RESEARCH ARTICLE OPEN ACCESS

Efficient Method In Web Service Recommendation Systems

G.Sailaja [1], A.Hari Kumar [2]
Dept. of CSE
Visakha Institute of Engineering and Technology
Andhra Pradesh - India

ABSTRACT

Web service recommendation systems can assist service users to locate the right service for the big number of available Web services. Avoiding recommending dishonest or unsatisfactory services is a key research problem in the design of Web service recommendation systems. Reputation of Web services is a widely-used metric that determines whether the renovation should be recommended to a user. The service reputation score is usually counted on using feedback ratings offered by users. Although the reputation measurement of Web service has been probed in the recent literature, existing malicious and subjective user feedback ratings often lead to a bias that leads down the operation of the service recommendation system. In this report, we propose a novel reputation measurement approach for Web service recommendations. We first detect malicious feedback ratings by getting on the Cumulative Sum Control Chart, and then we burn down the effect of subjective user feedback preferences employing the Pearson Correlation Coefficient. Moreover, in parliamentary procedure to defend malicious feedback ratings, we propose a malicious feedback rating prevention scheme employing Bloom filters to enhance the recommendation performance. Extensive experiments are taken by practicing a real feedback rating dataset with 1.5 million Web service invocation records. The experimental results show that our proposed measurement approach can reduce the variance of the reputation measurement and enhance the success ratio of the Web service recommendation.

Keywords:—Web service recommendation, feedback rating, reputation, Cumulative Sum Control Chart, Pearson Correlation Coefficient.

1 .INTRODUCTION

Web service technologies create an environment where users and applications can search and compose services in an automatic and seamless fashion. In the service-oriented environment where everybody is allowed to propose services, it is lifelike that there will be numerous offers of services providing equivalent or similar functionality [1]. Moreover, Web services that span diverse organizations and computing platforms can be framed to create new, value-added service- oriented applications efficiently. However, some Web services may act maliciously. Hence, a key require- ment is to offer an effective mechanism in recom- mending trustworthy services for users.

Web Service recommendation systems can be em- ployed to recommend the optimal Web service for satisfying the user's requirements [2]. Service recommen- dation is helpful for users when two or more Web services deliver the same functionality but different Quality-of-Service (QoS) performance. QoS is defined as a circle of non-operational properties, including rep- utation, response time, dependability, etc. Web service recommendation can provide the user with necessary data to help decide which Web service should be selected [3].

Most QoS-aware Web service recommendation schemes are established along the qualities promised by Sir- vice providers. Still, service providers may fail partially or fully in giving up the promised quality at runtime [4]. It is not an easy job since some service providers may not meet their promised share- vice quality. The reputation of Web service needs to be taken when creating a service option. Web service reputation is involved as a metric of its future behavior. It is a collective measurement of the sentiments of a community of users regarding their actual experience with the Web service [3]. It is calculated as an aggregation of user feedback ratings over a specific period of time (a sample interval) and reflects the reliability, trustworthiness, and credibility of the Web service and its supplier.

With the Web service reputation taken into regard-Aeration, the probability of recommending the optimal service and the success ratio of the composite services can be increased. Nevertheless, as it is not realistic to promise that the user feedback ratings are fairly Ac- curate and non-malicious [5], several surveys have acknowledged the importance of reputation measure- ments of Web services. The proposed solutions [6-10] employ different techniques to measure Web service reputations based on user feedback ratings. Although previous work has explored the efficiency and robusts of several measurement approaches, most of them [6-10] suffer from the weaknesses described as follows.

Foremost, it is hard to guarantee the purity of user feedback ratings because of the existence of male- course users. Malicious users could provide malicious feedback ratings to affect the measurement results for commercial benefit. In open service-oriented environ- ments, at that place are no widely-employed user verification mechanisms. Participating users are usually repaired- sent by a pseudonym. In such environment, a spe- cial threat comes from Sybil attacks [11]. This attack leaves a single malicious user to be epitomized by an arbitrary number of forged users. Hence, malicious users can initiate a flood of malicious feedback ratings to subvert the reputation system of Web services.

