



Research article

SaaSRec+: a new context-aware recommendation method for SaaS services

Hossein Habibi¹, Abbas Rasoolzadegan^{1,*}, Amir Mashmool², Shahab S. Band^{3,*}, Anthony Theodore Chronopoulos^{4,5} and Amir Mosavi^{6,7}

¹ Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

² Department of Computer Engineering, University of Birjand, Birjand, Iran

³ Future Technology Research Center, College of Future, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan

⁴ Department of Computer Science, University of Texas, San Antonio TX 78249, USA

⁵ Department of Computer Engineering & Informatics, University of Patras, Rio 26500, Greece

⁶ Faculty of Civil Engineering, Technische Universität Dresden, Dresden 01069, Germany

⁷ Institute of Software Design and Development, Obuda University, Budapest 1034, Hungary

*** Correspondence:** Email: rasoolzadegan@um.ac.ir, shamshirbands@yuntech.edu.tw.

Abstract: Cloud computing is an attractive model that provides users with a variety of services. Thus, the number of cloud services on the market is growing rapidly. Therefore, choosing the proper cloud service is an important challenge. Another major challenge is the availability of diverse cloud services with similar performance, which makes it difficult for users to choose the cloud service that suits their needs. Therefore, the existing service selection approaches is not able to solve the problem, and cloud service recommendation has become an essential and important need. In this paper, we present a new way for context-aware cloud service recommendation. Our proposed method seeks to solve the weakness in user clustering, which itself is due to reasons such as 1) lack of full use of contextual information such as cloud service placement, and 2) inaccurate method of determining the similarity of two vectors. The evaluation conducted by the WSDream dataset indicates a reduction in the cloud service recommendation process error rate. The volume of data used in the evaluation of this paper is 5 times that of the basic method. Also, according to the T-test, the service recommendation performance in the proposed method is significant.

Keywords: cloud service recommendation; context-aware service recommendation; cloud service composition; cloud service selection, personalized recommendation; spatial effects; hybrid filtering;

content-based filtering; collaborative filtering; QoS

1. Introduction

The use of cloud service is increasingly expanding. Different models of cloud service include software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), and anything as a service (XaaS) [1,2]. The increase in the number and variety of cloud services is due to reasons such as the increase in the number of cloud service providers and the migration of web services to the cloud platform. Due to the increasing number and variety of cloud services, there are several ways to recommend cloud service to help users find service that suits their preferences [3–8]. However, research on cloud service recommendations has several issues to face such as limitations and shortcomings, some of the most important of which are:

- 1) Lack of sufficient attention to the contextual information.
- 2) Using inaccurate methods to determine the similarity in cloud service recommendations.

The first and second limitations increase the error of the recommendation method. The proposed method in the present study seeks to improve the cloud service recommendation process by removing the first and second limitations. We call this method SaaSRec+. The position of the proposed method, following the standard provided by the NIST organization, is in the cloud computing reference architecture mediator section [9]. More specifically, in this article, contextual information means information about the user's location (user's IP) and the location of the deployed cloud service (service's IP).

In the proposed solution, a list of cloud services is first received. This list is then modified after collaborative and content-based filtering. Then, using a ranker, the services in this list are sorted. After using the offered service, the user's feedback is recorded in his profile. In the end, the solution proposed for the users' clustering is updated using a similarity-determining formula based on their profiles. Each user's profile contains information such as contextual information.

The main contributions of the manuscript can be summarized as follows:

- 1) Using contextual information of users and cloud services to more accurately cluster users by applying the $P(u,s)$ formula.
- 2) Applying a formula (RA) for precise determination of similarity between two vectors to more accurately cluster users.

To evaluate the method presented in this paper, the proposed method is tested with the most comprehensive method of cloud service recommendation [5] using the WSDream common dataset. The volume of data used in the evaluation of this paper is 5 times that of the basic method. The results of this test and the analysis are presented in the evaluation section. Based on the results obtained, the values related to the mean absolute error (MAE) criterion for both methods are calculated. Following the above criteria and T-test, the proposed method offers a significant improvement compared to the base method.

The focus of this article is to provide a context-aware cloud service recommendation method. According to Section 2 below, cloud service recommendation methods using the user-service matrix (which is the same as the history user) seek to recommend cloud services that the users are satisfied with. The user-service matrix is a two-dimensional numeric matrix in which each row represents a user and each column represents a service. The value of each cell in this matrix also indicates the

level of user satisfaction with a particular service. Cloud service recommendation methods, with a user-service matrix, try to estimate the value of each cell of this matrix for which no value has been recorded so far (which means that the specific user does not provide any feedback on that particular service). The higher the values of the service user matrix cells, the more satisfied the particular user is with the particular service. Often, the range of values of this matrix is from 1 to 5. In this paper, each of the WSDream-RT and WSDream-TP datasets are used as the user-service matrix, and in accordance with Subsection 4.2 below, in order to evaluate the proposed method, their entries are mapped in the range of 1 to 5. More details are provided below (Subsection 4.2).

The organizational structure of our paper is summarized as follows. We review related works in Section 2. Section 3 presents the proposed recommendation method SaaSRec+ that is based on the context-aware recommendation of the cloud services technique. In the Section 4, the proposed solution is evaluated, the evaluation results are presented and analyzed. Results and discussions are presented in Section 5. At last, in Section 6, we conclude the whole paper and point out the prospective improvement directions in future works.

2. Related works

Several studies have been conducted on web service recommendation methods. Nevertheless, minimum attention has been paid to the cloud service. Fang et al. presented a cloud service model [4]. By explicitly acknowledging the qualitative characteristics desired by the user and the importance of each of them as the user's preferences, their proposed model organizes and recommends the selected cloud service. In this study, an inter-data weighting summary that shows the user's preferences has been used to obtain the rank of each service. Their focus has been on creating and improving their fuzzy cloud ontology with the help of their users. The study did not use collaborative and content-based filtering techniques.

Afify et al. [5] proposed a method of cloud service recommendation in 2016. The presented recommendation method receives a list of selected services and collaborative filtering and it is performed first. Then, depending on the user's cluster, the degree of a reputation of the selected service is obtained from the users of that cluster. Then, based on the obtained reputation values, each of the selected services receives a score. User clustering is done using the HAC clustering method [10]. This method uses similarities between users to perform clustering. In fact, the HAC method itself uses the results of similarity comparison formulas between users. However, the two-vector comparison formulas (equivalent to comparing the similarity of users) in the cloud service recommendation methods are formulas such as PCC [12], Cosine [14]. In fact, any of the PCC [12] and Cosine [14] formulas can be used in the HAC method [10]. Each user's preferences are determined by himself/herself and by inserting a set of keywords. A user's numerical feedback is the score that he attributes to a particular service, thus showing his satisfaction with the use of that service. In the next step, content-based filtering is performed, and in the final step, the proposed services are ranked.

Jiang et al. [6] proposed a method of cloud service recommendation. This method consists of 2 phases. In the first phase, the textual description of the cloud service is first compiled by a crawler from the web. Processes such as deleting stop words are then performed on the collected texts. At the end of the first phase, after processing these texts, cloud services are clustered. In the second phase, based on the altered PageRank algorithm, and the cloud service label, the services in each cluster are

ranked. In this phase, a list of ranked services is obtained. Recently, Fan et al. provided a context-aware recommendation method on web service [11]. In this research, two types of contextual information, namely geographical location and time, have been considered. The method presented in this study has three steps. The first two steps are service list filtering, and the last step is filtered service quality estimation. In the first and second steps, geographic and time filtering are performed, respectively, and in the last step, using the business model, web service recommendation is done. The formula used in time filtering is based on PCC [12].

In a study conducted by Wu et al., a new formula was proposed to calculate the unique similarity between two vectors [13]. This paper presents the RA formula. They showed that Formula RA was more accurate than PCC [12], Cosine [14], and NR [15] formulas. This formula was applied to recommend web services. However, in the field of cloud service recommendation, this formula was not applied, and incidentally, this is one of the shortcomings of cloud service recommendation methods.

In another study conducted by Mezni et al., a cloud service recommendation method based on collaborative filtering was suggested [16]. Fuzzy Formal Concept Analysis has been used in this method. This mathematical analysis is based on the lattice theory. In this method, three lattices were created. These three lattices were used to determine similar users, similar services, and appropriate services for users. Using a mathematical model to accurately identify similar services and users is one of the advantages of this method. This method can also be used for different types of cloud services. In the proposed method, a threshold value is used to monitor services in the construction of a service lattice. Determining the exact value is effective in the time required to implement this method and the quality of the results. However, this method does not specify how to determine this threshold. Another disadvantage of this method is the lack of attention to the geographical location of users and cloud services in the service recommendation. Another disadvantage of this method is the lack of reference to how to calculate the similarity between the two users. There is also no mention of how to determine the threshold value used to create users' lattice. Also, as mentioned in this paper, the computational cost of this method is high.

