Atomic Web Service Reliability Prediction

Preethi W, Binu Rajan M R

Abstract: ***Web service is one of the main supporting underlying technologies in Service Oriented Architecture (SOA). This work is focused on atomic web service reliability, as one of the most important non-functional properties. Service reliability can be defined as the probability that a service invocation gets retrieved successfully, i.e. correct response to the service invocation gets successfully retrieved under the specified conditions and the time constraints. A model-based collaborative filtering approach CLUS (CLUStering) is used to estimate the reliability of an ongoing web service. It considers user, service and environment specific parameters to provide a more accurate description of the service invocation context. Incorporating K-Strings clustering algorithm is highly prominent for clustering of high dimensional data rather than using K-Means algorithm. This aims to generate higher accuracy and efficiency to the prediction model.***

Keywords: reliability prediction, K-Strings, atomic web services, QoS prediction, K-Means.

I. INTRODUCTION

Web service helps end users to directly communicate each other; also it acts as a platform for developing interoperable distributed applications, which allows programmers to communicate with other information providers, without bothering about backend or front-end tasks. Performance of the employed Internet Web services greatly affects the performance of the service oriented systems. To communicate with each other and clients web services are used, they also permit number of applications to communicate from various resources too. They are highly flexible as they are not limited to any operating system or programming languages. Typically these services are offered by third-party providers from various organizations or enterprises that lookup service implementations, supplying service descriptions also providing related technical and business support [1]. Different services are developed using distinct technologies that are deployed over different platforms and are delivered via various communication links. But the Quality of Service (QoS) they offer may vary even though their functionalities are the similar. An important aspect towards acceptability of a web service is how they meet the performance requirement. As web service is one of the main supporting underlying technologies in Service Oriented Architecture (SOA), its performance need to be studied primarily. Behavior of Web Services is dynamic, so that predicting its response time during early stages of Software Development Life Cycle (SDLC) becomes more complex.

Manuscript published on 30 October 2016.
***** Correspondence Author (s)
Preethi W, Department of Computer Science, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram (Kerala)-695018, India.
Binu Rajan M R, Assistant Professor, Department of Computer Science, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram (Kerala)-695018, India.

Rather than functional requirements, non functional requirements, i.e. QoS properties such as, response time, throughput, reliability, failure rate etc. plays a vital role in user's requirement. QoS properties are dynamic in nature that changes frequently in real-time. So that, in service computing, many researches are now a day's carried out on QoS prediction [2]. To construct a reliable system the developer should ensure the reliability of each individual atomic web service involved. Change in QoS of atomic web services may lead to change in QoS of composite web service; it may degrade the performance of a service oriented system.

Among the above mentioned QoS attributes such as availability, reliability, throughput etc. reliability is primarily chosen as the prediction object due to its significance. It is because, they have a close relationship with both hardware and software configuration, network behavior, load, user/service location which all indirectly lead to change the observed reliability value. Behavior of software system becomes abnormal when reliability fails to meet basic requirement, it lead to generate immeasurable loss in several domains such as, bank, military, aerospace etc. that put high demand on reliability. User perceived reliability and service specified reliability vary accordingly [3], [4], [5].

This work focuses on reliability of atomic web service, one of the main non-functional properties. Service reliability can be defined as the probability of successfully completing a service invocation under specified time constraints and conditions. It can be determined using past invocation data samples as the ratio of number of successful invocations to the sum of invocations performed. An efficient way to utilize these samples is to collect partial but relevant sample data from past invocations and then applying prediction algorithms for missing or unknown records. These samples are gathered via collaborative feedback or service monitoring [6], [7], [8]. Here a model named CLUS (CLU Stering) based on collaborative method is introduced, also the K-Means clustering algorithm in CLUS model is replaced by an advanced K-Means algorithm named K-String algorithm which is very good for high dimensional data [9]. In the next section, various related works on reliability prediction methodologies are introduced. Section 3 shows an overview of our prediction model, Section 4 contains the proposed system, section 5 shows the results obtained and section 6 concludes the work. This is how rest of the paper is arranged.

