International Journal of Intelligent Information Systems
2015; 4(1): 8-15
Published online January 23, 2015 (http://www.sciencepublishinggroup.com/j/ijiis)
doi: 10.11648/j.ijiis.20150401.12
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

Research on service reputation evaluation method based on cloud model

Tingwei Chen, Jing Lei

College of Information, Liaoning University, Shenyang, Liaoning, China

Email address:
twchen@lnu.edu.cn (Tingwei Chen), 478444577@qq.com (Jing Lei)

To cite this article:
Tingwei Chen, Jing Lei. Research on Service Reputation Evaluation Method Based on Cloud Model. *International Journal of Intelligent Information Systems.* Vol. 4, No. 1, 2015, pp. 8-15. doi: 10.11648/j.ijiis.20150401.12

Abstract: With the increasingly evident advantages of service-oriented software architecture, Web service received widespread attention, and the numbers of Web Services are increasing constantly. It is more difficult to select high-quality Web service that meets user requirements. Because traditional service reputation evaluation approaches cannot ensure the authenticity and reliability of user ratings, this paper proposes a cloud-based reputation evaluation approach for assessing the history behavior of service consumers, and also takes into account the rating similarity to generate rating quality cloud. With the parameters of cloud model, we can measure the quality level and stability of rankings, which provide additional evidence for trust decision-making. The result of simulating experiments shows that the proposed approach can improve the accuracy of reputation evaluation and the quality of Web service selection, and defend against malicious attacks, so that the interests of service requesters and service providers can be protected.

Keywords: Web Services, Trust Model, Cloud Theory, Recommendation Trust

1. Introduction

Service Oriented Architecture (SOA) provides business with a competitive environment, for enterprise can shorten product cycles, save development costs and enhance enterprise competitiveness using SOA-based applications. And service consumers are even not concerned with how these services will execute their requests. Figure 1 shows the collaboration among the entities in a service-oriented architecture. The collaborations in SOA follow the “publish, find, bind and invoke” paradigm.

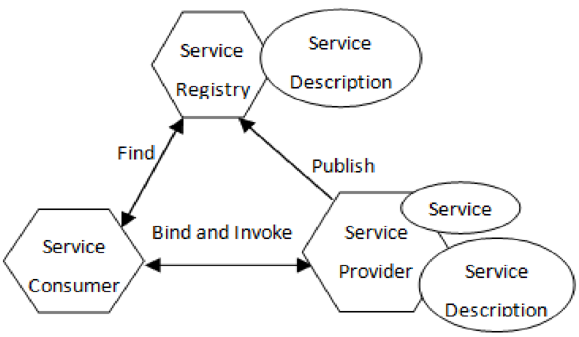


***Figure 1.** SOA's Publish-Find-Bind-Invoke Paradigm*

Web services as a SOA implementation has begun to caught great attention of industry and academia. With the expansion and increasing number of Web services, it is inevitable that there will be a lot of the same or similar functions Web services. As a result, how to choose a high-quality Web services to help customers meet their requirements has become a hot issue in the field of service-oriented architecture.

Currently, there has been an extensive amount of research focused on reputation evaluation and computation, and many scholars have made a wide range of analysis and some achievements. In respect of calculation methods, there are weighted average method, trust model based method, fuzzy theoretical based method, QoS-aware method and so on. Meanwhile, there are distributed statistical model and centralized statistical model in respect of the statistical architecture of reputation evaluation [1-5].

Fu Xiao-Dong [2] designed a QoS-aware reputation mechanism to measure service reputation using the similarity of service QoS true value and declaration. However it only takes service QoS evaluation into account, ignoring customer experience and subjective feelings; Hien Trang Nguyen[3] designed a trust and reputation evaluation system based on Bayesian network. The system evaluated direct and indirect trust relationships among agents by analysis the past behavior of service. However it didn’t take users’ fake ratings into account, and the result deviates from the true value of the reputation evaluation; Sun Qiu-Jing [4] proposed a recommended trust computation approach based on similarity of users' ratings, which can reflect differences of assessing standards for the same service between different users, ignoring the uncertainty and ambiguity of some qualitative concepts.

Therefore, considering the fake ratings in network, it is necessary to measure the quality of customer ratings and its uncertainty during the evaluation of service reputation. Cloud model is an appropriate tool to describe uncertain concepts and transform quantitative expressions into qualitative concepts. The trust decision evidence will be more sufficient because Cloud model can give description and characteristics of a concept from multiple angles more than just one numerical value. With these research problems in mind, we develop a Cloud-based Reputation Evaluation Approach (CREA), which could be applied to Web service selection. The results of simulation-based experiments show that CREA can provide a high success rate of service transaction and resist fake ratings of malicious attacks.