In a study conducted by Wei et al., a method is introduced for privacy-aware recommendation method based on Boolean historical invocation records [17]. Making changes to the Simhash technique, authors introduced the Multi-probe Simhash (MPS) technique and named it RecMPS. Considering the possibility of distributing sensitive users' data on multiple cloud platforms, it is important to protect them. They compared RecMPS in terms of accuracy with the three methods DistSRLSH [18], WSRec [19], and ICF. Their results showed that RecMPS was better in this respect than the other three methods. One of the advantages of this method is the reduction of time required for cloud services recommendation, using the Simhash technique. Another advantage of this method is paying attention to users' data protection. One of the disadvantages of this method is the lack of attention to factors other than Boolean service invocation records in the process of recommendation. This method also relies on the Simhash technique. However, because this technique is probabilistic, this method is not always able to successfully recommend services.

In Zhang et al. [20], a cloud service recommendation approach namely Personalized Cloud Service RECommendation (P-CSREC) is presented and it works based on the characterization of heterogeneous information networks. Initially, a similarity measure is computed by including the interest areas of the users and a model is developed. The K-harmonic means algorithm is applied to deal with the convergence and the frequent pattern growth (FP-Growth) association rules are then

formed for framing the personalized recommendation for users. This work claims that the utilization of clustering algorithm and FP-Growth algorithm proves better accuracy rates.

Chai et al. [21] modeled the recommendation system as a multi-objective optimization problem, and aims to find tradeoff solutions between accuracy and diversity. There are two conflicting goals for the recommendation system. Authors firstly used singular value decomposition to get the recommendation list, then a multi-objective immune algorithm is used to optimize it. The use of the Cosine formula [14] for comparison between users is one of the demerits of their study. Nevertheless, Wu et al. provided a more accurate comparison formula called RA [13]. Guo et al. [22] presented a personalized QoS prediction for service recommendation with a service-oriented tensor model. They proposed a QoS value prediction model by extending the two-dimensional user service matrix into a three-dimensional user-service-service tensor model, called service-oriented tensor (SOT). The prediction approach incorporates service collaboration from other similar services and relevant users by using a three-order tensor.

Table 1. Comparison of works in the field of service recommendation.

Comparison criteria	Collaborative filtering	Content-based filtering	Using web feedbacks	Time aware	Service location-aware	User location-aware	Cold start problem aware	Feedback filtering aware
Related works								
Cloud Services	[4]	Yes	No	No	No	No	No	No
	[5]	Yes	Yes	No	No	Yes	Yes	Yes
	[6]	Yes	No	No	No	No	No	No
	[16]	Yes	No	No	No	No	Yes	No
	[17]	Yes	No	No	No	No	Yes	No
	[20]	Yes	Yes	No	No	No	No	No
	[23]	Yes	No	No	No	No	No	No
	[24]	Yes	No	Yes	No	No	No	Yes
	[25]	No	Yes	No	No	No	No	No
Web Services	[11]	Yes	No	No	Yes	Yes	No	No
	[13]	Yes	No	No	No	Yes	Yes	No
	[21]	Yes	Yes	No	No	No	No	No
	[22]	Yes	Yes	No	No	No	Yes	No

A cloud service recommendation system is presented based on clustering trust measures in a multi-cloud environment in Priya et al. [23]. Finding trustworthy service is a problem. Due to this problem, end-users need trust-based appropriate service with minimum time consumption, and the service should be reliable too. This work attempts to keep this as the goal and presents a cloud service recommendation system using a clustering-based trust degree computation algorithm. The merit of this study is its attention to the following three factors at the same time: 1) The amount of trust in the service 2) The time needed to recommend the service 3) The accuracy of the recommendation method. Lack of attention to the underlying factors is one of the weaknesses of this study in recommending cloud services.

Indira et al. [24] proposed the clustering recommendation system executed in a cloud

environment. Irrelevant features are present in data and in this proposed scheme, an effective feature selection approach named modified LDA has been utilized for acquiring the relevant information only. DBSCAN (density-based spatial clustering of applications with noise termed) is utilized as a clustering approach that provides better quality in terms of segregating the number of movies. Attention to the performance and accuracy of its method is one of the advantages of this study. However, its weaknesses consist of the lack of attention to cloud services and the method restriction in recommending various videos based on user feedback. Besides, the underlying factors have not been paid attention.

A broker-based framework for the recommendation of cloud services is presented in Hentschel et al. [25]. This paper proposed a framework that would identify, rank, and recommend cloud services from multiple components to individual users. They have iteratively developed the first version of their virtual broker as a service framework, called Vibros, as part of their design-oriented approach.

Then, as indicated in Table 1, nine valid and novel cloud service recommendation methods and four examples of web services recommendation methods are compared. The titles of the fourth, fifth, and sixth columns are the underlying factors concerned in the present article. For the methods in Table 1 that have been considered each of these three factors is concerned with context-aware methods. That is, references [5,11,13] are context-aware. None of the cloud service recommendation techniques have paid attention to the location of cloud services and the time. Besides, most methods have employed at least collaborative filtering in their recommendation method.

In accordance with Table 1, the study carried out by Afify et al. [5] is one of the most comprehensive, reliable, and novel cloud service recommendation methods. However, it can be easily understood that this method has also overlooked some essential points in its procedure. These points include 1) The use of web user feedback, 2) Consideration of time in the expression of quality, and 3) Paying attention to the place of service. Taking into consideration the listed features is effective in increasing the precision and recall and reducing the mean absolute error of the proposed method.

According to the work reviewed in Section 2, reference [11] introduces the formula $P(u,s)$, and reference [13] provides a more accurate formula than the PCC [12] and Cosine [14] formulas. These studies, however, were conducted separately to recommend web services. Whereas before, cloud service recommendation methods were less accurate due to the lack of proper use of user and services data and the application of the above formulas. By integrating these formulas with the basic method and the implementations described in subsection 4.2, we present the results of applying these formulas and show that the basic method can be improved.

3. Proposed method

In this section, a context-aware cloud service recommendation is introduced. We call this method SaaSRec+. SaaSRec+ is inspired by the SaaSRec [5] method and aims to increase accuracy and reduce the error rate of its recommendations. Figure 1 shows an abstract overview of the proposed method. It should be noted that according to the exact wording of the basic method article [5], in the current article, topics such as: how to extract cloud service metadata, how to store cloud service metadata and user explicit preferences/profile, how to use them in comparisons, how to describe cloud services, local and global reputation calculation formula will not be addressed in detail. These issues are discussed in detail in the basic method article [5]. Also, for same reason, in

detail explanation related to $P(u,s)$ [11] and RA [13] formula will not be addressed.

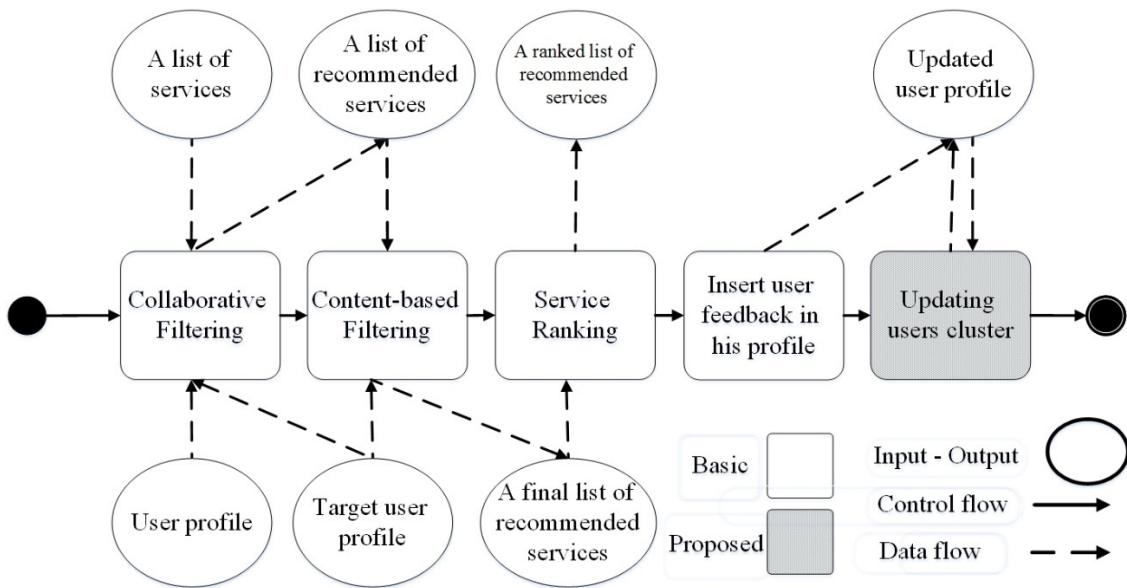


Figure 1. Activity diagram of the context-aware cloud service recommendation method.

In the proposed method, a list of cloud services is first received. This list is then modified using collaborative and content-based filtering. In the following, the services in this list are ranked and provided to the user. During the collaborative filtering process, two parameters about 1) local reputation and 2) global reputation for each of the services in the list of services received by the user are calculated according to which cluster of users the user is in. The ranker uses the calculated values for these parameters.

