A Privacy Preserving Improvised Approach for QOS aware Web Service Recommendation

Arif Ahmed Khan
Computer Science Engineering
Government College Of engineering
Aurangabad, India
arif.k8893@gmail.com

V. A. Chakkarwar
Computer Science Engineering
Government College Of engineering
Aurangabad, India
vrush.a143@gmail.com

***Abstract*— Suggestions for web services along with the recommendation have been very popular lately in IT research. When creating composite quality-of-service sequences based on service-oriented systems, it is imperative to evaluate the non-performance characteristics of potential candidates for service selection. In this article, we will present Improvised clustering recommendation System (ICRS), a model for analyzing and predicting the absolute reliability of the predicted atomic web service that can also evaluate the reliability of a continuous service call based on data collected from previous invocations with a purpose. To improve the accuracy of the most advanced forecast models, we include user-specific parameters and user-specific contexts. To reduce the flexibility problems presented in a modern way, we will collect the above usage data using the K-means and ICRS grouping algorithm. When evaluating the different qualities of our models, we experimented with Planet DB Dataset and the services registered in WSDream dataset. The results confirm that our models can be guessed and scaled accurately.**

Keywords*— Web Service, Predictive models, Vectors, K-means ICRS clustering, QoS, recommendation, Collaborative filtering, QoS values, Web service recommendation; QoS prediction , privacy preservation.

I. INTRODUCTION

Web services are software components that support interoperability between machines. The increase in visibility and acceptance of web services on the World Wide Web require effective suggestions and techniques to guide the right web user from the various web services available. The amount of web services to add to the Quality of Service written as (QoS) [1] is generally used to describe the ability of web services to respond to the user with the appropriate data. Some features of web services are independent of the user's location, while others vary according to their efficiency with different users in different locations. Some vary in attributes such as price, popularity, availability, etc., external providers or registered third parties. (For example, UDDI) Often provides a value independent of QoS characteristics. In turn, some characteristics of the user's QoS are dependent and have different values for different users. (For example, response time, failure rate, etc.). The evaluation of web services on the client side requires that the real web service is executed and meets the following disadvantages:

1) First, the invocations to the web service really determine the cost to the subscriber and use the resources of the service provider. Some web services may also be charged.

2) Second, there may be many web analytics providers and some reasonable web services in the evaluation list may not be detected and registered by the subscriber.

3) Ultimately, most users are not experts in evaluating web services and limited time constraints to restrict specific web services. However, if there is not enough evaluation on the client side, you will not be able to find out the true value of the QoS property. Choosing the right services and web suggestions is very difficult.

To have more advanced functionality, SOA allows designers to integrate atomic services into more complex services. While the integration of services is important for developers to choose high quality atomic suppliers [1], [2] due to the quality of their applications, they are reliable in terms of quality and performance. In order to create powerful composite applications, developers must receive reliable information about all atomic functions and non-functional attributes such as time, response, reliability, availability.

This forecasting concept focuses on the reliability of atomic services as one of the most important non-functional characteristics. Here, the reliability of the service means the probability that the service request is complete. Credibility in descriptive terms is more desirable for web services, since inviting services is a unique and relatively rare event. Based on the definitions used, the reliability of the previous sentence sample can be estimated as the relationship between the successful number and the total number of calls processed. However, passing examples becomes a very challenging task for several reasons.

There are family characteristics for service-oriented systems to perceive reliability from the perspective of users and service providers. [6] In general, the reliability values are calculated based on the information that the service provider has. The receiver can represent a specific user due to the vibration offered by various context variables. From the user's perspective, additional barriers to collecting information relate to the cost of ownership and performance issues. The strategy to overcome this credibility assessment challenge is to obtain a sample of some historical records. But related and use predictive algorithms to evaluate the reliability of missing records.

Experts have predicted many forms of prediction based on collaborative filtering, which is often used in modern guidance systems. [12] Although the existing collaborative filtering procedures are achieving results after the actual work has been completed. This may be enough as an important reason for the complication that may be associated with prediction-based precision within dynamic spaces and environments where there may be scalability problems caused by sample sizes. About the accuracy of the joint filter, predictions will provide accurate guidance in a static environment where the data collected is relatively stable. That means that the registry will be updated for a long time. However, the service-oriented system is active on the Internet, which is a dynamic environment where the provider records key load patterns during the day. [16] In a dynamic environment, the user's reliability and perceptions may vary completely with respect to the actual moment of the appeal. In addition, collaborative filtering will gather credibility for individual users and service partners.