2. Related Works

2.1. Trust Mechanism

"Trust" is a similar concept to another concept "reputation", and they are often confused because they are very close and share some common ground diversity and dynamics. There is no uniform standard of definition of the concept of "trust", since people have different views about trust and reputation. In this paper, we make the integration of other researches and give the following definition of trust, reputation and referential concepts [5]:

Definition 1: Trust is a subjective evaluation based on customers' experience. For instance, trust is a prediction of customer A according to his/her knowledge and network environment， which reflect the credibility of service B for its ability to perform a specific action or provide some function.

Definition 2: Reputation is a quality of service evaluation of the customer, which reflects the performance of service for fulfilling its quality standard declared and trust level evaluation of the service from other customers in network.

Customers' trust in service comes from two aspects, one is direct trust in service based on customers' using experience, and another is indirect trust based on recommend from other customers. Followings are definitions of direct trust and indirect trust in Beth model:

Definition 3: If customer A's trust in service B is got from direct experience, then the trust is called direct trust denoted as DT.

Definition 4: If customer A's trust in service B is got from other customers' recommend or rating, the trust is called indirect trust expressed as IT.

For a service never used before, a customer's trust and judgment most depend on indirect trust and reputation; when the frequency of use increase to some extent, customer's trust will depend on indirect trust from most part. Therefore, other customers' rating and reputation are very important for strange services.

Customer ratings play an important role in reputation calculation. However, there are many abnormal customers in network. They are driven by business benefit, for benefit-connection service, they give an extreme high rating; for competitive-connection service, they give an extreme low rating, as a result, reputation evaluation will be affected. One way to address this problem is to develop evaluation methods. In this paper, we develop a Cloud-based reputation evaluation mechanism for establishing rating quality cloud of customers that can eliminate or punish users' rating if its rating quality is under the benchmark. At the same time, the parameters from rating quality cloud reflect stability of user rating quality, so service reputation evaluation result will be more accurate.

2.2. Cloud Model

LI Deyi proposed the concept of cloud model [6] based on fuzzy theory and probability theory, focused on studying fuzziness and uncertainty of concepts. Cloud model not only can convert qualitative concepts into many quantitative values with certain distribution pattern and characteristic, but also can pick up significant information of qualitative concepts from quantitative value expressions.

Definition 5: Supposing U is quantitative discourse domain expressed by value, C is a concept in U. If quantitative value $x \in U$ and x is a random implementation of concept C, then x's certainty of C, $\mu(x) \in [0,1]$ is a random number $\mu(x) \in [0,1]$, $\forall x \in U, x \to \mu(x)$. The distribution of x in discourse domain U could be called cloud expressed as *C(X)*, and every x is called a cloud drop.

Expected value "Ex", entropy "En" and hyper entropy "He" are three most important digital characteristics of cloud model, which show major point of qualitative concept. In general, "Ex" reflects the center of cloud, which is the expected value of cloud drops in discourse domain, moreover, it is the most representative sample point; "En" reflects the dispersion of cloud drops, which is the measurement of uncertainty and fuzziness of qualitative concepts. For the graphic of cloud, "En" refer to the width of cloud; "He" is the measurement of uncertainty and fuzziness of entropy "En". In addition, the larger Cloud's width is, the bigger "He" is.

Cloud generator refer to generation algorithm for cloud model, including forward cloud generator, backward cloud generator, X conditions cloud generator, Y conditions cloud generator and so on. The most significant are forward and backward cloud generator. Forward cloud generator can be used for converting qualitative concept into quantitative value expression, and backward cloud generator can be used for building cloud to describe the characteristics of qualitative concept. The cloud model can be divided into several categories depending on its shape. There are normal distribution cloud, triangle cloud，echelon cloud，Γ cloud and so on. Normal cloud is the most important cloud model, because normal cloud can be applied to most fuzzy concepts in realistic environment. Li Xiong[11] stated the universality of normal cloud, and all of the cloud generators used in this paper are backward normal cloud generator. Following are its algorithm description:

Algorithm 1. Backward normal cloud generator

Input: data sample xi, i=1,2,3…n.