II. RELATED WORKS

To estimate web service reliability characteristics chosen are different for different prediction models or methodologies.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Here, various web service reliability prediction strategies based on various parameters are studied. Web services are dynamic in nature that makes their functionalities available over number of interfaces over internet. Many researchers usually focus on studying service reliability while designing new models for service oriented system [10]-[14]. Studies reveal that collecting sample data for this task is very difficult. As the number of parameters used for generating sample data increases, it in turns helps to increase the prediction accuracy. From many existing approaches for reliability prediction, collaborative filtering is one of the efficient techniques [15]. Collaborative filtering is basically classified into memory-based, model-based and hybrid. Existing works prove that collaborative filtering approaches results in promising result [16], [17]. But their main drawback of is its scalability and accuracy issue, also they need additional storing space for each observed service-user value pair. Such an approach does not scale when millions of users and service occurs. As stated above prediction accuracy depends on variety of factors, i.e. they may have impact when they are considered or not.

2.1 Influencing factors

As internet will always fluctuate depending of various environmental factors or hardware resources, outcomes of service invocations also depends on such factors. Due to this dynamic nature of web service most of the existing traditional methods are not suitable for determining the reliability of web service invocation. While developing a model for reliability prediction various approaches choose various parameters or factors, they generate different impact on different model.

Nonfunctional quality of a service is mainly influenced by the location of service and the user's [18]. Another factor that influences the system performance is the internal service complexity; potentially it impacts the service complexity also [19]. Time of invocation of service has significance as the load in a day varies accordingly [20].

2.2 Different strategies for web service QoS prediction

Composite service is composed by a set of atomic services. As we know, even if different web services have similar functionalities, their nonfunctional properties (QoS) may vary. For proper selection of relevant atomic web services for a composite service its QoS value need to be predicted, if the value of Qos is greater, then the service will be reliable [21]. Another popular method based on collaborative filtering is Matrix Factorization (MF) which is recently used for web service reliability prediction also [22]. It makes use of user's location and network map for prediction. Network map is used for measuring distance between users in the network. Missing QoS values can be predicted by building an ensemble of non-negative latent factor (NLF) models, it helps to generate unknown QoS data values with high accuracy from previously invoked history of web services [23]. Usually, QoS values of unknown services are predicted by using the known QoS values of existing users. Sometimes they may lead to inaccurate results if the QoS values are taken from unreliable users [24], [25]. To address this issue, the reputation of QoS value contributing users is first calculated and is then used in the Reputation-based Matrix Factorization (RMF) [26].

To predict the availability of atomic web services, a model named as LUCS (service Load, User location, service Class, Service location) is introduced [27]. The process starts with the classification of the collected past invocation data, based on the users' and services' geographic locations, the load of the service provider at the actual time of the invocation, and the computational requirements of the invoked service. Request classification is done based on previously classified groups when a new ongoing service request is received. Similarity measure of that invocation with other previously observed invocations is carried out, which subsequently determines the most similar set of entities. Based on the results obtained, the estimated availability of a service invocation is calculated by considering impacts of the four LUCS parameters.

A reliability prediction model named CLUS is used to estimates the reliability for an ongoing service invocation based on the data assembled from previous invocations [28]. The reliability prediction process is carried out in two phases: a data clustering phase and a prediction phase. Clustering of the history invocation sample is performed before prediction, K-Means clustering algorithm is used for the process. Based on the environment conditions, the time windows in a day are clustered based on the reliability performance from the past invocation samples. Similarly, users and services are also clustered considering their reliability performance within each time window cluster. Finally, a three-dimensional space D containing clustered data is created. After completing the clustering phase, prediction of the atomic services reliability can be performed in the prediction phase using linear regression algorithm. To reduce the scalability issues present in the state-of-the-art approaches, the past invocation data are aggregated using K-means clustering algorithm. This model produces more scalable and accurate predictions compared to previous existing methods.CLUS model addresses various disadvantages in collaborative filtering based approaches, i.e. accuracy and scalability issues. LUCS model is more applicable only when the input parameters are highly available. Whereas, in order to increase the accuracy of CLUS prediction model, K-Means algorithm used in CLUS is replaced by K-Strings algorithm in our proposed work [28].

III. OVERVIEW OF CLUS AND K-STRINGS

Here an overview of CLUS with K-String model for atomic web services reliability prediction is introduced. To make a prediction model more accurate, required parameters should be selected wisely. In CLUS model, user, service and environment specific parameters are chosen which helps to precisely determine service invocation context more effective than other prediction models. To rectify the scalability issue in existing model the collected invocation samples are grouped into three different dimensions associated to three different parameters (User, Service, Environment) using the K-String clustering algorithm.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

K-string clustering algorithm is evolved from K-means algorithm's idea. An overview of the prediction model is shown in Figure 1. It is highly prominent for the clustering of high dimensional data rather than using K-Means algorithm. This aims to generate higher accuracy and efficiency to the prediction model.