II. Recommender System

Users need a special system that understands their interests and recommends the best service. In this case, the Recommendation System can help users with articles that fit their interests, being considered as one of the best solutions. Recommendation systems can be classified as hybrid filters, content filtering, hybrid modeling. [2] The guidance system can help consumers and the most valuable items by calculating similarities among other consumers.

III. Collaborative Filtering Methods

Identical user-oriented web service identities and similar user provisioning, such as mutual filtering is based on the history of the previous web service. The user can rarely restore all the services that QoS (RTT, round trip time) represents for the service that the user did not execute. Therefore, accurate QoS forecasting of web services is imperative for service providers. Depending on the planned QoS, you can choose which service you want. In the Web Services Guide, the main issue of CF is to find similar user groups, similar service groups and service matrices in QoS values generated by the service. The service to the Matrix user is certainly very inferior in real practice. The accuracy of QoS services is greatly reduced. Therefore, we expect the QoS value of the array to look for the previous QoS data for similar users or similar services and recommends a web service with the appropriate QoS value for the active user.

IV. Collaborative Filtering algorithm

The Collaborative Filtering algorithm uses two process

A. Predictive process [3] [4], a numerical value that represents the predicted probability of a web service that can not support some users. The predicted value is the same level as that of the same user that gives the same value.

B. Prediction, which lists the N recommended items that the active user likes the most. This recommended list includes users who do not have access to the web service. This joint filtering algorithm with Top N recommendations [13] is called the collaborative filtering procedure. It is not useful for all users to actively evaluate QoS due to the high cost of running a service. To solve this problem, collaborative QoS predictions have recently been proposed and have become an important step in the introduction of QoS web services. [3] [4], [5] In particular, there are two categories. The CF methods have been studied to predict the QoS of web services.

In recent literature the two types of filtering algorithms are as follows.

1) *Model-Based Collaborative filtering*.
2) *Collaborative filtering from memory*.

A. Model-Based Collaborative Filtering

It is modeling based on scoring data. In other words, we will extract some data from the data set and use it as a model to guide it without the need for a complete set of data. This can be useful for both speed and flexibility. Using modeling algorithms, we can study the QoS model, which is a model used to predict QoS models based on CF algorithms, including Bayesian (probabilistic) and ICRS models, [6] CF techniques using models [6] delivery of predefined predictors of adjusted QoS data. Trained models can predict unknown QoS values. Matrix factorization [7] is a standout amongst the most famous CF models. First, it was presented to solve the problem of QoS prediction, the Matrix factorization model [7] that is considered a scarce problem and generally more effective than the nearest method. A typical example is a user's guide (eg, UPCC [8]) that uses similar user QoS data to predict.

B. The Memory Based Collaborative FilteringApproach

The memory algorithm uses a collaborative data filtering algorithm based on Breese et al. [9] Trying to find a similar user to the active user, (That is, the user on which we want to make predictions), and use the configuration to predict scores for active users. Algorithm that uses memory to predict that is stored in memory through the use of data (Users and QoS data). They can be classified in the closest approximation algorithm and in the introductory step. The N-neighbor algorithm is the memory most frequently used by the user's CF procedure. The CF method uses the observed QoS data to calculate the similarity between the intended user or the intended service and uses the data to determine the similarity between the user and the service. The next QoS suggestion is the introduction of Top-N N web services, many of which will be used only with interest. The Top N technique [10] are used to establish relationships with users or user services and to calculate recommendations.