Output: digital characteristics of qualitative concept (Ex, En, He)

Calculate mean of data samples $\bar{x} \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i$ and sample variance $S^2 \leftarrow \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$

Expected value $Ex \leftarrow \bar{x}$

Entropy $En \leftarrow \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n}\left|x_i - Ex\right|$

Hyper entropy $He \leftarrow \sqrt{S^2 - En^2}$

3. Cloud-based Reputation Evaluation Approach

3.1. Quantization of User Ratings

There are two different kinds of user ratings: explicit rating and implicit rating. Explicit rating comes from user feedbacks, which can be used after simple quantization. However, the major consumers of service are computers, and it is unrealistic to force clients submit their feedbacks every time when they used services. Therefore, this paper take only implicit ratings [7] into account, using Delphi[8] method to build rating index system and FAHP[9] to confirm weights of indexes from every level. Finally, user rating value will be got by collecting and statistics of bottom rating index.

Define user rating as vector $\vec{V}_{i,j}''' = (V_1''', V_2''', V_3''', ...V_n''')$, and V_n''' denotes customer ID_i's feedback value on n.th bottom index during his/her use of service j. following is the final expression of user rating :

$$EV_{i,j} = \sum_{s=1}^{m} \omega_s'' \left(\sum_{k=1}^{n} \omega_k''' \times V_k'''\right) \tag{1}$$

Table 1. *Implicit rating index system*

Stair index	Secondary index	weight	Bottom index	weight
Implicit rating index	"Mark" behaviour	0.36	Adding to bookmark	0.21
			Login or entry	0.15
	"Conserve" behaviour	0.28	Save the page	0.14
			Open new page	0.10
			printing	0.04
	"Repeat" behaviour	0.25	Clicking the cursor	0.09
			Dragging the scroll bar	0.08
			Copying and pasting	0.08
	"Quote" behaviour	0.11	Opening a hyper link	0.70
			Copying a hyper link	0.40

3.2. Rating Quality Cloud Generation

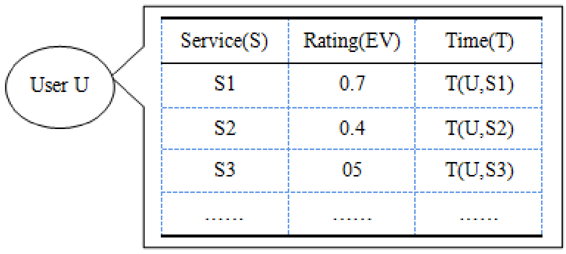
Customers' feedbacks are curial, since service reputation relies on the comprehensive perspective and subjective feeling of customers. However, ratings in network are in chaos. If cannot distinguish noise and malicious feedback from all user feedback, the result of reputation evaluation will be inaccurate. It is necessary to measure the trustworthiness of every customer's rating. In another word, measurement of rating quality is needed.

Definition 5: Rating quality is the evaluation and prediction of customer's rating ability through assessing and quantifying history feedback behavior of customers. In this paper, rating quality denoted by ω_i, the larger ω_i is, the higher the user's rating quality could be, and his/her feedback will be more accurate and objective for showing the real performance of services.

The assessment of rating quality consists of two parts. The first part is customer feedback similarity for each customer. We designed a customer feedback similarity algorithm for assessing each customer's history rating set to quantify customer's preference and rating ability. The customer's feedback similarity shows whether his/her rating accord with normal judgment standard. Malicious customer and customer with deficient rating ability have lower feedback similarity. Consequently, if a customer's feedback similarity is below benchmark, his/her rating should be eliminated for reputation evaluation. Following is definition of feedback similarity among customers:

Definition 6: Customer feedback similarity is the result of assessing customer's rating difference with other customers who come from the same user set by evaluating customer's history ratings, which reflects customers' rating quality from one aspect.

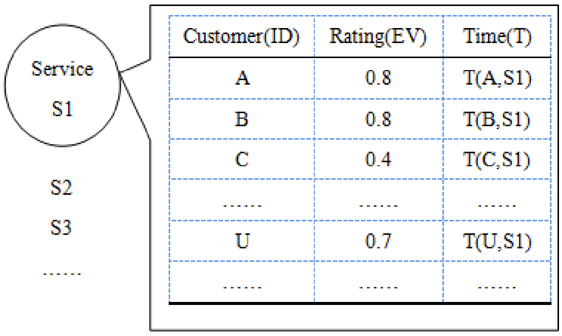
The process to calculate customer's feedback similarity: First, we need to collect and search service set that ever used by the given customer, such as Figure 2; Second, search user set and their rating information of each service as Figure 3, and compare the ratings with given customer's rating for feedback similarity calculation. The mentioned user set could be original user set in early days, when there are enough feedback and transactions, it could be user set clustered by customers' preference and interest [10], and the feedback similarity will have more worthiness.



Service(S)	Rating(EV)	Time(T)
S1	0.7	T(U,S1)
S2	0.4	T(U,S2)
S3	05	T(U,S3)
......	......	......