During the content-based filtering step, according to the user's history, the compatibility of the services in the list with his profile is calculated. User explicit preferences and user history are stored in the user profile. User explicit preferences refers to a set of keywords (for example: security and availability) that the users write explicitly in their profile to express their expectations. A user's history is the user's scores given to the services used. Each score is a number between 1 and 5. The highest score indicates the highest level of user satisfaction with that service. Two comparisons are made to determine the compatibility of services with the user's profile.

The first comparison is made between the description of each of the services in the list and the description of each of the services scored by the user. Description of cloud services is done with the help of text description provided by cloud service providers. The first comparison is done by comparing two descriptive vectors. For each service on the list, the highest similarity between that service and the scored service is considered as the historical similarity parameter. This parameter is calculated for all services on the list. The second comparison is made between the metadata of each of the services in the list and the explicit preferences of the user; a certain number is obtained for each service. The sum of these two numbers is obtained as a number that indicates the degree of compatibility of the service in the list with the user's profile. Metadata are keywords, describe cloud services, extracted from text service description provided by service providers. Keywords such as security, scalability and response time are three examples of metadata.

Finally, having the values of several parameters and calculating a weighted sum between these values, the ranker determines the rank of each service. These parameters are 1) the cost of the service, 2) the degree of compatibility of the service with the user's profile, 3) the level of the global reputation of the service and 4) the level of local reputation of the service. The amount of weight that will be given to each of these numbers is also received from the user. After calculating this weighted sum, the number that expresses the rank of the service is obtained. The output of the proposed recommendation method is generated in the current paper after the performance of the ranking activity. This output includes a downward ordered list of services. Finally, after performing these activities, the user selects one of these organized services. After the user uses the selected service, his feedback will be recorded in his profile; it means that the selected service, the user's score for that service, the location of the selected service, and the user's location will be registered in his profile. This data will be used to update the user's clustering.

To perform collaborative and content-based filtering, the cluster in which the user is a member must be identified. In the present paper, various data are proposed to determine the similarity between users in the clustering process. These data include: 1) the service used, 2) the score assigned to the service by the user, 3) the location of the service, 4) the user's location, and 5) the preferences explicitly expressed by the user as keywords. Of these data, case 3 did not appear in the basic study [5] and case 4 was not fully used. Each user is considered as a vector with the above 5 data.

The process of filtering users' information and clustering them is done according to the contextual information. This process is shown in Figure 2 below. In the first step of this process, a set of services that the user prefers is obtained by receiving the user's IP and service's IP using the $P(u,s)$ formula [11]. In this formula, the appropriateness of services and users is examined in terms of their geographical location. In the next step, from the data registered in the user-service matrix, only the data related to the user's preferred services are extracted. In this way, the user-service matrix is filtered. In the last step, the users are clustered using the HAC [10] method and RA [13] formula.

Table 2 is presented to complete Figure 2, which shows the process of filtering and clustering users using contextual information.

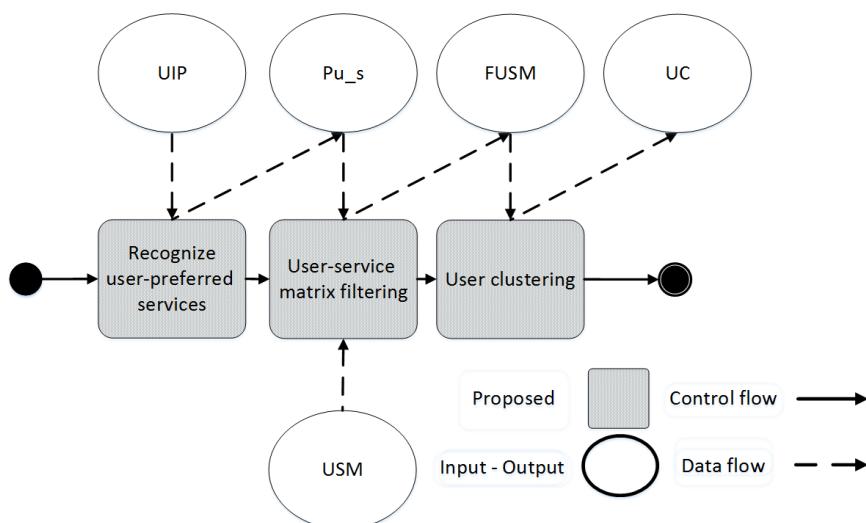


Figure 2. Process of users' context-aware clustering.

Table 2. Abbreviations used in Figure 2 and their meanings.

Abbreviation	Explanation
UIP	User location IP
Pu_s	User-preferred services
USM	User-service matrix
FUSM	Filtered user-service matrix
UC	User's cluster

The $P(u,s)$ [11] formula to obtain the invocation records of cloud service that correspond to the user preference is:

$$P(u, s) = w_1 P_{u,RCS} + w_2 P_{u,NDS} \quad (1)$$

here, $P_{u,RCS}$ is the impact of regional correlation on user preference and $P_{u,NDS}$ is the impact of network distance on user preference. Also, w_1 and w_2 are different weights to the effect of both regional correlation and network distance. In this formula u represents user and s represents service.

The RA [13] formula to measure the similarity between user's u and v is:

$$\text{Sim}(u, v) = \frac{\sum_{i \in I} \frac{\min(r_{u,i}, r_{v,i})}{\max(r_{u,i}, r_{v,i})}}{|I|} \quad (2)$$

here, $r_{u,i}$ and $r_{v,i}$ are the values when users u and v invoke item i ; $\min(r_{u,i}, r_{v,i})$ is the minimum of $r_{u,i}$ and $r_{v,i}$; $\max(r_{u,i}, r_{v,i})$ is the maximum of them; I is the set of items that are invoked by both users u and v , and $|I|$ is the number of items in I .

In the HAC method [10], each user is first considered as an independent cluster. Then, based on the proximity of the centers, these clusters are merged. This process will continue until the cluster size reaches a number k . Also, since each user is considered as a vector, the RA formula [13] is used to determine the similarity between users.

4. Validation

In this section, the evaluation metric is first stated. The results of the proposed evaluation method and discussion are then described. Then, to evaluate the improvement of the proposed method, a T-statistical test is used, and the relevant results and analysis and finally the performance analysis of the proposed method are presented.

4.1. Evaluation metric

The mean absolute error (MAE) criterion will be used as the criterion for evaluating the proposed recommendation process. This criterion is one of the most important and widely used criteria for evaluating the results of cloud service recommendation methods invalid research. This criterion can be calculated according to Eq (3). Where the $\text{calculated}_{u,s}$ indicates the calculated value of the intended score of user u to service s . $R_{u,s}$ represents the actual value of the score provided by the user u to service s . P also represents the total number of calculations.

$$MAE = \frac{\sum_{u,s} |calculated_{u,s} - R_{u,s}|}{P} \quad (3)$$

4.2. Results and discussion

In the current study, the valid WSDream dataset [26] is used to evaluate the proposed method. Since valid researches such as the introduced method named SaaSRec in the base study [5] has used this dataset for their evaluation, this dataset has also been used in the current study. This dataset contains two sets of data. Each of these datasets is a two-dimensional matrix with 339 rows and 5825 columns. Each row represents a user and each column represents a service. One of the above-said datasets indicates the amount of service response time for each user and the other dataset shows the rate of throughput for each user. Also, as no values were used for user explicit preferences, for evaluating our proposed method, user profile is equal to user history.

The user-service matrix is used to evaluate cloud service recommendation methods. The numbers in the entries of the above-mentioned matrix are in the range of 1 to 5. Given that the WSDream-RT and WSDream-TP datasets are used to evaluate the proposed method, and that the numbers in this dataset are not in the range of 1 to 5, these numbers were mapped in the range of 1 to 5. The mapping function is also implemented according to the mapping function introduced and used in the basic study [5].

In order to fairly evaluate the proposed method and compare it with the base method, and considering that the base study [5] used a maximum of 9% of the volume of WSDream-RT and WSDream-TP datasets, in the current study, with the help of a of Java code, part of the dataset has been deleted. Finally, a matrix containing 339 rows and 525 columns was obtained.

After reducing the volume of the user-service matrix to 9% of its original volume and mapping the numbers in it in the range of 1 to 5, the above-said matrix has been used as the common input of the proposed method and the base method.

In the proposed method, the user-service matrix is filtered according to the process shown in Figure 2; but the user-service matrix is not filtered in the base method.

In the following, users are clustered in the proposed method and the basic method. Based on user clustering and according to the process described in Figure 1, 203 entries are calculated from the user-service matrix. Finally, for each of the stated entries, the difference between the calculated value and the actual value of the entry is calculated as the error rate. According to Figure 2, the user's IP is important in filtering the user-service matrix. Therefore, the dataset should be used to evaluate and compare the proposed method and the basic method that contains the user's IP information. The WSDream dataset provides users' and services' IPs.