3.1 Parameters in CLUS

Three groups of parameters are distinguished in CLUS model [29]. They are user, service and environment specific parameters.

3.1.1 Environment-specific Properties

It specifies certain environment-specific parameters related to the current environmental conditions such as network performance, service provider load at the time of an invocation. Due to practical limitations, service load is only considered as environment parameter. Service load can be defined as the number of requests in a second. User-observed values for QoS properties vary widely for different user's influenced by heterogeneous user environments or unpredictable Internet connections. Fluctuations in service load significantly influences QoS factors such as availability or reliability. During the course of a day considerable load variation may occur, so the day is divided into arbitrary number of time windows assuming the load to be constant for each time window and the past invoked samples are dispersed over them [30]- [33]. It helps to improve prediction accuracy. These time windows are clustered based on the reliability performance based on past invocations using K-Means clustering Algorithm.

3.1.2 User-specific Properties

User-specific properties include a various factors such as user's location, network usage, device capabilities and usage profiles that might impact the reliability of a service. In order to incorporate user-specific parameters into the prediction model, users are clustered based on the reliability performance based on past invocations using K-String clustering Algorithm.

3.1.3 Service-specific Properties

Service-specific parameters represent the impact of service characteristics on the reliability performance. Factors such as service's location, service's computational complexity and system resources, i.e. CPU, RAM, disk and I/O operations may include. Here, only service's location is considered as service-specific parameters for the prediction process. Finally, services are clustered based on the reliability performance based on past invocations using K-Strings clustering Algorithm.

3.2 K-Strings Algorithm

K-strings clustering algorithm is evolved from K-means algorithm's idea [34]. It replaces only centroids of a cluster, which is generated by K-means by a backbone i.e. a set of centroids. It helps to increase the quality of that cluster. K-string algorithm includes 3 main stages, initially forming groups of attributes and generating initial clusters, then updating the set of centroids of clusters and finally output k-strings clusters. The basic idea is, instead of treating the center point of a cluster as the unique center point of a circle, k-strings treats it as a medial axis of a shape. So they may be capable to offer clusters having high quality.

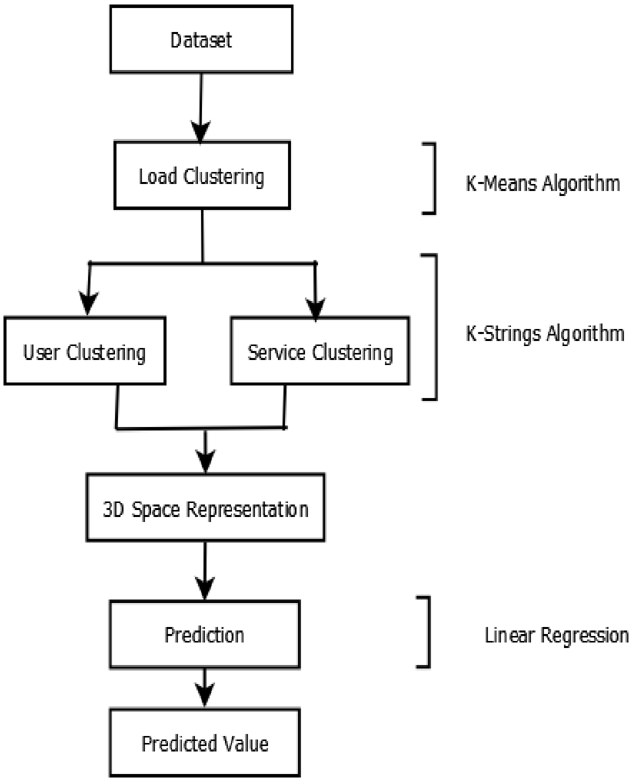

Figure 1: Reliability Prediction Process

IV. CLUS WITH K-STRINGS PREDICTION MODEL

Steps in data clustering process of CLUS defines how prediction is performed from clustered data. A service invocation can be defined as: R (u,s,t). *u* is the user who is executing the service, *s* is the invoked service and *t* is the time of invocation. Past invocation sample contains the above addressed parameters.