Figure 2. *Given customer U's service rating set*

Because of large amount of malicious customers in Web service environment, some of them tend to disguise their malicious behavior. For example, if an abnormal customer provide fake feedback to benefit-correlated services, and provide normal feedback to other uncorrelated services. As a result, their feedback similarity will keep at a high level for disguising their attack behavior. This behavior will lead to a more rating fluctuation than other normal customers. So the second part of assessment of rating quality is rating stability, is defined in definition 7. For customers with lower rating stability than benchmark, rating quality will be decrease, it means that they have attack or malicious tendency, and their feedback will be eliminated or punished for reputation evaluation.



Customer(ID)	Rating(EV)	Time(T)
A	0.8	T(A,S1)
B	0.8	T(B,S1)
C	0.4	T(C,S1)
......	......	......
U	0.7	T(U,S1)
......	......	......

Figure 3. *User set and rating information of given service S1*

Definition 7: Customer feedback stability is the concept shows that his/her rating ability won't change in a given time window by assessing customer's history rating information, which reflects customer's rating quality from another aspect besides feedback similarity.

Since cloud model is a good way to measure stability problem, we introduced it in this paper for converting quantitative value expression into qualitative concept with three important parameters, Ex, En and He. We applied cloud model to sketch fuzzy concept, rating quality, and evaluate its feedback similarity and stability. Following is an example for generating rating quality cloud of customer A, the algorithm is given in Algorithm 2.

Algorithm 2 Rating quality cloud generation of customer A

Input: Customer A and his/her history service rating set

Output: Rating quality cloud of A

$$Cld(User_a) = (Ex, En, He)$$

Set(si)←Search Services Set of Customer A in Given Time Window $win = [t_{start}, t_{now}]$

for i←1 to $N_a(s)$ do

Search Customer Set Is of Given Service Si

for j ← 1 to $N_{si}(u)$ do

Search Each Customer's rating for Service Si in User Set Is

dv_s ← Compute Difference of Ratings between Each Customer in User Set Is

item [i,j]←Put Feedback Similarity $1-dv_s$ into Backward normal Cloud Generator

end for

end for

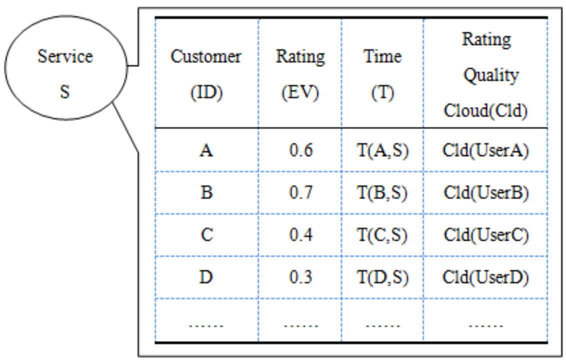
The expected value of customer A's rating quality cloud from algorithm 2 is denoted in (2):

$$Ex_{(a)} = \begin{cases} \dfrac{1}{N_{total}} \sum_{i=1}^{N_a(s)} \sum_{j=1}^{N_{si}(u)} (1-dv_s), & N_{total} \neq 0 \\ 0, & N_{total} = 0 \end{cases} \quad (2)$$

In formula (2), the sum of sample is denoted as N_{total} and $N_{total} = \sum_{i=1}^{N_a(s)} N_{si}(u)$; sum of customers in user set of service Si is denoted as $N_{si}(u)$. Cloud model uses three parameters to describe qualitative concept. Concretely, Ex reflects customer A's rating quality, which means trustworthiness level of his/her rating; Another two parameters En and He, reflects the stability of the rating, when the sum of sample is certain, the discrete level just rely on En and He, so we use $\lambda = \sqrt{En^2 + He^2}$ to measure the stability of rating quality.

3.3. Computation of Service Reputation

We have discussed the method to evaluate customer rating quality, but there remains another question that how to compute service reputation based on customer rating quality. Following is an example of service S to show the process to calculate service reputation.



Customer (ID)	Rating (EV)	Time (T)	Rating Quality Cloud(Cld)
A	0.6	T(A,S)	Cld(UserA)
B	0.7	T(B,S)	Cld(UserB)
C	0.4	T(C,S)	Cld(UserC)
D	0.3	T(D,S)	Cld(UserD)
......	......	......	......

Figure 4. *Customers' Rating Quality Information Set of Service S*

Firstly, collect and search users who ever used service S in a given time window $win = [t_{start}, t_{now}]$, in other words, get the user set who have interacted with S in the past.

Secondly, generate each customer's rating quality cloud through algorithm2, as shown in Figure 4.