Second, previous approaches fail to ensure the AC- curacy of feedback ratings. There are a large variety of users along the Internet. Users have different feedback rating styles [12]. Different users often give different feedback ratings to the same service. For a reputation mechanism to be fair and objective, it is essential to measure reputation on the basis of fair and objective feedback ratings.

Finally, most previous research focused on various feedback rating aggregation schemes of reputation measurement, and little work investigated preventing malicious feedback ratings. If the Web service rec- ommendation system cannot prevent malicious feed- back ratings, any effective reputation measurement approach will become invalid since these malicious feedback ratings suppress benign feedback ratings. Hence, an effective malicious feedback rating pre- Vientiane scheme is really indispensable for the reputation measurement of Web services.

In our previous work [13], we briefly analyze the importance of a reputation measurement in service computing, which lacks of deep research on Repu- tension measurement and malicious feedback rating prevention. To come up to these weaknesses, this report extends our previous work by designing a reputation measurement approach to reduce the variance of the reputation measurement of Web services and to improve the success ratio of the service recommendation. Moreover, to prevent malicious users from sup- pressing benign feedback ratings, this newspaper introduces a malicious feedback rating prevention scheme.

This report makes the contributions:

1) We adopt the Cumulative Sum Control Chart to identify malicious feedback ratings to lessen the influence of malicious feedback ratings on the trusted reputation measurement.

2) We devise feedback similarity computation to shield the different preferences in feedback ratings of users using the Pearson Correlation Coefficient.

3) We propose a malicious feedback rating prevention scheme to prevent malicious users from suppressing benign feedback ratings using a standard Bloom filter.

4) We validate our proposed malicious feedback rating prevention scheme through theoretical analysis, and also evaluate our proposed measurement approach experimentally on a real feedback rating dataset in-evolving 1.5 million real-world Web service invocation records.

The rest of this report is organized as follows. Part 2 introduces related work. Part 3 describes the proposed reputation measurement approach. A malicious feedback rating prevention scheme is pro- posed in Section 4. Part 5 presents the theoretical analysis about the proposed measurement approach. Part 6 conducts experiments to assess the proposed measurement approach and Section 7 concludes the composition.

II. RELATED WORK

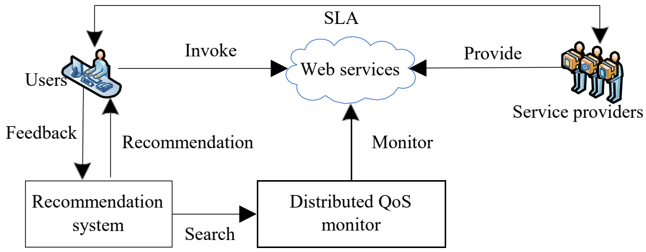
To provide accurate reputation measurement for We- b service recommendation, some notable reputation measurement schemes has been offered.

Conner et al. [7] Proposed a reputation-based trust management framework that holds up the synthesis of faith-related feedback ratings from multiple user- vices that are hosted within an infrastructure. The nitty-gritty of the framework is a trust management service (TMS). TMS allows each service to employ its own trust metrics, to satisfy its local trust requirements, and to support multiple reputation scoring functions. This framework holds a substantial advantage in that it supports, multiple reputation measurement approaches, which are suited to multiple Web service environ- ments. TMS involves the client, the service, the nor- malized transaction feedback rating, and the set of optional attributes to make a service invocation history record. Yet, for malicious feedback ratings, malicious users often collude with other users. Then TMS cannot find malicious feedback ratings. Moreover, TMS calculates the trustworthiness of a given peer as the average feedback weighted by the stacks of the feedback users. Regrettably, a feedback, user with high trustworthiness is not consistently true and it also provides malicious feedback ratings for the illegal acquisition of economic benefits. Hence, TMS cannot get accurate the reputation when good feedback users become bad or bad users become proficient. Lima and Boutaba [10] proposed a feedback can- putation model, derived from the expected discount- formation theory from market science, was used to

Generate a feedback from service utility and price, and then a reputation derivation model had also been proposed to aggregate feedbacks into a reputation value that better reflects the behavior of the service at selection time. Nevertheless, the model cannot shield users' different feedback preferences, which establishes the reputation value biased, and decreases the accuracy. Moreover, it is really hard to predict the feedback ratings in real Web service environments.