V. Related Work

There are several ways to model the reliability of traditional software systems in the literature. [7] Web services are still a dynamic invention of software with functionality through a public access interface. Internet is a dynamic environment where the results of recruitment services depend on a variety of different effects, which define the context of entry. Therefore, the traditional software reliability modeling method is not adequate to evaluate the reliability of a web service. While completely new models are developed for service-dependent systems, most researchers generally focus on the reliability of service components. The existing methods do not vary the QoS depending on the user's location to consider Z. Zheng, X. Chen, Z. Huang, H., X. Liu and Sun [13], offers a new collaborative filter design for large instructions of web service. in QoS Location Locations. First, it incorporates a mechanism based on CF and memory used for web service suggestions, which clearly shows the accuracy, orientation and complexity of time, compared to the previous service submission process. Second, they create visually appealing interfaces to view the recommended web services, which better understand the service's performance. Their algorithms use QoS characteristics of users in multiple regions. Depending on the characteristics area, the closest distillation algorithm is proposed to generate QoS predictions. The final instructions of the service are on the map when placing the infrastructure of the QoS area to show and help users to accept the term.

Similarly, M. Tang, Y. Jiang, J. Liu and X. Liu [6] propose methods for known locations. CF-based web services are intended for users who can recommend both user-based and service-based websites. Unlike collaborative user-based filtering to find similar users for the target user, instead of looking for user groups, they focus on the physical users closest to their goals. Likewise, the similarity measure of the existing mutual filtering services used by the service location data, based on the hybrid filtering technology, also change. After finding similar users and services, they used similar measures to predict missing QoS values. Introduce the users of the web with the best QoS values in a conscious location. How they obtain QoS data First time history and user location information The location manager handles the active user location information and the destination services. QoS values are lost for active users. The user matrix of the service records each QoS experience in the web service of the invoking user. For similar users, user similarity metrics are based on the QoS data of the calculated user found in nearby active users defined by the location data manager.

Similarly, service similarity metrics are calculated based on service-specific QoS records that are close to the destination service defined by the location manager. Similar users and related services are used for active users and target services, or CF algorithms, and elements used for user forecasts and missing QoS searches. We briefly describe each type of collaborative filtering in the following sections:

A. The Memory-Based Collaborative Filtering (MBCF)

MBCF is a technique commonly used in next-generation recommendation systems [7], [8], [9] This filter technique retrieves data or patterns by correlating data with data received from various sources, entities such as agents, views or resources. The advantage of memory interoperability in a filter is that the lack of information for a particular entity can be predicted using the data available from the most similar statistical agencies. This type of collaborative filter uses a user matrix to store data for prediction of reliability. Each value based in the matrix obtained represents the reliability factor of the service that is accessed and that which can be perceived by the user. Matrix systems can have millions of users and services, while service partners: new providers come in real time. In addition, each user has access only to the subset of the service. Therefore, the user matrix is sparse and has an empty cell number that reflects the missing trust that must be foreseen. Two kinds of memory interoperability filters can be applied. UPCC collects data collected from different users and predicts missing confidence values using the information available from similar users. The IPCC collects information from different services and estimates the value of the missing credibility, based on the value of a very similar service. The hybrid methods [9], [10] achieve better predictions using both the information extracted from users and similar services, and the predicted loss values due to the UPCC and the IPCC are a linear combination.

B. The Model-Based Collaborative Filtering

It is known that the models are complex and difficult to use. These methods often involve more sophisticated techniques, such as machine learning algorithms or data mining, to learn predictive models by recognizing complex patterns using training data. For example, Yu et al. ,he proposed a normative basis to track the use of a normal feasibility matrix to predict the reliability of a web service. Zheng and Lyu use matrices in their approach with small hypotheses that affect the credibility of the user. However, these guidelines may not really incorporate any specific environmental parameters. In the prediction process, the general representative of the filtered interoperability model is the linear regression technique [6]. For example, the linear regression [7], [8] is the most suitable for predicting numbers. There are no existing linear regression models that incorporate environmental parameters in the reliability predictions of the web service. Collaborative filtering Hybrids are very effective in dealing with the disadvantages of memory-based filtering. [9] However, the main disadvantage of these methods is that they often rely on a more specific domain, which describes a system within a system. Demonstrating that this is a challenging task to obtain this information in practice. In our recent work [10], we discussed the disadvantages of joint filtering by improving accuracy, predictability and flexibility.