For picking up credible and high-quality customers' ratings due to evaluation for performance of Web services, we measure the parameters of rating quality cloud of each customer. On one hand, we set benchmark for parameter Ex for eliminating ratings of customers whose rating quality cloud parameter $Ex \le Ex'$, which ensure that customers' feedback and judgment standard are similar for a given level of service performance, in addition, malicious and inexperienced customers' ratings are reduced.

We set the penalty factor α and fluctuation value limit *Δ* for fluctuation factor $\lambda = \sqrt{En^2 + He^2}$. Customer ratings with low stability will be punished, and their rating quality, denoted as ω_i, will be applied to reputation calculation.

$$\omega_i = \begin{cases} Ex \times \alpha^{\Delta - \lambda} & \lambda > \Delta \\ Ex & otherwise \end{cases} \tag{3}$$

Service reputation calculation relies on customers' ratings and their rating quality. It equals to the weighted average of customer rating, and the weight factor is the revisions of *Ex* based on rating stability.

$$rep(s) = \frac{\sum_{i=1}^{n} (\omega_i \times EV_i)}{\sum_{i=1}^{n} \omega_i} \tag{4}$$

3.4. Service Comprehensive Trust

Customers' trust in services comes from two parts. First part is direct trust DT in service based on interacted experience in the past. Second part is indirect trust IT based on other users' experience and recommend of the given service. Along with the increase of interacted experience, the weight value of direct trust will grow larger. Following is formula(5) of comprehensive trust computation.

$$CT_{a,s} = c \times \mathrm{DT}_{\mathrm{a,s}} + (1-c) \times rep_s \tag{5}$$

In formula(5), c is weight factor of direct trust; rep_s is reputation of service *S*; $DT_{a,s}$ is direct Trust of customer A for service S, and it equals to the average of rating values in a given time window of customer A as shown in formula (6).

$$\mathrm{DT}_{\mathrm{a,s}} = \frac{1}{N_a} \sum_{i=1}^{N_a} ev_{a,s}^{i} \tag{6}$$

Customer's trust in a given service decides whether it will interact with the service. It is important that select service based on service reputation when there is no interacting experience avoiding blindness for service selection, so that transaction success rate can be guarantee.

4. Experimental Results and Analysis

4.1. Experiment Environment

We performed experiments to evaluate the effectiveness, robustness and accuracy of CREA. The experiments simulated by software QualNet, consists of 3 servers, 300 customers and 2000 services as shown in Figure 5. Each server equipped with a processor (Intel ® Xeon ® 3.0GHz), 1GB of memory, 2MB L2 cache, a 250GB SCSI disk, and an Intel Pro1000G NIC. We will test CREA in simulation experiment with different scenarios, and we will be comparing the result and performance of CREA with other approaches [14] to analyze and prove the superiority of CREA.