Hence, the model cannot obtain the deserved reputation value. Z. Wang and J. Co [14] proposed a two-layer method for evaluating and selecting QoS guaranteed resources from a number of potential Grid resource candidates. On the bottom layer, the informed users contribute their experiences and make fuzzy-based judgments about a resource individually. In the top layer, the approach selects judgments from all representatives and makes a comprehensive decision. The two-layer method is stable and accurate in different grid envi- ronments. R. Zhou and K. Hwang [15] proposed a P2P reputation system called PowerTrust. The Pow- erTrust dynamically selects a small number of power nodes and then by using a look ahead random walk strategy and leveraging the power nodes, the Pow- erTrust improves the global reputation accuracy and aggregation speed. What's more, the Power Trust is robust and scalable in pre joining, peer leaving and malicious peers, which can significantly achieve high query success rate in P2P file-sharing applications. Nevertheless, most P2P systems deployed on the Internet are unstructured. Alas, the schemes [15,16] can not support unstructured P2P system.

S. D. Kamvar et al. [17] showed an efficient method to minimize the impact of malicious peers on the operation of a P2P system. The method computes a global trust value for a peer by Coca- letting the left principal eigenvector of a matrix of normalized local trust values, thus taking into con- sideration the entire system's history with each single match being able to lessen the number of innocent- tic files in the P2P system. J. Caverlee et al. [18] presented the SocialTrust framework that supports tamper-resilient trust establishment in the presence of large-scale manipulation by malicious users, clique formation, and dishonest feedback. By distinguishing relationship quality from a faith, incorporating a per- sonalized feedback mechanism for adapting as the community evolves and tracking user behavior, the Social Trust can significantly confirm the robust trust establishment in online social nets. In contrast to existing systems which suffer from low performance because of malicious feedback ratings and different user



tastes, our approach can mitigate these

Users' feedback preferences on the accuracy of Web service reputation measurement, but existing systems cannot support the substantive case.

III. THE REPUTATION MEASURE

The report represents a collective perception of the users in the community about a Web service, that is, the reputation of a given service is a collective feedback rating of the users that have interacted with or used the service in the yesteryear.

Feedback rating is the perception of each user about invoking services. It could be a single value representation- in an overall perception or a vector representing a value for each QoS attribute of a Web service, such as a response time, reliability, and availability.

Fig. 1 shows what happens when a user commits a service request to the recommendation system. With a Service Level Agreement (SLA) between a user and a avail provider, the user selects a Web service that satisfies his QoS requirements and then invokes the help. Subsequently the service is consumed, the user describes a feedback rating for the service regarding the execution of the Web service. Ultimately, the rec- ommendation system collects the feedback evaluation and other feedback ratings from other users with a Data Collector, calculates the reputation (scores), updates these scores in a QoS repository, and provides the scores when recommending services to the users.

In this work, for the j-the invoked service SJ (j = 1, 2,...), a user provides a feedback rating that India- cuts the level of gratification with the service after each interaction with the overhaul. A feedback rating is simply an integer that ranges from 1 to R (e.g., R=10), where R means extreme satisfaction and 1 means extreme dissatisfaction. Then users maintain n feedback ratings, which represent their perception of s_j's performance. We take q (SJ) to represent the reputation score of SJ over a worldwide time.

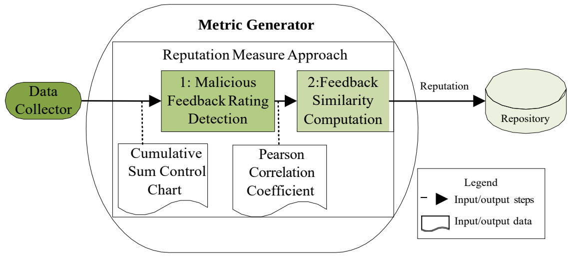


Fig. 2. Procedures of the reputation measurement approach.