To make reliability predictions more scalable and accurate for future invocations data needs to be transformed into more structured and compact form. For that data are stored into a three dimensional space D [u,s,e] where *u*, *s* and *e* each group of parameters. Based on these clustered data further prediction is performed. Figure 1. shows the overall process of reliability prediction. The dataset contains average reliability values collected from past invocation.

4.1 Clustering Phase

To correlate available history invocation record for the prediction process clustering is the initial step that is to be carried out.

4.1.1 Environment Clustering

The dataset contains several average reliability values based on all the three different parameters mentioned above. For prediction process, we need to cluster the available dataset for each distinct parameter.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In environment specific data clustering n number of distinct environmental conditions (E) are specified based on different loads, E = $\{e_1,e_2,...,e_i,...,e_n\}$, where e_i specifies the environmental condition based on the service provider load. After determining the time window as stated above, average reliability $\overline{p_{w_i}}$ value for each time w_i need to be calculated.

$$\overline{p_{w_i}} = \frac{1}{|W_i|}\sum p_r$$

where W_i is the collection of records for the time window w_i, r is a record from the invocation samples and p_r is the user perceived reliability for that invocation [29]. When K number of environmental conditions exists, we need to partition the data points into K-different clusters, each cluster representing each environmental condition. For this K-Means algorithm is used. In this work our dataset contains seven different environmental conditions representing seven different loads.

4.1.2 User Clustering

In User-specific data clustering several user groups have to be defined. Each user group u_k contains users having similar reliability performance. For each user's in a user group a n-dimensional reliability vector need to be calculated which contains average reliability value attained by the given user during the environment condition e_i. After calculating the reliability vector and assigning them to individual users K-strings clustering algorithm is performed [34]. It clusters users into different user groups according to the similarity in reliability value. This helps to easily correlate existing invocation samples to an appropriate user group for prediction.

4.1.3 Service Clustering

Service clustering process is similar as that of user clustering. Here, several service groups s_j have to be defined which contain services having similar or same reliability performance. Each individual service s contains a n-dimensional reliability vector need to be calculated. It contains average reliability of a service s invoked during an environmental condition e_i. After determining the n-dimensional vector, services are clustered into several service groups using K-strings clustering algorithm based on their reliability vector values. Now each available previous invocation records can be easily correlated to the appropriate service group.

4.2 Space D representation and Prediction

3D space representation is done in order to rectify the scalability issue in existing approaches. It represents reliability value into a compact form along with considering our three different parameters. After completing the clustering phase each history invocation record r(u,s,t) can be associated with the corresponding data clusters u_k, s_j and e_i. Then space D can be generated by:

$$D[u_k, s_j, e_i] = \frac{1}{|R|}\sum_{r\in R} p_r$$

Here p_r represents the user perceived reliability for an invocation r and R is set,

$$R = \{r(u.s,t) | r \in u_k \wedge r \in s_j \wedge r \in e_i\}.$$

This is how the space D is calculated [29].

Suppose, if we have to predict the average reliability p_c of an ongoing web service invocation $r_c(u_c, s_c, t_c)$. Among all the environment conditions clusters generated in environment specific clustering process, average reliability of all closest environment conditions cluster is calculated. It helps to map them to the corresponding load condition in the environment. Once they are associated with the actual load conditions in the environment based on the respective environment conditions cluster w_i, we need to check whether there is a set H in the past invocation sample which contains records with the same invocation context parameters of ongoing service r_c.

$$H = \{r_h | u_h = u_c \wedge s_h = s_c \wedge t_h , t_c \in w_i\}$$

If H is non empty the reliability value p_c is calculated using the existing reliability values in the set H:

$$p_c = \frac{1}{|H|}\sum_{r\in H} p_r$$

Else, if set H is empty, calculate the reliability pc using the data stored in the space D as, $p_c = D[u_k,s_j,e_i]$. Space D should be updated when each time past invocation sample gets changed. In this way reliability value is estimated for prediction.

V. RESULTS AND DISSCUSSIONS

In this section, experiments are carried out to compare CLUS model using K-Means and CLUS model using K-String algorithms. Experiments are conducted on real data-set containing average reliability values of different services which were deployed in seven different geographic locations worldwide. It contains service specific parameter by placing 49 web services in seven available Amazon EC regions: Ireland, Virginia, Oregon, California, Singapore, Japan and Brazil, having one service class in each region. User-specific parameters are included, by placing the amount of 50 instances for simulating users in different locations within the cloud. Environment-specific parameters were generated by creating test cases with different load generators defined by the time interval between subsequent invocations. Reliability Values in our real data-set were actually collected under these specific criteria.