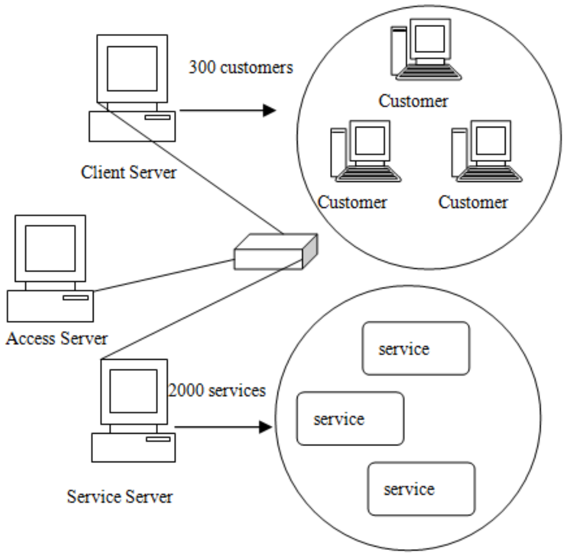


Figure 5. Simulation Environment Implementation

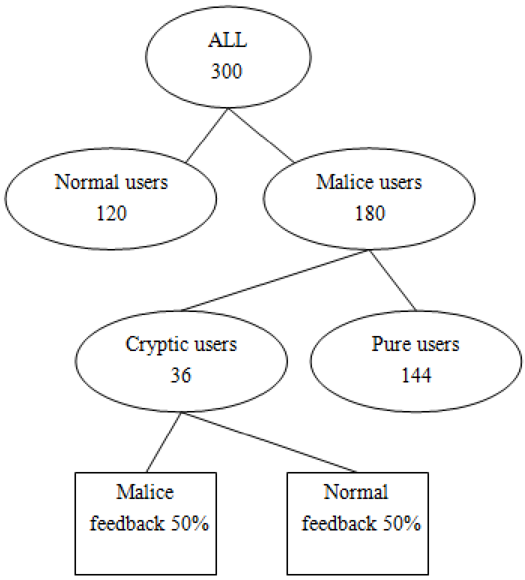


Figure 6. Customer Type Proportion

According to real circumstance in network, we set certain abnormal customers with proportion 0.4, and normal customers with proportion 0.6. There are two different patterns of abnormal customers. One type is pure malicious customers with proportion 0.8, who would always provide malicious ratings. If the service is profit-correlation, they provide extreme high rating to rise up the service reputation called speculation. If the service is competitive-correlation, they provide extreme low rating to debase the service reputation called slander. In contrast, disguise malicious customers provide normal ratings for services with proportion 0.5 to accumulate rating quality for concealing his malicious behavior.

4.2. Resistance against Malicious Customer Attack

In the first scenario, all of malicious customers provide extreme high ratings for profit-correlation services; we compared and analyze reputation change with other approaches in 10 recent time window to prove effectiveness for resisting malicious customer attack. The reference standard is ideal reputation, and it means that all customers provide the same rating for the same service in ideal circumstance. The more close to ideal reputation, the more accurate the reputation evaluation result is.

The results showed on figure 7. We can see that Beth and adaptive Beth[14] didn't test and resist the attack of malicious customers' speculation, service reputation deviate ideal reputation. Beth can't resist the speculation so that reputation will be rise up maliciously and deviate from ideal reputation, because Beth lacks of effective measure for abnormal ratings. Although adaptive Beth has taken actions to resist against fake ratings, it couldn't test the fluctuation of customer behavior. Because adaptive Beth is not available if disguise malicious customers provide fake ratings once in a while after they have accumulated a good rating quality. CREA could measure customers' rating quality and its stability through parameters Ex and λ of cloud model, so the fake ratings from abnormal customers will be eliminated. After 10 time window, the reputation evaluation result is 0.546, still close to ideal reputation.

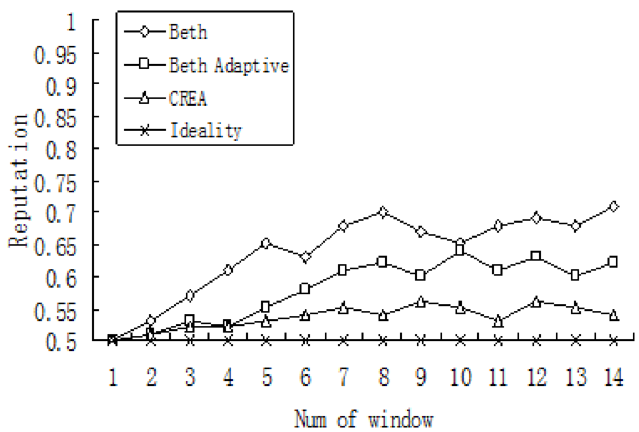


Figure 7. *Resistance of Malicious Customers' speculation*

In the second scenario, abnormal customers will debase competitive related services. Compare reputation change of CREA in 10 time window with other two approaches, as shown in Figure 8.

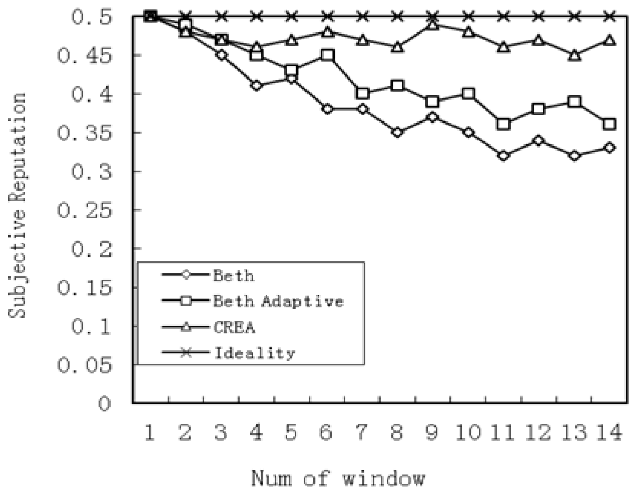
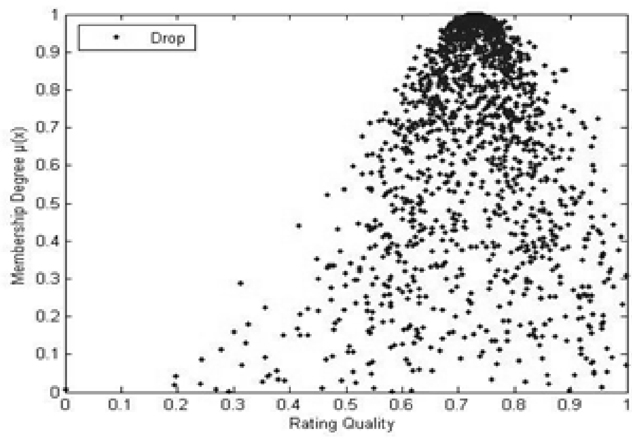
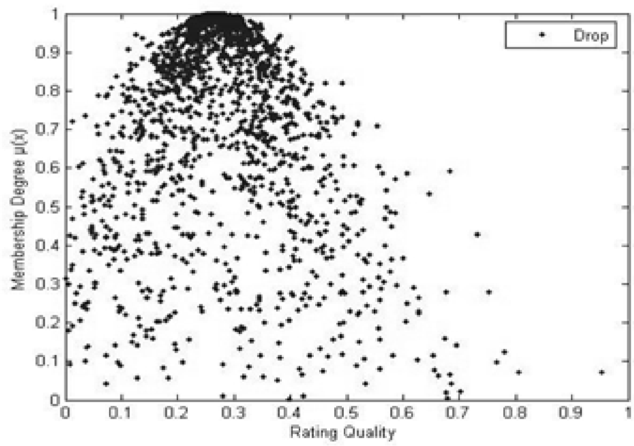