Nonetheless, because the reputation influences the rec- ommendation of an interaction partner, some Dishon- East service providers misuse the system. These service providers might have a direct involvement in bettering the opportunities of a certain candidate to become selected or to decrease the prospects of others. Moreover, the feedback rating can be individual because it is based on users' personal expectations and judgments. The different users that invoke the same service may pro- vide varied feedback ratings. Thus, the main call- lenge is addressing services that seek to provide misleading feedback ratings, either unfair or subjec- tive feedback ratings. Hence, a recommendation says- team needs appropriate mechanisms for filtering and weighting services with a reputation metric. Nonetheless, in evidence-based reputation measure approaches, the trust an entity has in some other entity is normally related to a pseudonym that influences the accuracy of the reputation measurement. Moreover, they may neglect to acknowledge the feedback ratings with users' preferences. They do not cater to the accuracy of a reputation measurement, which gets the reputation of a Web service deviate from its genuine value in a paper scheme or an e-commerce application. Hence, to resolve the problem, we propose a reputation measurement approach that is grounded on a feedback rating evaluation for the Web service recommendation.

As shown schematically in Fig. 2, our proposed measurement approach mainly contains two phases, i.e., a malicious feedback rating detection and a feedback rating ad- justment. The first phase involves detecting malicious feedback ratings collected by a Data Collector using the Cumulative Sum Control Chart (called CUSUM). The second phase involves computing the feedback similarity of different users using the Pearson Cor- relation Coefficient to adjust the feedback ratings. Ultimately, the Repository stores the reputation measured scores and provides the scores when requested by the recommendation system. Details of these phases are given in Section 3.1 and Section 3.2, respectively.

a. Stage 1: Malicious Rating Detection

i. Data sampling and the CUSUM

A special threat to the reputation measurement of Web services comes from malicious feedback ratings such As Sybil attacks [19-20]. Hence, malicious feedback ratings must be considered in reputation measurements of Web services.

Under normal situations, each user selects a recom- mended Web service, invokes it with an expected QoS, and ends with a feedback rating. When malicious users attack the reputation system, there are more negative feedback ratings than the usual spot (an instance of the malicious feedback ratings is shown in the appendix). Thus, under abnormal situations, there would be more malicious feedback ratings than benign feedback ratings in a sampling interval. In practical applications, the reputation system of Web services can become invalid with mass malicious feed- back ratings. Therefore, the reputation system is unable to reply to user recommendation requirements effectively. Hence, our objective is to recognize attacks by detecting an imbalance in the feedback rating flow for an anomalous shift in the positive or minus charge.

In this work, in order to more accurately detect the anomalous shift than the above example, we use the Cumulative Sum Method (CUSUM) to discover and handle malicious feedback ratings.s. The CUSUM as a sequential analysis technique is typically applied for monitoring change detection based on hypothesis testing. It is produced for independent and identically distributed random variables. For instance, for a process Yi (I = 1, 2,...), there are two hypotheses, θ0 and θ1, with probability density functions pθ0 (Yi) and pθ1 (Yi). The first hypothesis corresponds to the statistical distribution prior to a change and the search- and hypothesis corresponds to the distribution after a modification. The CUSUM for signaling a change is based along the log-likelihood ratio can, which is afforded by the following:

$$C_n = \sum_{i=1}^{n} c_i \qquad (1)$$

The typical behavior of the log-likelihood ratio con- tains a negative trend before a change and a positive drift after the change (we make an example in Fig. 3). Hence, the relevant information for detecting a change is about the difference between the log-likelihood ratio, C_n (n = 1,2,.), and its current minimum value