According to Figure 2, which shows the process of users' context-aware clustering, the codes in the GitHub source are have been used to cluster users in the recommendation section and to implement the HAC method [10]. Of course, the final implementation has been achieved by making the necessary changes to the initial implementation available on the GitHub site. Also, according to Section 3 above , the parameter k was introduced to determine the maximum size of users' clusters; k is used in HAC method [10]. In order to make a fair evaluation and compare the proposed method and the basic method. Considering that the basic method has set $k = 40$, we also set the value of this parameter is 40.

As stated in Section 3 above, the $P(u, s)$ formula [11] has been used to identify user-preferred services shown in step 1 of Figure 2. Therefore, to implement this formula, the GeoLite dataset and Geo IP library have been used. Using the dataset and library, the geographical distance between users and services in the WSDream-RT dataset and the WSDream-TP dataset is calculated.

Afify et al. [5] have evaluated their work with a maximum of 40 services. It has estimated a maximum of 40 entries of each of the WSDream-RT and WSDream-TP matrices. However, here 203 entries of these matrices are estimated. Details of the results for calculating the MAE of each of the above methods are provided in the Appendix.

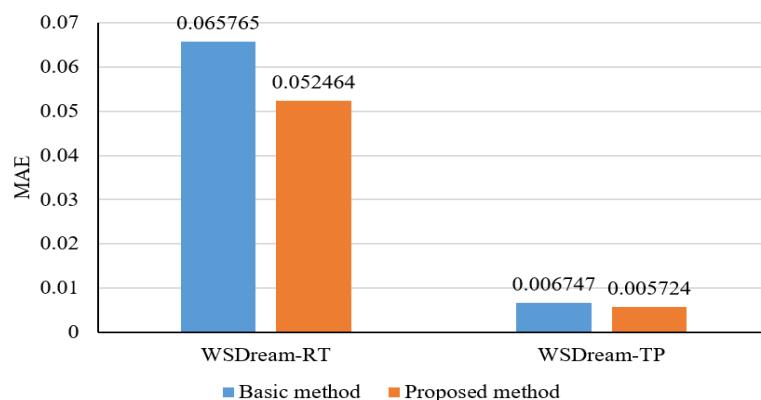


Figure 3. Comparison of the MAE values of the implementation of the basic method and the proposed method on the WSDream dataset.

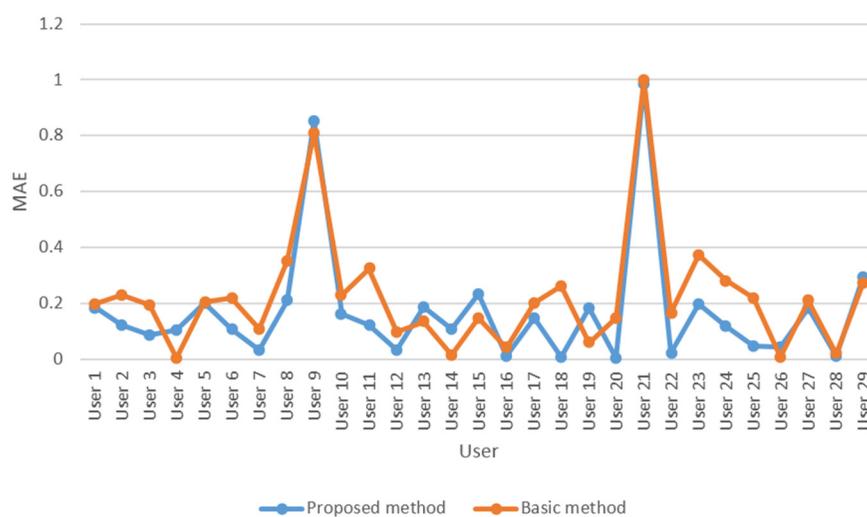


Figure 4. Comparison of the error rate of the basic method and the proposed method for service 1 in the WSDream-RT.

In Figure 3, the MAE of the basic method and the proposed method on the WSDream-RT and WSDream-TP datasets are compared. The results show that the proposed method has succeeded in improving the basic method. The observed improvement was achieved by applying the stated

innovations in the proposed solution. These innovations include more attention to contextual preferences and the use of an accurate similarity determination formula.

Figure 4 compares the error rate of the proposed method and the basic method in recommending a service to different users. Other examples of this comparison can be seen in Figures 5–11. These figures are based on Tables A1–A4.

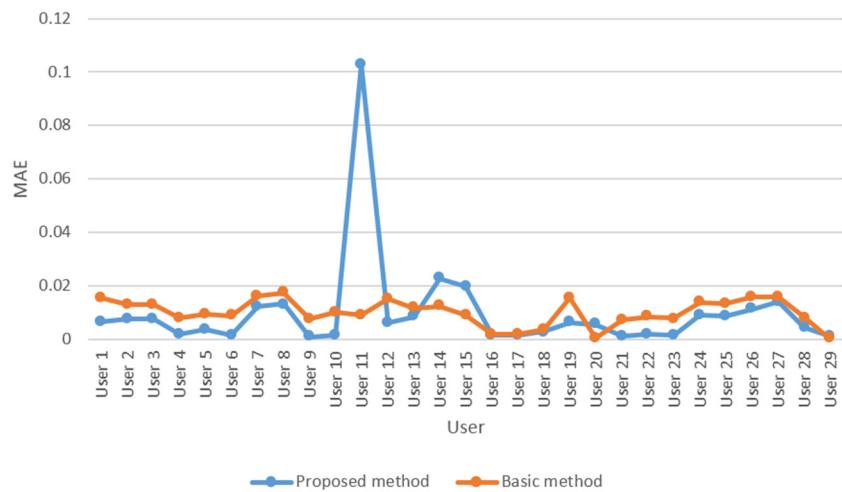


Figure 5. Comparison of the error rate of the basic method and the proposed method for service 2 in the WSDream-RT.

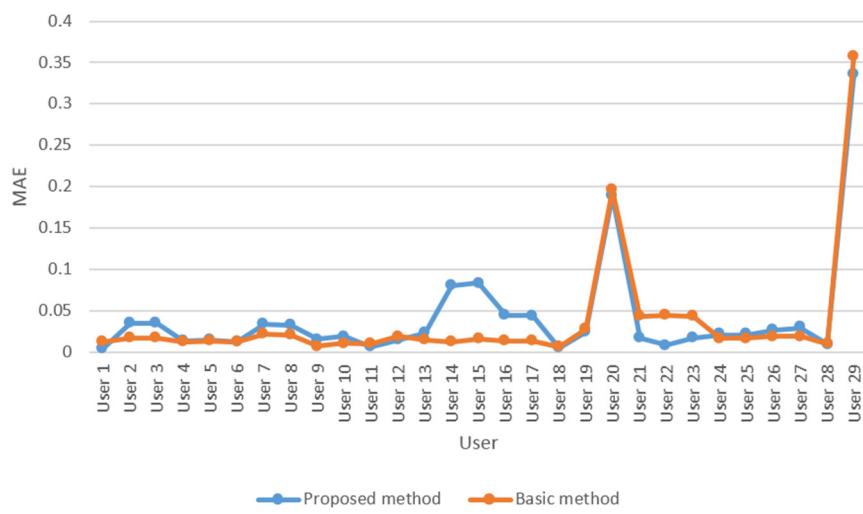


Figure 6. Comparison of the error rate of the basic method and the proposed method for service 3 in the WSDream-RT.

In Figures 4–11, the MAE values obtained from both the basic and proposed methods are reported on the WSDream-RT and WSDream-TP datasets. At first glance, some figures express similar behavior to two methods, such as Figures 4–6 and 10. However, in these figures, in some places, the two graphs are separated from each other. Also in other figures, in some cases, similar behavior is observed and in others, different behavior. For a more detailed study and to provide

analytical statistics comparing the behavior of the two methods, we obtained Tables A5 and A6. Table A5 is the absolute difference between the data in Tables A1 and A2, respectively, and Table A6 is the absolute value of the difference between the data in Tables A3 and A4. According to Table A5, 95% of the data in this table are in the range of 0 to less than 0.15, and only 5% of these values are in the range of 0.15 to a maximum of 0.25. According to Table A6, 97% of the data are in the range of 0 to less than 0.014. According to this statistic, in 5% of cases in WSDream-RT dataset, the two methods had more than 0.15 differences. In fact, in most cases, the two methods behave almost similarly in both the WSDream-RT and WSDream-TP datasets, but this does not mean that both methods always behave similarly.