In [20] a method has been discussed that utilizes the QoS information present at client-side and that neglects contextual attributes present in a service. Based upon the basic fact that, quality of Web service is usually affected because of its contextual features, the author then proposes an altogether new QoS-aware Web service recommendation system, that considers the contextual attribute similarities of various services. The method that is proposed first extracts the contextual attributes from WSDL[20] files to cluster Web services based on their feature similarities, and utilize the improved matrix factorization method to allow recommendation of services for the users

However, the proposed model (LUCS) is available in environments with model input parameters. For example, we classify the services in the service layer, considering the complexity of the service calculation, and we consider that each service class is explicitly known as an input parameter. Because the number of services with missing input parameters results in worse predictability. These defects are handled by ICRS and the linear regression methods are described in the following sections.

VI. System design

A. *ICRS Prediction Overview*

ICRS, a model to predict the reliability of web atomic services. In order to improve predictive accuracy, we will define user parameters and specific environments that define the service call context to match the most relevant predictive model. In addition to the scalability, we grouped the aggregation samples into three dimensions related to the parameters defined by the K-means IC measurement algorithm. K-means implies between different ICRS clustering algorithms [11] which is a combination of accurate and fast data when it comes to service reliability data.

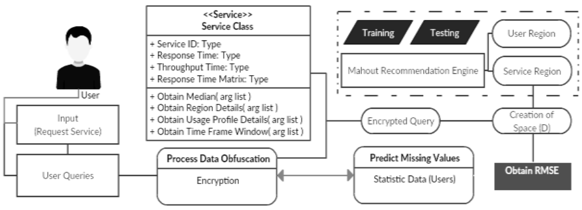

Fig 1. Web Service QoS Prediction

As the amount of web services available on the Internet grows, service providers focus on QoS and not work. Most QoS features include inactive features such as response time, forwarding, availability. The use of web services [11], [12] (Wang Wang et al., 2013). Elements of service (Feng, Ngan et al. 2013). Service suggestions (Wu et al 2011. Jiang, Liu et al.,High) and other popular topics in computer services. In this section, we will present work related to QoS-aware Web service recommendations. [12].

They suggest how to predict the QoS value of a web service using traditional methods of user unification and a collaborative filtering method. Its improved approach does not necessarily require any call and assistance from the web service by analyzing similar user QoS data. Users discover the right web services. In the evaluation of web services [12], in the paper report, to reduce the impact of web service calls to real web services, they chose a unique operation of the web services to evaluate and use the power of action. This is with performance when offering web services.

B. *Web Service Recommendation Methods Based on Personalized Collaborative Filtering*

There are several ways to choose a web service and filter recommendations. Y. Jiang, J. Liu, Tang, X. Liu [14] provides a collaborative way of introducing custom filtering [18] for web services. An important part of these techniques is to calculate the similarity of web services. Unlike the similarity values of the Pearson product moment correlation coefficient (PCC), they consider the personal impact of user interactions and personal effects when measuring the similarity of the calculated services. Based on the way to obtain the similarity of the web services, they developed a hybrid filter technology (HICP) for custom integration algorithms based on the user.

VII. Framework of QoS-Aware Web Service Recommendation

This section is an online services search scenario to illustrate the research problem of this document. The basic idea of this approach is that users who are closer are more likely to experience a similar service than those who are far away. We use the idea of user interoperability in our web-based reference system. The QoS data generated by the user offers more precise service suggestions that users will receive while analyzing. User data based on data provided by the user. When compiling a QoS record, our guided approach is designed as a two-phase process. In the first phase, we divide our users into regions based on their physical location and their past QoS experience. In the second phase of web services, we find similar users for current users and the predicted QoS for services. Non-optimized QoS services are recommended to current users.

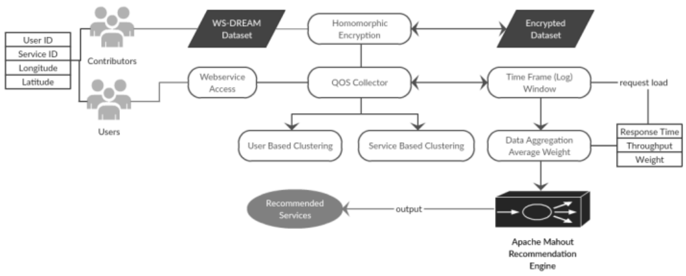

Fig 2. Web service recommendation process

A. *Location Information Representation, Acquisition and Processing*

In this section we will discuss how to display, receive and process location data for web services and subscribers, which are essential to implement a recommended method for web-based services.