Figure 8. *Resistance of Malicious Customers' Disparagement*

Similar to first scenario, when abnormal customers provide disparaging ratings, Beth and adaptive Beth didn't test and resist these attacks. As a result, service reputation is debased and far away from ideal reputation. However, CREA could keep the reputation evaluation result at 0.472 after 10 time window, which is still close to ideal reputation.

4.3. Resistance Against Malicious Customer Attack





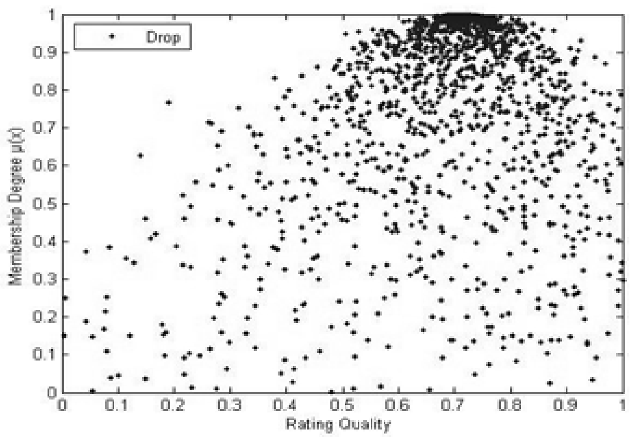


Figure 9. *Different Types of Customers' Rating Quality Cloud Comparison*

Abnormal customers' attack could be test easily, because CREA relies on cloud-based rating quality evaluation model. Following is an example of rating quality cloud comparison of three different types: a normal customer A, a pure malicious customer B and a disguise malicious customer C.

We can see that customer A's rating quality is kept at a high level (Cld_a(0.71,0.08,0.245)), because he/she always provided trustworthy ratings and the stability is good; Pure malicious customer B's rating (Cld_b (0.27,0.11,0.29)) is eliminated for reputation calculation, because B always provided fake ratings and Ex of the rating quality cloud is lower than benchmark Ex'(Ex'=0.64 in this paper). Disguise malicious customer C sometimes provided credible ratings, and sometimes provided fake ratings, so that the Ex of rating quality cloud could be kept at high level Ex=0.76. However, fluctuant behavior of C lead to a low stability and fluctuation factor $\lambda > \Delta(\Delta=0.38)$, so that C' ratings will be punished. Furthermore, customer C's influence for reputation evaluation will be less with low-weighted ratings.

4.4. Transaction Success Rate

In third scenario, we performed an experiment to prove the transaction success rate of CREA and its superiority. Transaction success rate (TSR), service selection success times to total service selection time's ratio, and reflects accuracy of service reputation computation in time window. We compared TSR of CREA with other approaches, and the formula of TSR is shown in (7).

$$TSR = \frac{1}{N}\sum_{i=1}^{N} k_i \times 100\%,\ k_i = \begin{cases} 0,\ if\ \ EV_{i,j} \geq 0.6 \\ 1,\ otherwise \end{cases} \tag{7}$$

TSR is an important index to reflect the safety of reputation evaluation model. In this paper, transaction success means customer's rating value $EV_{i,j}$ is equal to or larger than 0.6.

