Web Services Recommendation system

Kashmira Dolas
ksd1747@g.rit.edu

Rashmi Dodeja
rjd1227@g.rit.edu

Sandhya Murali
sm2290@g.rit.edu

ABSTRACT

The concept of Web services has become a widely applied paradigm in research and industry, with the number of services published on the Internet increasing rapidly over the last few years[12]. The rapid increase has brought forth challenges in terms of large data which leads to time consuming expensive computation in-order to find services for recommendation. To overcome these challenges, many efficient mechanisms of web service discovery in response to user's request have been proposed in order to leverage web service selection process. This is a survey paper which discusses the state-of-the-art approaches which use temporal and spatial features, collaborative filtering (CF), clustering based on various factors, etc.

1. INTRODUCTION

In the recent years, Service-Oriented Computing (SOC) has proved to be a favorable computing paradigm in the field of software engineering and distributed systems[9].

Web services are a set of software components to provide interoperable machine-to-machine interaction over a network [3]. Web service recommendation deals with recommending the top k web services based on functional and non-functional aspects. These non-functional aspects are known as Quality of Service(QoS) and include response time, throughput, availability and so on.

The paper, [2] talks about making a personalized QoS web service recommendation for different users and help the users select the suitable web service based on his/her requirements. Kumara et al. discusses the various approaches for web service recommendations, the authors categorizes the approaches into the following: 1) Content-based approach - is concerned with similarities on basis of service functionalities, either derived from data or meta-data, 2) CF - focuses more on recommending services based on the past service usage of other end-users with similar profiles, and 3) Hybrid approach - is a combined approach that uses both content-based approach and CF [12]. Hence, [12] proposes a cluster-based recommendation which uses hybrid approach for the CIS (currently invoked web services).

The papers [2, 5, 6, 9, 10, 13, 15, 16], use hybrid recommendation system that utilizes both functional and QoS recommendation approach. The paper, [15] talks about improving the performance of recommending web service in response to a user's request by using a personal profiling mechanism. Liu et al. works on improving the existing CF based Web Service recommendation technique by leveraging location information while selecting similar users and web services [13]. The authors of [5] and [6], published two papers that extend on context aware service recommendation (CASR). In 2015, the authors proposed Context-Aware Services Recommendation based on Temporal Effectiveness (CASR-TE) method which uses an enhanced time decay model derived from combination of existing time decay function and current similarity measurement methods. The next year, they extended the CASR to CASR-TADE method that considers the effectiveness of True Abnormal Data Evaluations (TADE).

The paper, [9] discusses about the increase in attraction of SOC domain. Kang et al. talks about recommending web services to user based on the user's interest and Qos preferences by exploring the web service usage history of the user [10]. The paper, [16] considers reputation as an important evaluation metric and recommend web services to the target user by computing a reputation score for each of the candidate web service. The paper [1], is inspired by the Google's instant search and proposes a real-time approach for recommendation by extensively using the execution log of composite services to recognize services and A* search to prune the list for top-k services.

The research issues are as follows:

- As number of web services increases rapidly, it becomes difficult to identify the optimal services using functional and non-functional properties thereby making web service discovery a critical task in terms of computation and time efficiency.
- The QoS performance is based on the Internet environment and user location since different users may observe different QoS performance for the same web service.
- The time complexity of memory-based CF recommender systems is an issue which makes it difficult to make recommendations for multiple users at real time.
- The current systems don't provide transparency of the recommendation results due to which the makes it difficult for users to trust recommendation results. Therefore, efficiently recommend non-malicious and trustworthy web services a critical task.
- An issue with CF is that it considers services that have different functionality and are used by similar users where as these services will never be desired by the target user.

- CF does not take into consideration the value-added services to the services used by the target user for recommendation.

- Non-standard ontology creates an issue since the semantic similarity is required to consume the web service during web service discovery.

- Some recommendation methods fail to consider dynamic features of the services as well as any temporal changes.

- The traditional similarity mining processes like CF are very likely to generate relatively big QoS value deviations and inadvertently decrease the accuracy of QoS prediction.

