computing: Core Enabling Technology of the Modern Services Industry. Tsinghua University Press, 2007.

[2] Z. Zheng, “QoS management of web services,” Ph.D. dissertation, The Chinese University of Hong Kong, 2011.

[3] Z. Zheng and M. R. Lyu, “Collaborative reliability prediction of service-oriented systems,” in Proc. of the ACM ICSE’10, 2010, pp. 35–44.

[4] K. Karta, An Investigation on Personalized Collaborative Filtering for Web Service Selection, Honours Programme thesis, University of Western Australia, Brisbane, 2005.

[5] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, “Personalized QoS prediction for web services via collaborative filtering,” in Proc. of the IEEE ICWS’07, 2007, pp. 439 – 446.

[6] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “WSRec: A collaborative filtering based web service recommender system,” in Proc. of the IEEE ICWS’09, 2009, pp. 437–444.

[7] Q. Xie, K. Wu, J. Xu, P. He, and M. Chen, “Personalized context-aware QoS prediction for web services based on collaborative filtering,” in Proc. of the 6th Int’l Conference on Advanced Data Mining and Applications (ADMA’10), 2010, pp. 368–375.

[8] V. Cardellini, E. Casalicchio, V. Grassi, and F. L. Presti, “Flow-based service selection for web service composition supporting multiple QoS classes,” in Proc. of the IEEE ICWS’07, 2007, pp. 743–750.

[9] J. E. Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz, “QoS-driven selection of web services for transactional composition,” in Proc. of the IEEE ICWS’08, 2008, pp. 653–660.

DOI: https://doi.org/10.54216/JISIoT.000101

[10] Z. Zheng and M. R. Lyu, "A distributed replication strategy evaluation and selection framework for fault tolerant web services," in Proc. of the IEEE ICWS'08, 2008, pp. 145–152.

[11] X. Chen, X. Liu, Z. Huang, and H. Sun, "RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in Proc. of the IEEE ICWS'10, 2010, pp. 9–16. [12] Y. Jiang, J. Liu, M. Tang, and X. F. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in Proc. of the IEEE ICWS'11, 2011.

[13] V. K. Pang-Ning Tan, Michael Steinbach, Introduction to Data Mining. First Edition. Addison Wesley Longman Publishing Co. Inc., 2005

DOI: https://doi.org/10.54216/JISIoT.000101