Algorithms are implemented using java programming and the experiments are conducted on core i3 PC running on windows 10. Experimental results show that the proposed method is more accurate and efficient than the old method. Accuracy is calculated by measuring the Root Mean Square Error (RMSE).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

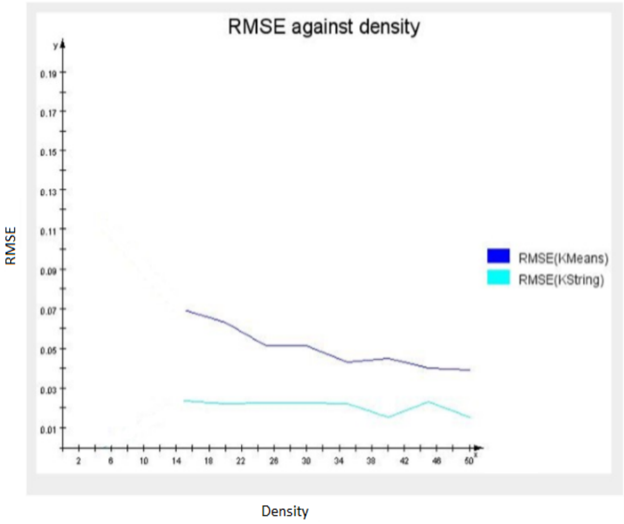

Figure 2. RMSE against Density

In our experiment the accuracy of both methods in different training percentages is conducted. Training percentage means data density; it is altered between 5% and 50% of the whole data-set. In our experiment, firstly we include an amount of 5% of the collected data for training and reliability for the remaining data is calculated. Based on the reliability values that measured in the experiments we calculate the RMSE values. In the next step, we randomly included another 5% of the collected data into training data and recalculated the predictions and performance measures. This process is repeated until the calculation is done for the density of 50%. Thus the impact of the collected data density on prediction accuracy is analysed and is shown above in Figure 2.

VI. CONCLUSION

As web services are platform independent and widely used in many industrial purposes, number of web service users is increasing rapidly. So that, to build high quality service-oriented system, predicting its reliability is very important. It is a predominant QoS factor. This work analyses various QoS value prediction techniques of web services for assessing its quality and to ensure its reliability. Our model estimates the reliability for an ongoing service invocation based on the data assembled from previous invocations by in-cooperating user, service and environment–specific parameters of the invocation context. Aggregation of the past invocation data are done using K-Strings clustering algorithm which is a very good algorithm for clustering high dimensional data. Here a comparison of CLUS using K-Means clustering algorithm and CLUS using K-String clustering algorithm is carried out. The evaluation results confirm that CLUS using K-Strings model produces more accurate predictions when compared to using K-Means algorithm. It is estimated by calculating Room Mean Square Error (RMSE).

ACKNOWLEDGEMENT

We would like to thank Mr. Marin Silic for his support to complete this work, also for sharing original dataset to conduct experiments.