- The current approaches fail to exploit user history and recommend web services based on the best QoS value on a certain QoS criterion due to which many redundant web services may be present in the top-k recommendation list.

The organization of the paper is as follows. Section 2 discusses the existing approaches in the field of web service recommendation. Section 3 states the motivation behind surveying the trending methods in web service recommendation and why we selected [9] paper for implementation. Section 4 details the progress made so far along with the technologies used. The paper is concluded with section 5.

2. EXISTING APPROACHES

This section discussed all the recently proposed approaches that are actively used today for web service recommendations. These approaches can be categorized into content-based approaches, CF based approaches and hybrid based approaches. Depending on the categorizes, the common and non overlapping themes of the reference papers are covered as follows:

2.1 Content based approaches

2.1.1 *Modeling Temporal Effectiveness for Context - aware Web Services Recommendation*

The CASR-TE approach is structured using 4 steps as shown in the fig. 2.1.1. This figure describes the entire architecture with the methods and metrics used.

The methods used for comparison with CASR-TE are RBA (recommendation by all), UPCC (similarity between users using PCC), IPCC (similarity between services using PCC), CASR, CASR-UP (CARS using user location) and ITRP-WS (time decay in UPCC). The authors have experimented with different thresholds, different ratios of training and testing data and using different number of neighbors. The evaluation metric used for comparison is mean absolute error (MAE) and root mean square error (RMSE). According to CASR-TE method, the best threshold is 0.775.

2.1.2 *Exploring the Effectiveness of True Abnormal Data Elimination in Context-aware Web Services Recommendation*

As explored in most of the review papers in this survey, CF which usually uses Pearson Correlation Coefficient (PCC) measurement is the traditional approach to predict

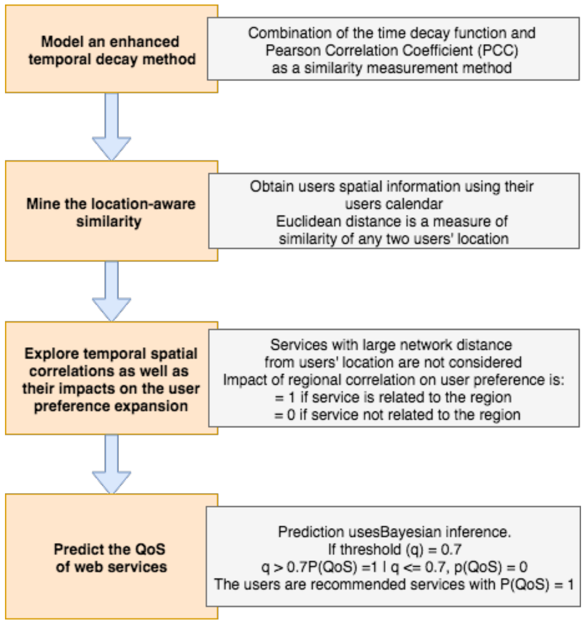

Figure 1: CASR-TE Architecture

QoS values. To avoid the error rate in prediction of QoS values, [6] finds the true abnormal data. After this true abnormal data is recognized, this data is simply eliminated from the prediction data. This prevents QoS values to have high deviation and accuracy increases. This method has 5 steps. The first step is to perform context-aware similarity mining. The context factor used here is location of the user. The next step is identify and eliminate the true abnormal data. Then, item based and user based CASR-TADE are performed. The results are combined and averaged to obtain the final prediction.

2.2 Collaborative Filtering based approaches

2.2.1 *Personalized QoS-Aware Web Service Recommendation and Visualization*

The paper proposes a system that is divided into two phases. In the first phase, the users are divided into different regions based on the user's physical location and the historical QoS experience on web services. In this phase, the approach mainly focuses on the QoS properties that are prone to change and those QoS properties that can be easily obtained from each user[2]. The response time non-functional property is considered in this approach. From these parameters, a region is defined which are a group of users who are in the same location and have similar QoS profiles[2]. This region creation is a three-step process. In the first step, users with similar IP addresses are put into a small region and extract region features. The second step is calculating similarity between different regions. In the third step, the highly correlated regions are aggregated to form a certain number of large regions. In the second phase, similar users are determined for the target user and missing QoS values are predicted for the unused services[2]. This can be done after region aggregation since searching neighbors and making predictions for an active user can be computed quickly.