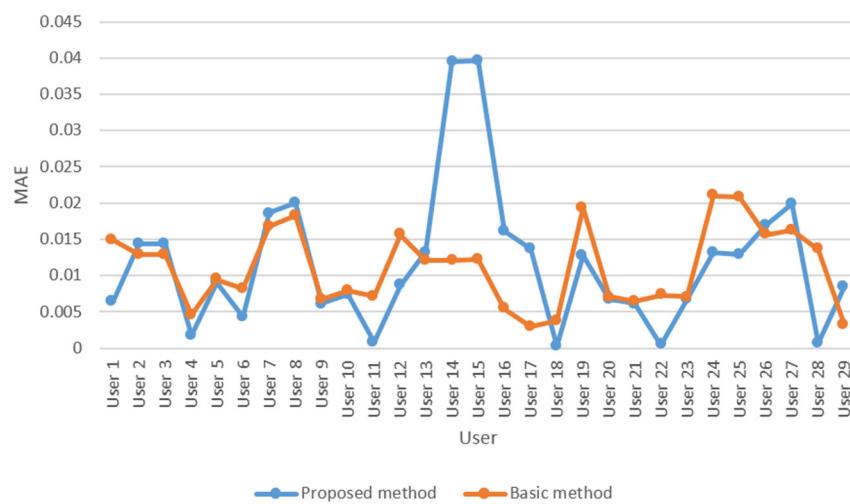


Figure 7. Comparison of the error rate of the basic method and the proposed method for service 4 in the WSDream-RT.

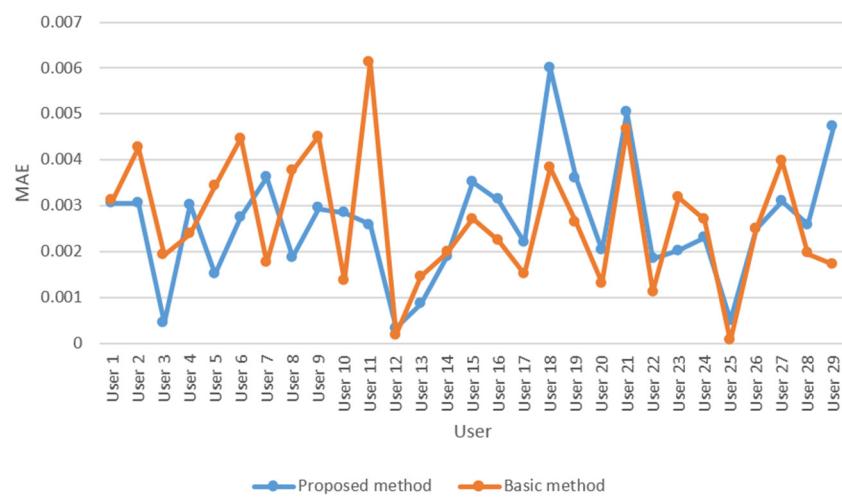


Figure 8. Comparison of the error rate of the basic method and the proposed method for service 1 in the WSDream-TP.

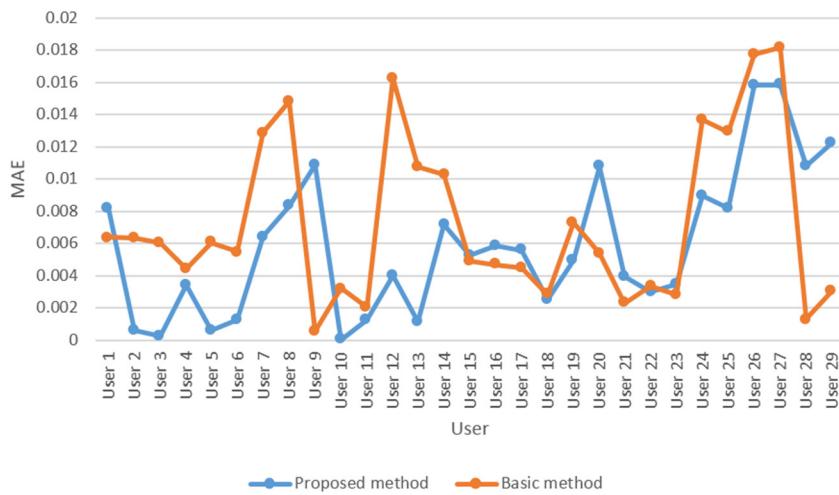


Figure 9. Comparison of the error rate of the basic method and the proposed method for service 2 in the WSDream-TP.

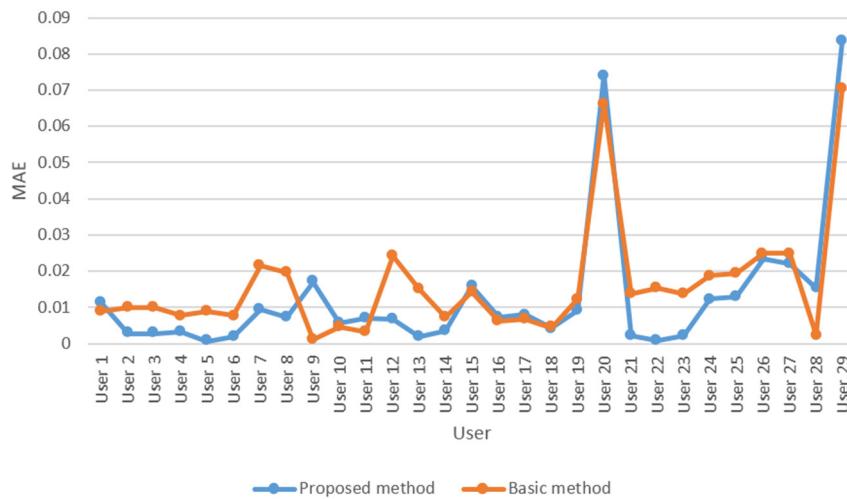


Figure 10. Comparison of the error rate of the basic method and the proposed method for service 3 in the WSDream-TP.

According to Figure 3 the proposed method is better than the basic method. In order to compare the two methods more accurately, we will provide more details on the amount of improvement obtained from the proposed method.

According to Figure 3, the difference in MAE values between the two methods on the WSDream-RT and WSDream-TP datasets is 0.013301 and 0.001023, respectively. As a result, the improvement rate of the proposed method on the WSDream-RT dataset is equal to $0.013301/0.052464 = 25.35\%$ and on the WSDream-TP dataset $0.001023/0.005724 = 17.87\%$. In summary, the proposed method has improved basic method [5] by an average of 21.61%. Also, the results of the evaluation of the basic and proposed methods (presented in Tables A1–A6 in Section 4.3 are statistically evaluated using the statistical method of T-test.

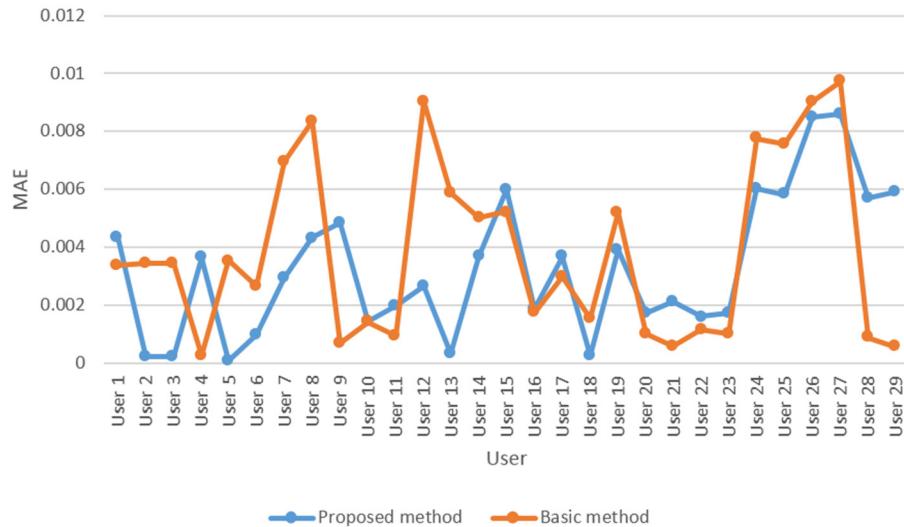


Figure 11. Comparison of the error rate of the basic method and the proposed method for service 4 in the WSDream-TP.

In the following, we study the impact of the matrix density on the recommendation performance of the proposed approach. We used densities in range from 1% to 9% using step of 2. The experiment results are shown in Figures 12 and 13.

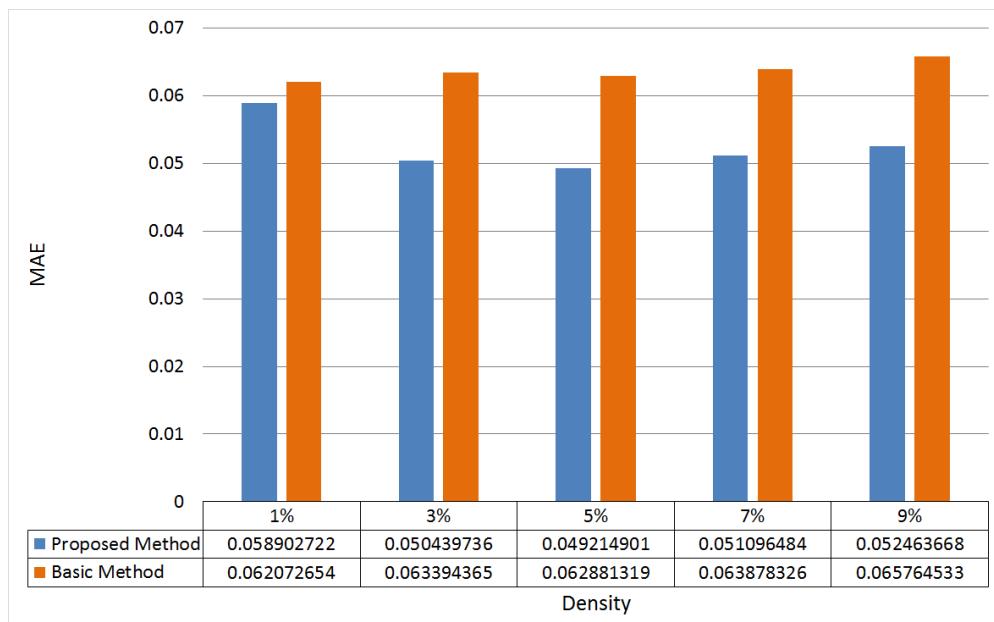


Figure 12. Comparison of the error rate of the basic method and the proposed method under different sparsity levels in the WSDream-RT.