1. *Location Representation*

We specify the user's location as [IP address], [country], [IP address], [AS], [Latitude], [Longitude]. The internet consists of thousands of ASs that connect with each other.

However, users in the same AS will not always close geographically and vice versa. Even if two users are in the same city, however, it may appear in different AS, which explains why we choose AS instead of other geographical locations, such as latitude and longitude, to indicate the user's location.

2. *Location Information*

Acquiring the location of web services and users can be done easily. According to the IP address of the user, it is already known to obtain a complete user location. We need to identify the AS and the country in which they are based on the IP address. There are a number of services and databases available for this purpose. In our implemented work, we have made IP with allocation of AS and IP to map the country using the autonomous GeoLite system number.

3. *Similarity Computation and Similar Neighbor Selection*

We use weighted PCC to calculate the similarities between users and web services, which will take into account the individual characteristics of QoS. [16] Finally, the author analyzes the user's integration and web-based location with similar neighboring selections.

4. *Similar Neighbor Selection*

This selection is a very important step of CF. In the traditional form of CF used by users, the Top-N neighbor selection algorithm is used consistently. [16] Choose a similar N-user. Use the most active as a neighbor. Similarly, a superior N neighbor algorithm can be used to select the N-Web service most similar to a traditional Top-N web destination service. Ignore this problem and continue choosing section N This is because the affected neighbors are not the same as the target user (service). This will reduce the accuracy of the forecast. Therefore, the abandonment of neighbors of neighboring upper class sets N is better if the similarities do not exceed zero. Second, web users may be aware of similar QoS values in a few web services.

5. *Prediction and Recommendation using Apache Mahout*

Mahout offers undistributed instructional tools. You must pass data values defined by the system for the items. And the results of this tool will be the estimation of a system for other articles.

The components provided by Mahout to build a prediction and recommendation engine are as follows:

- DataModel
- UserSimilarity
- ItemSimilarity
- UserNeighborhood
- Recommender

Algorithm (Psuedo Code)

1. Create Data Model Object, Tanimoto Coefficient Similarity class requires a data model object, which holds a file that contains the Users, Value, and Preferences details of a record.
2. Create UserSimilarity Object
3. Create UserNeighborhood object
4. Create Recommender Object
5. Recommend Items to a User

```
//Creating data model

DataModel datamodel = new FileDataModel(new File("data")); //data


//Creating UserSimilarity object.

UserSimilarity usersimilarity = new PearsonCorrelationSimilarity(datamodel);


//Creating UserNeighbourHHood object.

UserNeighborhood userneighborhood = new ThresholdUserNeighborhood(3.0, usersimilarity, datamodel);


//Create UserRecomender

UserBasedRecommender recommender = new GenericUserBasedRecommender(datamodel, userneighborhood, usersimilarity);


List<RecommendedItem> recommendations = recommender.recommend(2, 3);


for (RecommendedItem recommendation : recommendations) {

System.out.println(recommendation);

}
```

6. *Privacy Preservation*

Impersonation is a key element in the predictive QoS of collaborative web services for the maintenance of personal information. The basic idea of data spoofing is data interference.

Raw data with randomness in such characteristic:

a) Random should be able to guarantee that there is no important data (that is, the QoS value of each user can be inferred from the obfuscated data).

b) Although the individual data are limited, the aggregate data of these users can also be estimated.

We use rui to represent the QoS value that the user uses for the web service i, ru for the vector, all the values of QoS are evaluated by the user u, and similarly Iui and Iu are shown. Cu = | Iu | is the amount of QoS values that the user has evaluated. U In our data, privacy is the key technique used. When hiding sensitive information, the Laplace mechanism [6] has a different privacy when increasing the Laplace transmission.

Where, e, the privacy parameter is set by the user. U. Δf is determined by the distribution of the QoS value, which is: Δf = max (rui-ruj) after the mimic sends its own disguise value Qu to the data server. Confidential data about the source data are kept at random. However, the total information of the user can also be estimated. Therefore, QoS prediction can be achieved through direct access to Qui.

$$Qui = qui + Laplace\ (\Delta f / e)$$

Accuracy is good when the number of users is very large. These features are useful for calculations based on general data.