The top-k best services will be predicted and provided by the user.

2.2.2 *Collaborative Personal Profiling for web service ranking and recommendation*

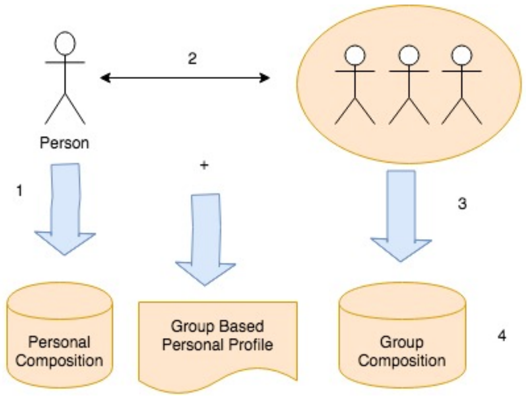

Figure 2: CF based personal profile generator

The fig 2.2.2 represents the flowchart of building a collaborative filtering based personal profile. The personal composition involves gathering all the web service composite transactions up to a particular interval where a web service composite transaction is obtained by combining several web services based on their dependencies. Group composition is obtained by identifying the set of similar users and then collaborating the web service composite transactions of these users. Group based personal profile is created by creating a group association rule set. This group association rule set is obtained by determining the association among the web services that form several web service composite transactions. After determining a set of web composition transactions from a set of similar users, a re-ranking mechanism is proposed for filtering the services quickly.

2.2.3 *Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation*

The fig, 2.2.3 represents the work flow of the proposed system. It deals with the following components. The user and service location information handler obtain the location information using network IP address of the users and web services. The find similar users and services components finds similar users and web services based on the QoS experiences and locations. The hybrid QoS prediction combines both the user and service based QoS prediction results to make the final QoS predictions for an active user. Based on the final predictions, optimal web services are recommended to the active user.

2.2.4 *Time Aware and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering*

The paper, [9] describes the solution by creating a user-service matrix containing the observed QoS value and corresponding timestamps. The missing QoS values are predicted at the current time when the recommendation is requested

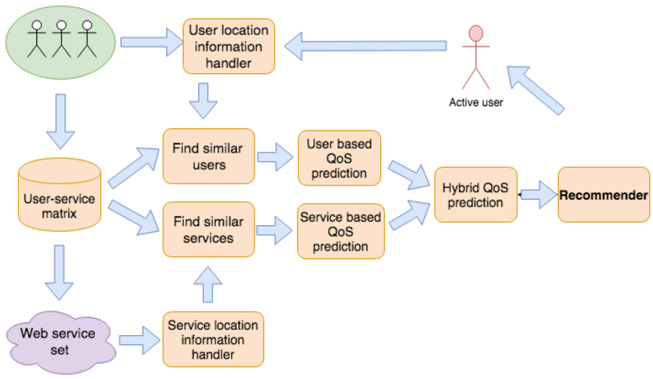

Figure 3: Architecture of location based Web service recommendation model

using the historical information. So, when a target user requests a functionality, the recommender system fetches all the web services which meets the functional requirements of the user. Next, user similarities and service similarities are inferred using the personalized random walk mechanism and time-aware similarities is present in the historical training data. Using these mechanisms, QoS predictions are made and used for user-based and service-based predictions which are then merged together. These merged results are used by recommender component to evaluate the web services and return it to the user based on multi-criteria decision-making procedure.

2.2.5 *Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems*

The proposed solution consists of two phases: a malicious feedback rating detection and a feedback rating adjustment. The first phase detects malicious feedback ratings which are collected in a data collector using a cumulative sum control chart[16]. In this way, when more malicious users are entering the system, there will be more negative feedback ratings in a sampling interval[16]. In order to find this anomalous shift, cumulative sum control chart is applied for monitoring the change in detection based on hypothesis testing in order to detect malicious feedback ratings. The second phase computes the feedback similarity of different users using Pearson Correlation Coefficient in order to influence the preferences of different users and adjust their feedback ratings. After these two phases, the next stage is the preventing scheme which contains two stages: Activation stage and Blocking stage. In the activation stage, bloom filter is implemented to identify the IP address that are associated with malicious feedback ratings. In the blocking stage, the remote service broker blocks the malicious users from rating those web services[16].