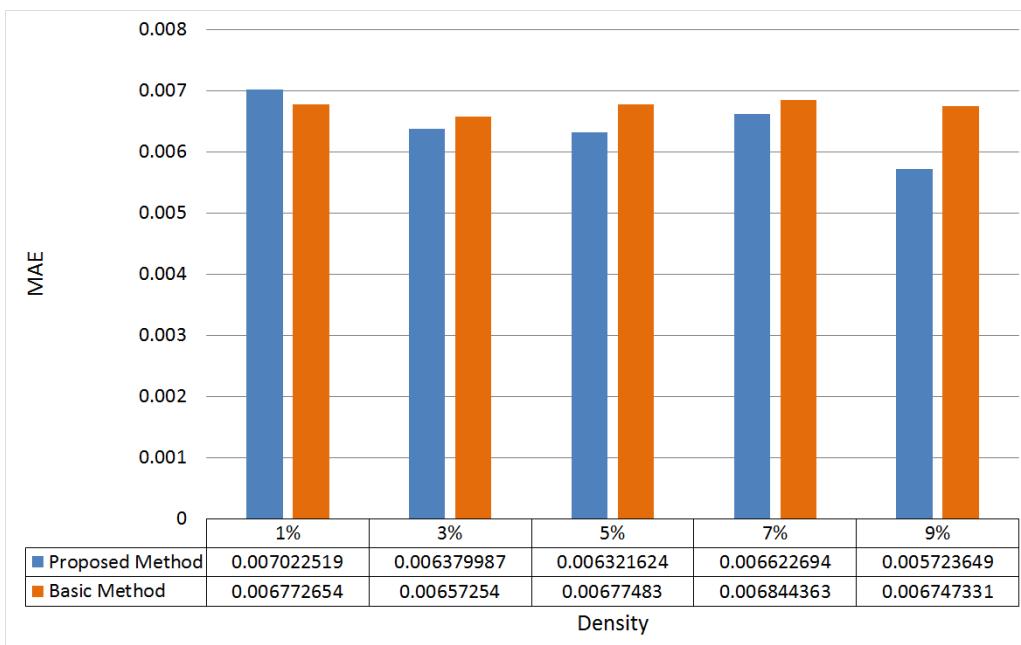


Figure 13. Comparison of the error rate of the basic method and the proposed method under different sparsity levels in the WSDream-TP.

Results in Figures 12 and 13 show that in both WSDream-RT and WSDream-TP datasets (except for one out of 10 experiments) the mean absolute error (MAE) of the proposed method was lower than the base method. This result clearly indicates better (more accurate) performance of the proposed method. This result is achieved by more accurate use of user and cloud services data (by applying formula 1 and formula 2 as shown in Figure 2) in the cloud services recommendation.

Definitely one of the serious challenges in the recommender system is the lack of data whether the lack of service metadata or the problem of lack of user feedback (which is also expressed as the problem of cold start). These challenges are important in evaluating recommendation methods. As stated, the purpose of reporting these results was to provide a clear picture of the performance of the proposed method with respect to the scarcity of WSDream-RT and WSDream-TP datasets. But it should be noted that if the sparsity of the history matrix is more than 99%, i.e., its density is less than 1%, we face the cold start problem. There may also be the limited availability of metadata. In these cases, according to the basic method [5], while encouraging users to complete their profile and cloud service providers to complete cloud services description and metadata, the recommendation of cloud services is based on explicit preferences.

Given recent advances in cloud services, containerizing services are being used as one of the new methods to deploy services. However, even with containerizing the services, their IP location will be clear. Consequently, having their IP, the proposed method will still have its application. It should only be noted that if the IP of the container is altered, this change must be applied to the user-service matrix; moreover, the new IP must replace the previous IP. Of course, this point is also common between a service that is deployed in this way and a service that does not use the containerization method. Furthermore, it is possible to change the IP of the service in both cases. Consequently, there is not much difference in the method of recommending cloud services in terms of using or not using containerized services.

The proposed method did not pay attention to the time contextual factor. For example, Ding et al. have paid attention to this factor [27]. A distributed recommendation of cloud services can also be mentioned. For example, in the method presented by Qi et al., information about several cloud providers such as Amazon and IBM was used to recommend cloud services to users [7]. The method presented in this study did not pay attention to the limitations of real-time applications. These limitations are important in areas such as online gaming, the Internet of Things, real-time mobile software, and big data processing. One of the authoritative papers that mention some of these cases is a study conducted by Zhang et al. [8].

4.3. The significance of the proposed recommendation process improvement

So far, several studies have been conducted to recommend cloud services, which were introduced in the Section 1 above. After selecting and improving the basic method [5], the improvement of this method is examined in the current subsection. The degree of improvement is provided by considering the location of cloud services and improving the formula for determining the similarity between users in the basic method. Therefore, the T-test is used to calculate the rate of improvement of the basic method. According to this statistical method, the probability value of p indicates the probability of T_0 being significant. T_0 here is the lack of a significant statistical difference between the calculated results of the absolute error of the basic method and the proposed method.

The T-test is performed in Excel software on two datasets, including 1) the results of the calculation of the error of the basic method and the proposed method on the WSDream-RT dataset and 2) the results of the calculation of the error of the basic method and the proposed method on the WSDream-TP dataset. These two datasets are presented in detail in the appendix. After performing the T-test on the mentioned data, two numbers were obtained for each dataset, 0.0000598037 and 0.014114, respectively. Thus, T_0 can be confirmed with a probability equal to any of the reported numbers; however, since each of the reported numbers is less than 0.05, in fact, T_0 can be rejected with the possibility of supplementation of these numbers. Therefore, with a probability of more than 0.9999401963 and 0.985886, a significant improvement in the proposed method can be confirmed.

4.4. Performance analysis

Performance is one of the most significant points in recommending services. The complexity of the basic method is discussed in the same article. However, regarding conciseness, the changes that have been made in our study and the analysis of changes are addressed.

The changes are using a more accurate similarity determination formula in comparison to the base article and refining the user-service matrix records based on the article. Given the performance, the first case has no additional performance overhead since the formula determining the similarity rate has been altered; moreover, no additional operations have been added. This is the case during the implementation of the third step in Figure 2. Therefore, the claimed improvement can be obtained simply by refining the similarity formula among users without additional overhead and changing the clustering method.

The second case, which is illustrated as the first and second steps in Figure 2, has the complexity of $O(n^2)$. As the application of the formula $P(u,s)$ is $O(1)$ and regarding the user-service

matrix, which has n users and m service, the amount of $P_{u,s}$ for this matrix is $O(nm)$ so that according to $n > m$, the complexity of these calculations reaches a maximum of $O(n^2)$. After calculating the amount of $P_{u,s}$ for the user-service matrix, which has the complexity $O(n^2)$, the row for each service is extracted based on the amount of $P_{u,s}$ of that service. The number of elements in this row is also n . Consequently, the maximum number of m -services that each row of user-service matrix information has n entries will be extracted; furthermore, the filtered user-service matrix will be achieved. This second step in Figure 2 is also performed with the complexity $O(n^2)$. Therefore, the execution cost of the first two steps is equal to $O(n^2) + O(n^2) = O(n^2)$. The overhead of the third step changes is also $O(1)$. As a result, the overall cost of changes is $O(n^2)$.

5. Conclusions

In this study, we introduced a context-aware cloud service recommendation called SaaSRec+. SaaSRec+ receives a list of services and recommends some of the users. In this method, to recommend cloud services, the following general steps have been defined: 1) context-aware collaborative filtering, 2) context-aware content-based filtering, and 3) ranking. The items covered in this study are: 1) Improving cloud service recommendation by adding a contextual preference, and 2) Improving cloud service recommendation by using a precise similarity determination formula. In the evaluation section, this study uses 5 times more data than those evaluated in the SaaSRec+ method. Test data were obtained from the valid WSDream dataset. According to the evaluation, and considering the MAE criterion and with the help of the T-test, the proposed recommendation method is a significant improvement over the existing method.