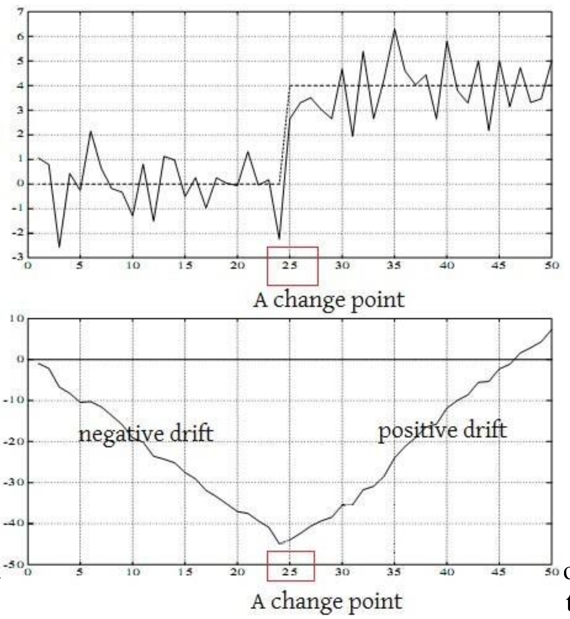


Fig. og-like the number of sampling intervals. The vertical axis in the above number represents the detection, object and The vertical axis in the below figure represents the detection results.

(i.e.,a threshold value), $h(h > 0)$. If $C \quad \geq h$, then a Positive shift occurs in the *on*-the sample, i.e., there is an abnormal detection point. It is well known that

CUSUM is well-suited for controlling the abnormal shift and has been widely applied for detecting the small and moderate mean shifts [21-22]. We can take out the male-course feedback ratings as the abnormal shift according to Eq.3; hence, CUSUM can be used for detecting the malicious feedback ratings. As shown in Table 1, in order to understand our detection mechanism by using CUSUM, we create a table listing parameters and substances for the reader's reference. The detailed detection mechanism is reported in the succeeding segment.

TABLE 1
CUSUM parameters

Parameter	Meaning
n	The number of sample intervals.
y_i	The number of detection objects (such as feed- back ratings) in the i-th sample interval.
θ_0	The parameter of the statistical distribution pr- ior to a change.
θ_1	The parameter of the statistical distribution af- ter a change.
$p\theta_0(y_i)$	The probability density function with the para- meter θ_0.
$p\theta_1(y_i)$	The probability density function with the para- meter θ_1.
C_n	The log-likelihood ratio.
h	The threshold value.

ii. Detection mechanism

This segment concentrates on the application of CUSUM to detect and handle the malicious feedback ratings, including positive malicious feedback ratings (i.e., unfairly high feedback ratings) and negative Mali- course feedback ratings (i.e., unfairly negative feedback ratings). Because a negative malicious feedback rating (negative drifts) detection is similar to a positive malicious feedback rating (positive drifts) detection [22], in this study, we only consider positive malicious feedback rating detection.

For each feedback rating, CUSUM monitors a set of n ($n = 1, 2,...$) feedback rating sample intervals $\{y_1, \cdots, y_n\}$. The variable $y_j (y_j = \sum_{i=1}^{m} r_i)(1 \leq j \leq N)$ ($m = 1,2,...$) is the sum of all of the feedback Ratings (called feedback rating traffic in this paper) in the j-the sample interval (a specific point of time) where m is the number of feedback rating in the sample interval.al. Agreeing to the literature [20,23], assume that the change feedback rating traffic yj is an independent Gaussian distribution with a known variance σ2 (the assumption of a Gaussian distribu- tion about yj can be utilized better by removing no stationary behavior [24]). The premise remains the same after the alteration, and μ0 and μ1 are the average feedback rating traffic before and after the modification.

Then CUSUM can be described as follows:

$$f_n = \left(f_{n-1} + \frac{\mu_1 - \mu_0}{\sigma^2}\left(y_n - \frac{\mu_1 + \mu_0}{2}\right)\right)^+, \qquad (2)$$

where if $f > 0$, $f^+ = f$; otherwise, $f^+ = 0$.