2.2.6 *Instant Recommendation for Web Services Composition*

The performance of WSCRec approach depends on the performance of one of its main component, the composite service recommender. The probabilities candidate services are calculated using Bayes theorem. The factors for the calculation of probabilities are based on the logs from the execution log database component of WSCRec and partial composed services. QoS is used to rank the services. The composite service recommender uses A* search known as H-WSCRec which is improved version of WSCRec. Three different pruning techniques viz., path pruning, pro pruning and sim pruning are used for improving the efficiency and to obtain the top-k composite services.

2.3 Hybrid based approaches

2.3.1 *Cluster-Based Web Service Recommendation*

Fig. 2.3.1 depicts the architecture of the proposed approach and following is the description of each step.

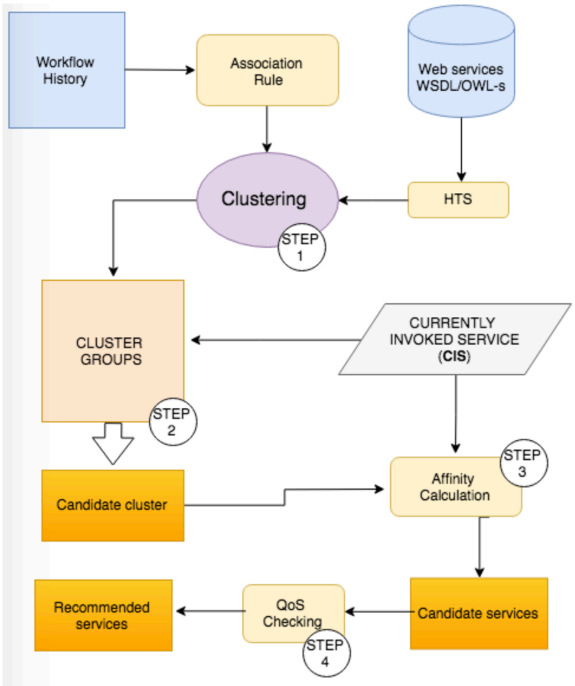

Figure 4: Architecture of cluster-based approach for recommendation

The first factor that considers functionality, semantic similarity values between web services are calculated using the hybrid term similarity (HTS) method, which was also proposed by the authors of this paper in [11]. The HTS calculates the similarity values using information retrieval (IR) and ontology learning. In step 1, the clusters are defined using the two factors discussed before. For step 2, using the CIS and the cluster pool, we calculate the affinity of CIS with all the cluster representatives (center value) to select the candidate cluster. After which, considering the then selected candidate clusters and CIS, the similarity values (again using HTS and association rule) are calculated in step 3. Step 4 is to rank the selected services in candidate cluster according to the similarity values, from which the better services with top rated QoS are selected as candidate services to be recommended to the end-user utilizing the CIS.

For proving the results, the authors implemented an approach without clustering using WordNet and the edge-count-based (EDB) methods to determine the services similar to the CIS. By plotting a bar graph of precision performance of each approach against the top 1 to top 50, authors prove that their approach outperforms every single time with a significant margin. [12] also demonstrates the clusters formed based on different factors to show that using 2-factor method by computing the HTS and association yields more sensible set of clusters.

2.3.2 *Diversifying Web Service Recommendation Results via Exploring Service Usage History*

The approach has four components namely functional evaluation, non-functional evaluation, diversity evaluation and diversified evaluation and diversified web service ranking. The functional evaluation is subdivided into 2 parts: Functional Evaluation-1 and Functional Evaluation-2. Functional evaluation-1 evaluates the historical interest with the web services based on content-based similarity measure that is obtained using text similarity. Functional evaluation-2 evaluates user's interest with the web services based on collaborative filtering approach. Non-functional evaluation determines the user's potential QoS preference by exploring the user's web service usage history. Diversity evaluation constructs a web service graph based on the similarity values computed between service candidates. Diversified web service ranking algorithm is performed based on the web service graph to determine top-k recommendation list for the active user.