This selection is a very important step of CF. In the traditional form of CF used by users, the Top-N neighbor selection algorithm is used consistently. [16] Choose a similar N-user. Use the most active as a neighbor. Similarly, a superior N neighbor algorithm can be used to select the N-Web service most similar to a traditional Top-N web destination service. Ignore this problem and continue choosing section N This is because the affected neighbors are not the same as the target user (service). This will reduce the accuracy of the forecast. Therefore, the abandonment of neighbors of neighboring upper class sets N is better if the similarities do not exceed zero. Second, web users may be aware of similar QoS values in a few web services.

Considering the location associated with the QoS of a web service, the author has included the location of the user and the web service in selecting similar neighbors.

B. *User-Based QoS Value Prediction*

The authors present a CF-based approach called ULACF [16] using the CF method [17] used by traditional users for predictions. This equation may not be accurate to predict the QoS value of a web service. Due to the QoS factors of web services, such as time and response time, their destination parameters and their values vary. Therefore, QoS predictions based on the average QoS [17] [18] [19] perceived by users (such as r (u)) are defective. Users close to 2 target users should have more confidence in QoS prediction than anywhere else.

Our system utilizes Blowfish encryption to record and store service's response time along with user's longitude and latitude. The system also generates Response Time Matrix and Throughput Matrix that is later used for generating recommendations for users based upon their location and other service parameters.

The system we offer employs Blowfish encryption algorithm [21], which encrypts the data obtained from the service and the user. The algorithm uses a 64-bit hexadecimal code, along with a variable length key. The main advantage of using this algorithm is that it requires less memory. The pseudorandom code algorithm in round 16 deals with each XOR cycle with the F (x) function. Each cycle uses an expansion key to encrypt the data.

VIII. EXPERIMENTAL SETUP

We evaluated the methods in the WS-DREAM data set [12], a common-use QoS data set that is widely used to predict QoS predictions. The ICRS can also achieve good prediction accuracy compared to the Basic Guidelines (CLUS) and the Contracting Parties (UIPCC).

The RMSE sensitivity is one of the most orderly concerns regarding its use as a metric factor. The central existence of abnormalities and probability of occurrence is clearly explained by the normal distribution that refers to the use of RMSE. In practice, it may be the reason to throw away the error. The order is greater than the other samples when calculating the RMSE, especially if the number of samples is limited. If the bias is severe, you may have to eliminate the system error before recalculating the RMSE value.

$$RMSE = \frac{\sqrt{e1^2 + e2^2 + e3^2 + e4^2 + en^2}}{n} \quad \text{...............} \quad \mathbf{(1)}$$

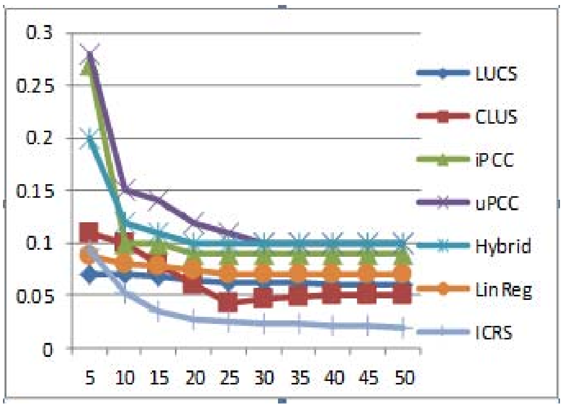

Fig 3 RMSE Values and Computational Performance of the Prediction

Figure 3 describes values of RMSEs for pseudo errors generated randomly with zero mean and considering the density of data as 50% exhaustively, we have summarized and calculated the results with average values. which shows that RMSE converges to 0.02938 to 0.11, an approximation to the expectation.

IX. CONCLUSION

The assembly of the various QoS properties is important for the realization of the web service technology. Due to the growing popularity of web services technology and the latency of the selection and integration of dynamic services, several service providers now provide parallel services. QoS is a modified factor to discriminate functionally similar web services. To make it more problematic, understanding is the progression of hiding unique data with arbitrary characters or data. The recommendation of a web service helps users find a mandatory service that has become a major problem in the calculation of the service.