In summation, to reduce the complex and time-consuming calculations, we regard a mere approach to applying CUSUM to x˜n, with the following:

$$\tilde{x}_n = x_n - \bar{\mu}_{n-1} (n = 1, 2, ...), \qquad (3)$$

Where Xian is the total of all of the feedback ratings in the n-the sample interval, and μ¯n is an estimate of the mean rate at the n-the sample interval.

We can obtain $\bar{\mu}_n$ of Eq.3 with an exponential weighted moving average, as follows:

$$\bar{\mu}_n = \lambda \bar{\mu}_{n-1} + (1 - \lambda) x_n, \qquad (4)$$

Where λ [0, 1] (its setting depends on the applica- tion preference) is the exponential weighted moving average (EWMA) factor, i.e, the weight given to the most recent rational subgroup mean.

The average feedback rating traffic of x˜n prior to a change is zero; hence, the mean in Eq.4 is μ0 = 0. A continuing issue that must be addressed is the value of μ1, which is the mean rate after the modification. This value cannot be known in advance and its setting depends on the application preference. Hence, we approximate it with αμ¯n, where α is an amplitude percentage parameter, which equates to the increasing mean rate after a modification.

Then the CUSUM from Eq.3 can be written as follows:

fn=Σfn−1+αμ−n1σ2 n−n−1−αμ n−1+2(5)

If $f_n \geq h$ ($h > 0$ is the predesigned CUSUM threshold malicious attack has been detected and the feedback ratings are dropped.

b. Phase 2: Rating adjustment

Although malicious feedback ratings can be detected using the different preferences of the user with the same service, which fails to ensure the accuracy of the feedback ratings. It is well known that there is a large variety of users on the Internet. These users, who have different preferences, report feedback ratings that are often subject to their preferences. Some users may be conservative, whereas some others may be the influence of conservative, aggressive, or neutral feedback ratings for the same

In our study, feedback similarity computation is proposed to shield the influence of different prefer- ences of users and to adjust their feedback ratings with the Pearson Correlation Coefficient (PCC) [25].

We assume that there are m users and n Web services, and the relationship between users and Web services is denoted with an $m \times n$ matrix [2]. Then each entry r a,i in the matrix denotes the feedback rating of entry $r_{a,i}$ in the matrix denotes the feedback rating of the Web service i rated by the user a where $r_{a,i}$ is from Phase 1 and is a normal feedback rating.

The PCC uses the following equation to compute the similarity between user a and user u based on their commonly-rated Web services.

II users (a larger value indicates a higher similarity), Ia Iu is a set of commonly rated Web services by both users a and u, Ra, I and Ra, I are the two feedback ratings of Web service I rated by user a and u, respectively (Ra, I and Ra, I am from Phase 1), and Ra represents the average feedback rating of all of the Web services that are ranged by user a.a.

After calculating and ranking the PCC similarity values between the current user and the other users, a set of similar users $S(a)$ can be identified, as follows:

$((a) = \{u|Si((a,u) \geq Sim_k, Si((a,u) > 0, a \neq u,$,

Where Simk is the k-the largest PCC value with the cur- rent user u where k presents the number of the similar users (i.e., they have larger PCC values than others), posed reputation measurement approach to raise the carrying out of the recommendation system. The idea is to identify the IP addresses with the offending feedback ratings and filter them away. In ordination to accomplish this, we use a

and $Sim(a, u) > 0$ is to exclude the dissimilar users (dissimilar users, e.g., those with negative PCC value, will influence the reputation measurement accuracy).

After obtaining the set of similar users, according to a set of community Web services $SS^k = s^k, \cdots, s^k$, 1 $l\Sigma$ which contains services used by the K users, we can calculate the feedback similarity between user a and user u.

service

where represents the similarity oftwo users (a larger value indicates a higher similarity) and SS^k is the number of services in SS^k.

Having calculated the feedback similarity, we can use $FS^k(a, u)$ to adjust the feedback ratings of user a according to the feedback ratings of other similar users.

where $\hat{r}_{a,i}$ is the adjusted feedback rating of the i-th rated service from user a, $r_{u,i}$ is the feedback ratings of Web service i rated by user u.