3. MOTIVATION

We have selected 'Time Aware and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering' for implementation. [9] uses QoS factors along with spatial and temporal information in a collaborative filtering technique. This technique is used to provide personalized QoS prediction which recommends top-ranked web services. The paramount inspiration behind selecting [9] is that the selected paper exploits maximum features such as temporal and spatial for predicting QoS considering the similarity between users, which in-turn is used for web service recommendation. Also, [9] which is published in 2015, is among the recent published approaches.

4. IMPLEMENTATION

The Dataset used for implementation is a freely available dataset provided by Zhang et al. [17]. It consists of 142 users and 4500 web services with each web service being invoked at 64 time-intervals by every user. The non-functional parameters such as response time and throughput are considered for web service recommendation in our project.

4.1 Dataset

The attributes and description is as follows:

User Information consists of user information.

userid: unique user id.

IPaddress: IP address of the user.

country: Country residing in.

ipno: IP Number.

autonomoussystem: Autonomous System of the IP

latitude: Latitude

longitude: Longitude

Web Service Information consists of web service information.

serviceid: unique user id.

wsdladdress: IP address of the user.

serviceProvider: Country residing in.

IPAddress: IP Number.

country: Country of web service.

ipno: IP Number.

autonomoussystem: Autonomous System of the IP

latitude: Latitude

longitude: Longitude

Web Service Response Time Information consists of web service response time information.

userid: user id.

serviceid: service id.

timesliceid: time slice interval id.

responsetime: response time at corresponding time slice interval id.

Web Service Throughput Information consists of web service throughput information.

userid: user id.

serviceid: service id.

timesliceid: time slice interval id.

throughput: throughput at corresponding time slice interval id.

4.2 Data preprocessing

On studying the dataset, certain pre-processing steps were carried out which involved removing duplicates from Web Service Response Time information and Web Service Throughput information. The web service information was incorporated with an additional column 'category'. It is used to categorize the web services based on its functionality. The need for this column arose as we were not able to filter the web services based on the functional requirements of the user. In addition to that, web service information file was converted to UTF-8 from ISO-8859 format for loading in PostgreSQL database using a python procedure.

4.3 Methodology

After preprocessing, the dataset is loaded into PostgreSQL [4] database. The application, initially, records the user location and desired functionality. Now, based on the location entered, neighboring users are selected from the user information table. Also,based on the desired functionality, candidate web services are generated. In order to recommend a web service efficiently from the candidate web services, location plays an important role as users of different location will have different experience(Quality-of-service values) for the same web service. After considering the location factor, it also becomes important to consider the temporal information as the web services used in the past will not necessarily provide the same quality of service. Thus considering location and time as the two main factors, a weight matrix is calculated between two users for the same location using Pearson Correlation Coefficient. Response time and throughput are considered as QoS parameters in our implementation.