REFERENCES

1. M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on. IEEE, 2003, pp. 3–12.
2. Hongbing Wang, Haixia Sun and Qi Yu, "Reliable Service Composition via Automatic QoS Prediction", IEEE 2013.
3. L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," Software Engineering, IEEE Transactions on, 2004.
4. D. Wang and S. T. KISHOR, "Modeling user-perceived reliability based on user behavior graphs," International Journal of Reliability, Quality and Safety Engineering, 2009.
5. V. Cortellessa and V. Grassi, "Reliability modeling and analysis of service-oriented architectures," pp. 339–362
6. Z. Zheng and M. R. Lyu, "Collaborative reliability prediction of service-oriented systems," in ACM/IEEE International Conference on Software Engineering - Volume 1, ACM, 2010.
7. Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," IEEE Transactions on Services Computing, 2011.
8. L. Baresi and S. Guinea, "Event-based multi-level service monitoring," in ICWS, pp. 83–90, 2013.
9. Marin Silic, Goran Delac, and Sinisa Srbljic, "Prediction of Atomic Web Services Reliability for QoS-aware Recommendation", IEEE 2014.
10. G. Delac, M. Silic, and S. Srbljic, "A reliability improvement method for soa-based applications," Dependable and Secure Computing, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2014.
11. V.GrassiandS.Patella,"Reliability prediction for service-oriented computing environments," IEEE Internet Computing, 2006.
12. W. T. Tsai, D. Zhang, Y. Chen, H. Huang, R. Paul, and N. Liao, "A software reliability model for web services," in International Conference on Software Engineering and Applications, 2004.
13. J. Ma and H.-p. Chen, "A reliability evaluation framework on composite web service," in IEEE International Symposium on Service-Oriented System Engineering, IEEE Computer Society, 2008.
14. B. Li, X. Fan, Y. Zhou, and Z. Su, "Evaluating the reliability of web services based on bpel code structure analysis and runtime information capture," in Asia Pacific Software Engineering Conference 2010, IEEE Computer Society, 2010.
15. X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Adv. in Artif. Intell., vol. 2009.
16. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, ''Item-Based Collaborative Filtering Recommendation Algorithms,'' in Proc. 10th Int'l Conf. World Wide Web, 2001, pp. 285/295.
17. Z. Zheng and M.R. Lyu, ''Collaborative Reliability Prediction of Service-Oriented Systems,'' in Proc. 32nd ACM/IEEE Int'l Conf. Softw. Eng., New York, NY, USA, 2010, vol. 1, pp. 35/44, ACM.
18. A. Klein, F. Ishikawa, and S. Honiden, ''Towards NetworkAware Service Composition in the Cloud,'' inProc. 21st Int'l Conf. World Wide Web, 2012, pp. 959/968.
19. N.B. Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas, and V.Issarny,''QoS-AwareServiceCompositioninDynamicService Oriented Environments,'' in Proc. 10th ACM/IFIP/USENIX Int'l Conf. Middleware, 2009, pp. 123/142.
20. Y. Wang, W.M. Lively, and D.B. Simmons, ''Web Software Traffic Characteristics and Failure Prediction Model Selection,'' J. Comput. Methods Sci. Eng., vol. 9, no. 1, pp. 23-33, Apr 2009.
21. "Service Selection for Web Services with Probabilistic QoS" , IEEE transactions on services computing, vol. 8, no. 3, may/june 2015.
22. Mingdong Tang et al., "Collaborative Web Service Quality Prediction via Exploiting Matrix Factorization and Network Map", IEEE 2015.
23. Xin Luo et al., "Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models", IEEE 2015.
24. Y. Xu, J. Yin, and W. Lo, "A unified framework of QoS-based web service recommendation with neighborhood-extended matrix factorization," in Proc. 6th IEEE Int. Conf. Service Oriented Computing and Applications (SOCA 2013), 2013, pp. 198–205.
25. X.Luo,Y.-N.Xia,andQ.-S. Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization," Knowl. Based Syst., vol. 27, pp. 271–280, 2012.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

26. Jianlong Xu, Zibin Zhen, and Michael R. Lyu, "Web Service Personalized Quality of Service Prediction via Reputation-Based Matrix Factorization", IEEE transactions on reliability, vol. 65, no.1, march 2016.
27. Xin Luo et al., "Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models", IEEE 2015.
28. R.XuandI.Wunsch,D.,"Survey of clustering algorithms", Neural Networks, IEEE Transactions on, 2005.
29. Marin Silic, Goran Delac, and Sinisa Srbljic, "Prediction of Atomic Web Services Reliability for QoS-aware Recommendation", IEEE, 2015.
30. Y. Wang, W. M. Lively, and D. B. Simmons, "Web software traffic characteristics and failure prediction model selection," J. Comp. Methods in Sci. and Eng., 2009.
31. Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana, "Predictability of web-server traffic congestion," Web Content Caching and Distribution, International Workshop on, 2005.
32. M. Andreolini and S. Casolari, "Load prediction models in web based systems," in International conference on Performance evaluation methodolgies and tools, ACM, 2006.
33. Y.-T. Lee and K.-T. Chen, "Is server consolidation beneficial to mmorpg? a case study of world of warcraft," in Cloud Computing, 2010 IEEE 3rd International Conference on, pp. 435–442, 2010.
34. Viet-Hoang Le and Sung-Ryul Kim, "K-strings algorithm, a new approach based on Kmeans", ACM 2015.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)