Cloud service recommendation methods can be used in real-time applications. The proposed method does not pay attention to the limitations of real-time applications. Therefore, in future work, we can consider these limitations. As an example, it is possible to improve the execution speed by using cloud processing power or parallelizing the processing flow of the proposed method. Improving the speed of implementation of the proposed method is also important in non-real-time applications; this is due to the fact that the expansion of cloud services and the increasing volume of feedback from cloud service users, will increase the volume of user-service matrices, and as a result, the cloud service recommendation will take more time.

Acknowledgments

Open Access Funding by the Publication Fund of the TU Dresden.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, 2011. Available from: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.

2. B. P. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in *2009 Fifth International Joint Conference on INC, IMS and IDC*, (2009), 44–51. doi: 10.1109/NCM.2009.218.
3. F. Aznoli, N. J. Navimipour, Cloud services recommendation: reviewing the recent advances and suggesting the future research directions, *J. Netw. Comput. Appl.*, **77** (2017), 73–86. doi: 10.1016/j.jnca.2016.10.009.
4. D. Fang, X. Liu, I. Romdhani, P. Jamshidi, C. Pahl, An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation, *Futur. Gener. Comput. Syst.*, **56** (2016), 11–26. doi: 10.1016/j.future.2015.09.025.
5. Y. M. Afify, I. F. Moawad, N. L. Badr, M. F. Tolba, A personalized recommender system for SaaS services, *Concurr. Comput.*, **29** (2017), e3877. doi: 10.1002/cpe.3877.
6. Y. Jiang, D. Tao, Y. Liu, J. Sun, H. Ling, Cloud service recommendation based on unstructured textual information, *Futur. Gener. Comput. Syst.*, **97** (2019), 387–396. doi: 10.1016/j.future.2019.02.063.
7. L. Qi, X. Zhang, W. Dou, Q. Ni, A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data, *IEEE J. Sel. Areas Commun.*, **35** (2017), 2616–2624. doi: 10.1109/JSAC.2017.2760458.
8. M. Zhang, R. Ranjan, M. Menzel, S. Nepal, P. Strazdins, W. Jie, et al., An infrastructure service recommendation system for cloud applications with real-time QoS requirement constraints', *IEEE Syst. J.*, **11** (2017), 2960–2970. doi: 10.1109/JSYST.2015.2427338.
9. R. B. Bohn, J. Messina, F. Liu, J. Tong, J. Mao, NIST cloud computing reference architecture, in *2011 IEEE World Congress on Services*, (2011), 594–596. doi: 10.1109/SERVICES.2011.105.
10. H. Schütze, C. D. Manning, P. Raghavan, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
11. X. Fan, Y. Hu, Z. Zheng, Y. Wang, P. Brézillon, W. Chen, CASR-TSE: context-aware web services recommendation for modeling weighted temporal-spatial effectiveness, *IEEE Trans. Serv. Comput.*, **14** (2017), 58–70. doi: 10.1109/TSC.2017.2782793.
12. U. Shardanand, P. Maes, Social information filtering: algorithms for automating “word of mouth”, in *Proceedings of the SIGCHI conference on Human factors in computing systems, 1995*, (1995), 210–217. doi: 10.1145/223904.223931.
13. X. Wu, B. Cheng, J. Chen, Collaborative filtering service recommendation based on a novel similarity computation method, *IEEE Trans. Serv. Comput.*, **10** (2015), 352–365. doi: 10.1109/TSC.2015.2479228.
14. M. Deshpande, G. Karypis, Item-based top-N recommendation algorithms, *ACM Trans. Inf. Syst.*, **22** (2004), 143–177. doi: 10.1145/963770.963776.
15. H. Sun, Z. Zheng, J. Chen, M. R. Lyu, Personalized web service recommendation via normal recovery collaborative filtering, *IEEE Trans. Serv. Comput.*, **6** (2013), 573–579. doi: 10.1109/TSC.2012.31.
16. H. Mezni, T. Abdeljaoued, A cloud services recommendation system based on Fuzzy Formal Concept Analysis, *Data Knowl. Eng.*, **116** (2018), 100–123. doi: 10.1016/j.datak.2018.05.008.
17. Q. Wei, W. Wang, G. Zhang, T. Shao, Privacy-aware cross-cloud service recommendations based on Boolean historical invocation records, *Eur. J. Wirel. Commun. Netw.*, **2019** (2019), 1–8. doi: 10.1186/s13638-018-1318-8.

18. L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, X. Zhang, Privacy-preserving distributed service recommendation based on locality-sensitive hashing, in *2017 IEEE International conference on web services (ICWS)*, (2017), 49–56.
19. Z. Zheng, H. Ma, M. R. Lyu, I. King, Qos-aware web service recommendation by collaborative filtering, *IEEE Trans. Serv. Comput.*, **4** (2010), 140–152. doi: 10.1109/TSC.2010.52.
20. C. Zhang, Z. Li, T. Li, Y. Han, C. Wei, Y. Cheng, et al., P-CSREC: a new approach for personalized cloud service recommendation, *IEEE Access*, **6** (2018), 35946–35956. doi: 10.1109/ACCESS.2018.2847631.
21. Z. Y. Chai, Y. L. Li, Y. M. Han, S. F. Zhu, Recommendation system based on singular value decomposition and multi-objective immune optimization, *IEEE Access*, **7** (2018), 6060–6071. doi: 10.1109/ACCESS.2018.2842257.
22. L. Guo, D. Mu, X. Cai, G. Tian, F. Hao, Personalized QoS prediction for service recommendation with a service-oriented tensor model, *IEEE Access*, **7** (2019), 55721–55731. doi: 10.1109/ACCESS.2019.2912505.
23. A. S. B. Priya, R. S. Bhuvaneswaran, Cloud service recommendation system based on clustering trust measures in multi-cloud environment, *J. Ambient Intell. Humaniz. Comput.*, **2020** (2020), 1–10.
24. K. Indira, M. K. K. Devi, Multi cloud based service recommendation system using DBSCAN algorithm, *Wirel. Pers. Commun.*, **115** (2020), 1019–1034. doi: 10.1007/s11277-020-07609-3.
25. R. Hentschel, S. Strahringer, A broker-based framework for the recommendation of cloud services: a research proposal, in *Conference on e-Business, e-Services and e-Society*, (2020), 409–415. doi: 10.1007/978-3-030-44999-5_34.
26. Z. Zheng, Y. Zhang, M. R. Lyu, Investigating QoS of real-world web services, *IEEE Trans. Serv. Comput.*, **7** (2014), 32–39. doi: 10.1109/TSC.2012.34.
27. S. Ding, Y. Li, D. Wu, Y. Zhang, S. Yang, Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model, *Decis. Support Syst.*, **107** (2018), 103–115. doi: 10.1016/j.dss.2017.12.012.

Appendix

Following Subsection 4.3 of this paper, to complete Table 2, the results of the calculation of the error of the basic method and the proposed method on the WSDream-RT and the WSDream-TP datasets are presented in Tables A1–A4. However, to comply with the abbreviation, the provision of information related to services 5, 6, and 7 has been abandoned.

Table A1. Basic method error calculation results on wsdream-RT dataset.

	Service1	Service2	Service3	Service4
User1	0.195908	0.015361	0.011978	0.014908
User2	0.228584	0.013043	0.017025	0.012878
User3	0.193472	0.01284	0.017025	0.012878
User4	0.002787	0.007971	0.012158	0.004562
User5	0.204432	0.009188	0.012766	0.00943
User6	0.219451	0.008782	0.012158	0.008213

Continued on next page

	Service1	Service2	Service3	Service4
User7	0.106615	0.016299	0.021224	0.016796
User8	0.352026	0.017332	0.020398	0.018242
User9	0.809831	0.007447	0.006823	0.006738
User10	0.229045	0.009918	0.010734	0.007974
User11	0.323488	0.008888	0.009911	0.00715
User12	0.095233	0.014975	0.018587	0.015631
User13	0.134043	0.011636	0.014622	0.012084
User14	0.014244	0.012347	0.012014	0.01204
User15	0.147102	0.009043	0.015098	0.01226
User16	0.040269	0.001762	0.013393	0.005417
User17	0.200237	0.001967	0.012983	0.002956
User18	0.261423	0.00345	0.006161	0.003723
User19	0.058522	0.015503	0.027954	0.019363
User20	0.146707	0.000361	0.1959	0.007091
User21	1.000104	0.00735	0.043365	0.00639
User22	0.165166	0.008285	0.0443	0.007324
User23	0.373125	0.007817	0.043365	0.007091
User24	0.282315	0.013727	0.016166	0.021015
User25	0.21836	0.013298	0.016381	0.0208
User26	0.005346	0.01569	0.018834	0.015611
User27	0.207801	0.015907	0.018834	0.016262
User28	0.019185	0.008087	0.010052	0.013589
User29	0.270643	0.000121	0.357807	0.003161

Table A2. Proposed method error calculation results on WSDream-RT dataset.