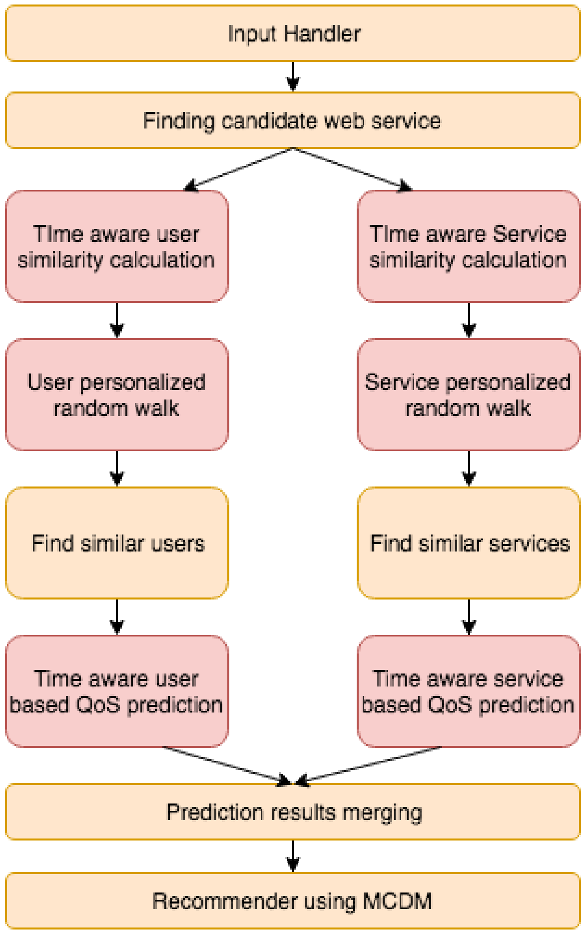

Figure 5: Time aware and data sparsity tolerant web service recommendation based on improved CF

$$W_{ij} = \sum_{S_k \in S}\left(\frac{1}{P_{ik}P_{jk}}\sum_{p=1}^{P_{ik}P_{jk}}\left(q_{ik}^{p}-\bar{q}_i\right)\left(q_{jk}^{p}-\bar{q}_i\right) \times f_1\left(t_{ik}^{p},t_{jk}^{p}\right)f_2\left(t_{ik}^{p},t_{jk}^{p}\right)\right) \Big/ \left(\sqrt{\sum_{S_k\in S}\frac{1}{P_{ik}}\sum_{p_i=1}^{P_{ik}}\left(q_{ik}^{p_i}-\bar{q}_i\right)^2} \times \sqrt{\sum_{S_k\in S}\frac{1}{P_{jk}}\sum_{p_j=1}^{P_{jk}}\left(q_{jk}^{p_j}-\bar{q}_j\right)^2}\right) \qquad (1)$$

In (1), W_{ij} refers to the similarity weight between two users. P_{ik} and P_{jk} represent number of times users u_i and u_j invoked s_k. q_{ik}^{p} and t_{ik}^{p} are the QoS value and time-stamp of u_i's, p_i invocation of s_k. f_1 and f_2 are the exponential decay functions ranging from 1 to 0. $\bar{q}_i$ and $\bar{q}_j$ are the average

QoS values for users u_i and u_j. The weight matrix computed only considers direct similarity between two users. Similarly, the weight matrix for similarity between two web services is also computed. Both weight matrices are sparse as active users call only a few web services and even the popular web services are called by only a few users. Therefore, many users may have no common web services and many web services may have no common user. To avoid this problem and exploit the indirect similarities between the two users, a personalized random walk algorithm is applied. After obtaining the direct and indirect similar users and similar services, missing QoS values are predicted based on user similarity and service similarities and historical QoS information The time-aware user-based QoS prediction result can be shown in (2)

$$\hat{q}_{ik} = \bar{q}_i + \frac{\sum_{u_j \in R_{u_i}} s_{ui}(j)\left(\frac{1}{P_{jk}}\sum_{p=1}^{P_{jk}} f_3(t_{jk}^p)(q_{jk}^p - \bar{q}_j)\right)}{\sum_{u_j \in R_{u_i}} s_{ui}(j)\left(\frac{1}{P_{jk}}\sum_{p=1}^{P_{jk}} f_3(t_{jk}^p)\right)} \tag{2}$$

In (2), $\hat{q}_{ik}$ represents the QoS predicted value for the target user u_i on the candidate service s_k. The time-aware service-based QoS prediction is computed in a similar fashion. The QoS values calculated with respect to users and services are merged using two confidence weights con_u and con_v to balance the prediction results. The con_u computation is shown in (3)

$$con_u = \sum_{u_i \in R_{u_i}} \frac{s_{u_i}(j)}{\sum_{u_j \in R_{u_i}} s_{u_i}(j)} \times s_{u_i}(j) \tag{3}$$

The computation of con_v is analogous to con_u. The weights h_u and h_v are used along with user-based and service-based prediction results to obtain consolidated Qos prediction. The h_u is computed as shown in (4)

$$h_u = \frac{\lambda \times con_u}{\lambda \times con_u + (1 - \lambda \times con_s)} \tag{4}$$

The computation of h_v is analogous to h_u. The final QoS predicted value is computed in (5)

$$\hat{q}_{ik} = h_u \times \hat{q}_{ik}^u + h_s \times \hat{q}_{ik}^s \tag{5}$$

After the QoS values are predicted, recommendation for optimal web service is done using multi-criteria decision making technique [7]. The top-3 web services are recommended to the user.