REFERENCES

[1] Jianxun Liu, Mingdong Tang, Member, IEEE, Zibin Zheng, Member, IEEE, Xiaoqing (Frank) Liu, Member, IEEE, Saixia Lyu, "Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation", IEEE Transactions on services computing, manuscript Val X , No XX, 2015

[2] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware web service recommendation via non-negative tensor factorization," in Proc. of the 23rd International World Wide Web Conference (WWW), 2014, pp. 585–596.

[3] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu, "Web service recommendation via exploiting location and QoS information," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1913–1924, 2014.

[4] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, ''Personalized QoS-Aware Web Service Recommendation and Visualization,'' IEEE Trans. Serv. Comput., vol. 6, no. 1, pp. 35-47R1st Quart., 2013.

[5] R. Salakhutdinov and A. Mnih, "Probabilistic matrix Factorization," in Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS), 2007.

[6] M. Tang, Y. Jiang, J. Liu, X. F. Liu: Location-Aware Collaborative Filtering for QoS-Based Service Recommendation. in Proc. 10th International Conference on Web Services, Hawaii, USA, June 2012, pp.202-209

[7] S. Wang, Q. Sun, and F. Yang, ''Towards Web Service Selection Based on QoS Estimation,'' Int'l J. Web Grid Serv., vol. 6, no. 4, pp. 424-443, Nov. 2012.

[8] G. Kang, J. Liu, M. Tang, X. F. Liu, K. K. Fletcher. "Web Service Selection for Resolving Conflicting Service Requests". Proceedings of International Conference on Web Services. IEEE Computer Society, pp. 387-394, 2011.

[9] Breese, J. S., Heckerman, D., Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14 th Conference on Uncertainty in Artificial Intelligence (UAI 1998), Madison, Wisconsin, USA, July 24-26, 1998, 43-52.

[10] Y. Zhang, Z. Zheng, M. R. Lyu, "WSExpress: a QoS-aware search engine for Web services", in Proc. 8th IEEE International Conference on Web Services, Miami, FL, USA, July, 2010, pp.83-90.

[11] M. Alrifai, and T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition," in Proc. of the International World Wide Web Conference, Apr. 2009, pp. 881- 890.

[12] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Adv. Artificial Intelligence, vol. 2009, 2009.

[13] Z. Zheng, H. Ma, M.R. Lyu, and I. King. "WSRec: A Collaborative Filtering Based Web Service Recommendation System," in Proc. 7th International Conference on Web Services, Los Angeles, CA, USA, pp. 437- 444, 2009.

[14] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, ''Personalized QoS-Aware Web Service Recommendation and Visualization,'' IEEE Trans. Serv. Comput., vol. 6, no. 1, pp. 35-47R1st Quart., 2013.

[15] Shao. L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H. (2007). Personalized QoS prediction for web services via collaborative filtering, Proceedings of the 14th IEEE International Conference on Web Services (ICWS 2007), Salt Lake City, Utah, USA, July 9-13, 2007, 439-446.

[16] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," ACM Transactions on Web, 1(1):6, 2007.

[17] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen,''Scalable Collaborative Filtering Using ICRSter-Based Smoothing,''in Proc. 28th Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval,2005, pp. 114-121.

[18] K. Karta, ''An Investigation on Personalized Collaborative Filtering for Web Service Selection,'' Honours Programme Thesis, Univ. Western Australia, Brisbane, Old., Australia, 2005.

[19] Z. Zheng, H. Ma, M. R. Lyu, and I. King "QoS-Aware Web Service Recommendation by Collaborative Filtering", IEEE Trans. on Services Computing, 2011, vol.4, no.2, pp.140-152.

[20] Shun Li, Junhao Wen, Fengji Luo, " A New QoS-Aware Web Service Recommendation System Based on Contextual Feature Recognition at Server-Side", IEEE transactions on network and service management, vol. 14, no. 2, June 2017

[21] Ms NehaKhatri , Prof. V. K Kshirsagar, "Blowfish Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 80-83,2014.