	Service1	Service2	Service3	Service4
User1	0.185104	0.006304	0.003425	0.006348
User2	0.123142	0.007646	0.034516	0.014403
User3	0.084656	0.007423	0.034516	0.014403
User4	0.105595	0.001729	0.012905	0.001797
User5	0.201655	0.003535	0.013808	0.00902
User6	0.107913	0.001433	0.011705	0.004314
User7	0.031071	0.012118	0.033342	0.01863
User8	0.208619	0.013127	0.032535	0.020042
User9	0.853587	0.000849	0.014933	0.006122
User10	0.161634	0.001569	0.018762	0.007331
User11	0.122863	0.102691	0.005978	0.000838
User12	0.033204	0.00596	0.014103	0.008643
User13	0.186061	0.008299	0.023182	0.013265
User14	0.108944	0.022752	0.080447	0.039432
User15	0.232907	0.019669	0.083325	0.039637
User16	0.011844	0.001412	0.044295	0.016182

Continued on next page

	Service1	Service2	Service3	Service4
User17	0.14723	0.001616	0.043887	0.013735
User18	0.008469	0.002805	0.004246	0.000255
User19	0.181397	0.006255	0.024002	0.012767
User20	0.002163	0.005448	0.189131	0.006726
User21	0.985671	0.001194	0.016962	0.006122
User22	0.019671	0.001801	0.007883	0.000463
User23	0.197193	0.001596	0.016962	0.006726
User24	0.117532	0.008923	0.020643	0.013168
User25	0.047681	0.008454	0.020877	0.012934
User26	0.042881	0.01124	0.02656	0.016903
User27	0.182368	0.013822	0.028896	0.019911
User28	0.011596	0.004417	0.008945	0.00062
User29	0.292929	0.001144	0.335908	0.008403

Table A3. Basic method error calculation results on WSDream-TP dataset.

	Service1	Service2	Service3	Service4
User1	0.003113	0.00636	0.008842	0.003359
User2	0.004261	0.006327	0.010052	0.003449
User3	0.001925	0.006011	0.010052	0.003449
User4	0.002383	0.004393	0.007692	0.000249
User5	0.003419	0.006042	0.008859	0.00353
User6	0.004463	0.005484	0.007692	0.00265
User7	0.001765	0.012895	0.021568	0.006943
User8	0.003754	0.014778	0.019503	0.008344
User9	0.004504	0.00052	0.001215	0.000681
User10	0.001352	0.003196	0.004599	0.001406
User11	0.006131	0.002054	0.003309	0.00092
User12	0.000191	0.016215	0.024091	0.009023
User13	0.001458	0.010722	0.014926	0.005868
User14	0.001974	0.010278	0.007299	0.005017
User15	0.002708	0.004871	0.01426	0.005221
User16	0.002235	0.004668	0.006177	0.00173
User17	0.001495	0.004469	0.006739	0.00299
User18	0.003819	0.002824	0.004355	0.001548
User19	0.002624	0.00729	0.012335	0.005202
User20	0.001317	0.005367	0.066319	0.000994
User21	0.004672	0.002277	0.013696	0.000573
User22	0.001126	0.003333	0.015186	0.001139
User23	0.003178	0.002801	0.013696	0.000994
User24	0.002698	0.013665	0.018775	0.007748
User25	5E-05	0.0129	0.019305	0.007542
User26	0.002495	0.017713	0.024866	0.009039

Continued on next page

	Service1	Service2	Service3	Service4
User27	0.003978	0.018142	0.024866	0.009715
User28	0.001947	0.001295	0.002283	0.000872
User29	0.001714	0.003027	0.070399	0.000545

Table A4. Proposed method error calculation results on WSDream-TP dataset.

	Service1	Service2	Service3	Service4
User1	0.003058	0.008209	0.01134	0.004336
User2	0.003058	0.000589	0.002804	0.000224
User3	0.000433	0.000235	0.002804	0.000224
User4	0.003023	0.003415	0.003104	0.003626
User5	0.001509	0.000592	0.000626	6.6E-05
User6	0.002749	0.001254	0.002011	0.000978
User7	0.003613	0.006447	0.00937	0.002927
User8	0.001871	0.008319	0.007317	0.004318
User9	0.002934	0.010873	0.017017	0.004841
User10	0.002845	1.37E-05	0.005575	0.001432
User11	0.002581	0.001264	0.006989	0.001964
User12	0.000327	0.004025	0.006686	0.00264
User13	0.000859	0.001117	0.001892	0.000313
User14	0.001886	0.007159	0.003573	0.003669
User15	0.00351	0.005274	0.015919	0.005981
User16	0.003123	0.00585	0.007121	0.001867
User17	0.002204	0.005566	0.007924	0.003667
User18	0.005997	0.002498	0.004305	0.000252
User19	0.003584	0.004981	0.009204	0.003907
User20	0.002019	0.010774	0.074035	0.001728
User21	0.00503	0.003912	0.002213	0.002106
User22	0.001847	0.002964	0.000875	0.001598
User23	0.002016	0.003442	0.002213	0.001728
User24	0.002289	0.008955	0.012265	0.006017
User25	0.000491	0.008181	0.012801	0.005808
User26	0.002486	0.015806	0.023238	0.008505
User27	0.003087	0.01585	0.02194	0.008583
User28	0.002587	0.010803	0.015308	0.005709
User29	0.004722	0.012218	0.083529	0.005899

Table A5. The absolute value of the difference between the values in Tables A1 and A2.

	Service1	Service2	Service3	Service4
User1	0.010804	0.009057	0.008553	0.008561
User2	0.105442	0.005397	0.017491	0.001525
User3	0.108817	0.005417	0.017491	0.001525

Continued on next page

User4	0.102808	0.006242	0.000747	0.002765
User5	0.002777	0.005653	0.001042	0.00041
User6	0.111538	0.00735	0.000453	0.003899
User7	0.075544	0.004181	0.012118	0.001834
User8	0.143407	0.004205	0.012137	0.0018
User9	0.043756	0.006598	0.008111	0.000616
User10	0.06741	0.008349	0.008028	0.000643
User11	0.200625	0.093803	0.003933	0.006312
User12	0.062029	0.009015	0.004483	0.006988
User13	0.052018	0.003337	0.008559	0.001181
User14	0.0947	0.010405	0.068433	0.027392
User15	0.085805	0.010626	0.068227	0.027377
User16	0.028425	0.00035	0.030901	0.010765
User17	0.053007	0.000351	0.030904	0.010778
User18	0.252954	0.000644	0.001915	0.003468
User19	0.122875	0.009247	0.003952	0.006596
User20	0.144544	0.005087	0.006769	0.000365
User21	0.014433	0.006156	0.026403	0.000267
User22	0.145495	0.006484	0.036416	0.006861
User23	0.175931	0.006221	0.026403	0.000365
User24	0.164783	0.004804	0.004476	0.007847
User25	0.170679	0.004844	0.004496	0.007866
User26	0.037534	0.00445	0.007725	0.001292
User27	0.025433	0.002084	0.010062	0.003649
User28	0.00759	0.003669	0.001107	0.012969
User29	0.022286	0.001022	0.021899	0.005242

Table A6. The absolute value of the difference between the values in Tables A3 and A4.

	Service1	Service2	Service3	Service4
User1	5.49E-05	0.001849	0.002498	0.000977
User2	0.001203	0.005738	0.007248	0.003225
User3	0.001492	0.005777	0.007248	0.003225
User4	0.000641	0.000978	0.004588	0.003377
User5	0.001909	0.005451	0.008233	0.003464
User6	0.001714	0.00423	0.005681	0.001673
User7	0.001848	0.006447	0.012199	0.004016
User8	0.001884	0.006459	0.012185	0.004025
User9	0.00157	0.010353	0.015802	0.00416
User10	0.001492	0.003183	0.000976	2.52E-05
User11	0.00355	0.00079	0.00368	0.001045
User12	0.000135	0.01219	0.017405	0.006382
User13	0.000599	0.009605	0.013034	0.005555
User14	8.76E-05	0.003119	0.003726	0.001348

Continued on next page

	Service1	Service2	Service3	Service4
User15	0.000802	0.000403	0.001659	0.00076
User16	0.000888	0.001182	0.000944	0.000137
User17	0.000709	0.001097	0.001185	0.000677
User18	0.002178	0.000326	4.98E-05	0.001296
User19	0.000959	0.00231	0.003131	0.001295
User20	0.000702	0.005407	0.007716	0.000734
User21	0.000358	0.001634	0.011483	0.001532
User22	0.000721	0.000369	0.014311	0.00046
User23	0.001162	0.000641	0.011483	0.000734
User24	0.000409	0.00471	0.00651	0.001731
User25	0.000441	0.004719	0.006504	0.001733
User26	8.88E-06	0.001907	0.001628	0.000534
User27	0.000891	0.002293	0.002927	0.001132
User28	0.000639	0.009508	0.013025	0.004837
User29	0.003009	0.009191	0.013129	0.005354



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)