The implementation is done in Java and uses Jama [8] library for Matrix related computations. The back end is implemented using PostgreSQL [4] database. A graphical user interface is developed using Swing [14] to provide user-friendly experience.

5. CONCLUSION

In this project, we implemented the paper [9] which proposes a modified neighborhood-based collaborative filtering technique to make more accurate QoS prediction. The approach takes into consideration the time component in similarity computation as well as QoS prediction which improves the QoS prediction accuracy. This approach is also scalable to multiple QoS parameters and also provides data support for evaluation of web services and in recommending the optimal web service.

References

[1] L. Chen, J. Wu, H. Jian, H. Deng, and Z. Wu. Instant recommendation for web services composition. *IEEE Transactions on Services Computing*, 7(4):586–598, 2014.

[2] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun. Personalized qos-aware web service recommendation and visualization. *IEEE Transactions on Services Computing*, 6(1):35–47, 2013.

[3] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu. Web service recommendation via exploiting location and qos information. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1913–1924, 2014.

[4] K. Douglas and S. Douglas. *PostgreSQL: a comprehensive guide to building, programming, and administering PostgresSQL databases*. SAMS publishing, 2003.

[5] X. Fan, Y. Hu, R. Zhang, W. Chen, and P. Brézillon. Modeling temporal effectiveness for context-aware web services recommendation. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 225–232. IEEE, 2015.

[6] X. Fan, Y. Wang, Y. Ma, Y. Hu, and X. Liu. Exploring the effectiveness of true abnormal data elimination in context-aware web services recommendation. In *Web Services (ICWS), 2016 IEEE International Conference on*, pages 300–307. IEEE, 2016.

[7] F. Hdioud, B. Frikh, and B. Ouhbi. Multi-criteria recommender systems based on multi-attribute decision making. In *Proceedings of international conference on information integration and web-based applications & services*, page 203. ACM, 2013.

[8] J. Hicklin, C. Moler, P. Webb, R. F. Boisvert, B. Miller, R. Pozo, and K. Remington. Jama: A java matrix package. *URL: http://math. nist. gov/javanumerics/jama*, 2000.

[9] Y. Hu, Q. Peng, X. Hu, and R. Yang. Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. *ieee transactions on services computing*, 8(5):782–794, 2015.

[10] G. Kang, M. Tang, J. Liu, X. F. Liu, and B. Cao. Diversifying web service recommendation results via exploring service usage history. *IEEE Transactions on Services Computing*, 9(4):566–579, 2016.

[11] B. T. Kumara, I. Paik, W. Chen, and K. H. Ryu. Web service clustering using a hybrid term-similarity measure with ontology learning. *International Journal of Web Services Research (IJWSR)*, 11(2):24–45, 2014.

[12] B. T. Kumara, I. Paik, T. Siriweera, and K. R. Koswatte. Cluster-based web service recommendation. In *Services Computing (SCC), 2016 IEEE International Conference on*, pages 348–355. IEEE, 2016.

[13] J. Liu, M. Tang, Z. Zheng, X. F. Liu, and S. Lyu. Location-aware and personalized collaborative filtering for web service recommendation. *IEEE Transactions on Services Computing*, 9(5):686–699, 2016.

[14] M. Loy, R. Eckstein, D. Wood, J. Elliott, and B. Cole. *Java swing.* " O'Reilly Media, Inc.", 2002.

[15] W. Rong, B. Peng, Y. Ouyang, K. Liu, and Z. Xiong. Collaborative personal profiling for web service ranking and recommendation. *Information Systems Frontiers*, 17(6):1265–1282, 2015.

[16] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu, and F. Yang. Reputation measurement and malicious feedback rating prevention in web service recommendation systems. *IEEE Transactions on Services Computing*, 8(5):755–767, 2015.

[17] Y. Zhang, Z. Zheng, and M. R. Lyu. Wspred: A time-aware personalized qos prediction framework for web services. In *Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on*, pages 210–219. IEEE, 2011.