For this experiment, CREA didn't show superiority at the beginning phase because of data sparseness, and there are not enough customer ratings. So TSR grew at lower speed than Beth and adaptive Beth. However, when time went by and transaction quantity accumulated to 400, CREA began to show its superiority and TSR grew at a stable speed. The other two approaches, by contrast, TSR growth rate decreased, sometimes TSR even fell down or fluctuated when transaction quantity accumulated because of the influence of malicious customers' fake ratings. In comparison, CREA performed a credible and stable service selection, so that TSR could be kept at high level and grew continuous.

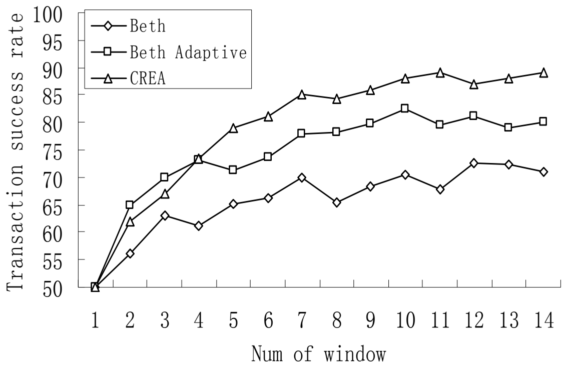


Figure 10. *Transaction Success Rate Comparison*

5. Conclusion

This paper proposes the approach and framework of cloud-based reputation evaluation when malicious customers provide fake ratings. This framework can generate rating quality cloud, and cloud parameters will reflect customer's rating ability and stability, so that malicious and disguise malicious customers could be distinguished from honest customers. The next step will be to study other trust mechanism, and to better improve reputation evaluation performance.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.60903008) and General scientific research project of the Education Department of Liaoning Province (No.L2011004).

References

[1] Min Luo, Mark Endrei, et al. Patterns: Service-Oriented Architecture and Web Services. International Technical Support Organization，Raleigh Center

[2] Fu Xiao-Dong, ZOU Ping, JIANG Ying. Web Service Reputation Measurement Based on Quality of Service Similarity. Computer Integrated Manufacturing Systems, Vol.14, No.3, March 2008, pp.0619-0624.

[3] Hien Trang Nguyen, Weiling Zhao, Jian Yang. A Trust and Reputation Model Based on Bayesian Network for Web Services[C] IEEE International Conference on Web Services,2010, pp.251-258

[4] Sun Qiu-Jing, Zeng Ping-Fan. Trust Model Based on Reputation and Cloud Model in P2P Environment. Journal of Chinese Computer Systems, Vol.31, No.7, July 2010, pp.1328-1332.

[5] Billhardt H, Hermoso R, Ossowski S, Centeno R. Trust-Based Service Provider Selection in Open Environments. In: Proc. of the22nd Annual ACM Symp. on Applied Computing. New York: ACM Press, 2007, pp. 1375-1380.

[6] Li DY, Du Y. Artificial Intelligence with Uncertainty. Beijing: National Defense Industry Press, 2005 (in Chinese).

[7] Claypool M et al. Implicit interest indicators[C]. In: Proceedings of ACM Intelligent User Interfaces Conference (IUI) , Santa Fe, New Mexico, ACM, 2001.

[8] Liu Xue-yi. Delphi Technique in the Assessment of Interdisciplinary Research. JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY, vol. 8, No. 2, 2007.

[9] Li Zhen, Yang Fang-Chun, Su Sen. Fuzzy Multi-Attribute Decision Making-Based Algorithm for Semantic Web Service Composition, Journal of Software, Vol.20, No.3, March 2009.

[10] Sun Ping, Jiang Chang-Jun. Using Service Clustering to Facilitate Process-Oriented Semantic Web Service Discovery. CHINESE JOURNAL OF COMPUTERS, Vol18, No.8, Aug. 2008

[11] Li Xiong, Liu Ling. Peer Trust supporting reputation-based trust for peer-to-peer electronic communities [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(7), pp.843-857.

[12] Li DY, Meng HJ, Shi XM. Membership cloud and membership cloud generator. Journal of Computer Research and Development, 1995, 32(6), pp.16-21 (in Chinese with English abstract).

[13] Wang Shang-Guang, Sun Qi-Bo, Yang Fang-Chang. Reputation Evaluation Approach in Web Service Selection. Journal of Software, 2012, 23(6), pp.1350-1367.

[14] Beth T, Borcherding M, Klein B. Valuation of Trust in OpenNetwork[C]//Proceedings of the European Symposium on Research in Security. Brighton: Springer-Verlag, 1994, pp. 3-18.