Having executed the two phases mentioned above, to gain the accurate reputation measurement, we transform Eq.1 into Eq.11 by the following equation:

$$q(s_j) = \frac{1}{m}\sum_{a=1}^{m} \hat{r}_{a,i}, \tag{6}$$

where s_j represents the j-th Web service and $q(s_j)$ represents the measured reputation of the service s_j.

IV. Malicious Rating Prevention

In this section, in order to prevent malicious feedback ratings from reaching the QoS repository of service brokers, we propose a malicious feedback rating pre vention scheme. standard Bloom filter to prevent the anomalous feedback ratings The Bloom filter was developed by Burton H. Salad days in the 1970s [26]. It is first "programmed" with each message in the set, and then queried to determine the membership of a particular message, i.e., whether an element is a member of a band. It is a data structure employed for presenting a set of messages succinctly, and is widely used for different purposes of Internet applications. For the convenience of the proofreader,

a. Overview of the Bloom Filter

We start by introducing the mathematics behind a standard Bloom filter. A standard Bloom filter for M bits, initially all set to 0. A standard Bloom filter uses different hash functions h_1, h_2,...,h_k, each of which maps or hashes some set element to one of the m array positions
With a uniform random distribution over the range 1... *m*. For each element, x S, the bits h_j (*x*) are set to 1 for 1 *j* *k*. A position can be set to 1 multiple times, but only the first change has an outcome. To determine if an item y is in S, we check whether all Hg (y) is set to 1. If not, then clearly *y* is not a member of *S*. If all *Hg* (*y*) bits are set to 1, *y* is in S. If all Hg (y) bits are found to be 1 and y is not a member of S, then it is a false positive (The false positive is sufficiently low, and almost can be discounted according to the practical application [26]).

b. Prevention Scheme

The tonality of the prevention is to identify the IP address-Es that are associated with malicious feedback ratings, and then inform the service agent to stop malicious users from rating these Web services. Hence, our proposed prevention scheme contains two levels, i.e., activating stage and drawing a blank level.

In the activating stage, the first measure to implement a Bloom filter is initializing the following parameters: the upper bound on false match probability of the Bloom filter, the filter size m of the Bloom filter, and the number of hash functions k of the Bloom filter.. The second measure is to place a malicious feedback rating IP address set S = mrip1, mrip2,..., mripn with no points. We will first present how a Bloom filter is presented through a serial publication of item insertion operations. Algorithm 1 includes the details regarding the process of the activating prevention operation. It is clear that when malicious feedback ratings are detected in the *me*-the sample interval by using the CUSUM algorithm (Section 3.1.2), the set *S* collects IP addresses of feedback ratings in the sample interval. Because attackers often provide malicious feedback ratings in a short time, we assume that *S* can collect all malicious IPs. The last tone is to use k independent hash functions h1, h2,..., hike to map each item of S to the bit vector 1,..., m uniformly. When inserting *rip*, Algorithm 1 sets the bits at all these positions to 1. Hence, it is convenient to represent *S* as a Bloom filter by invoking Algorithm 1 repeatedly. After achieving the Bloom filter, the blocking stage starts to prevail from the (I + 1) -the sample interval to the on-the sampling interval when I h in the I-the sample interval. It can block malicious feedback ratings by going over IP addresses based on the Bloom filter instead of S. The detailed blocking process is illustrated in Algorithm 2, which uses an item *IP* as input. If all the hash [j] bits are set to 1 for $1 \le j \le k$ in the Bloom filter, then the

Algorithm 1 Activating Prevention

Input: *mrip*, malicious feedback rating IP address elements
Output: An activated Bloom filter

```
Initiate k and m;
Obtain the set S;
for i=1 to n do
    for j=1 to k do
        Vector(hash[j](mrip[i])) 1←
    end for
end for
end if
```

Item *IP* is a member of *S*, i.e., it is with a malicious feedback rating. Otherwise, *IP* is not a member of *S*, i.e., it is with a benign feedback rating. In the freezing stage, we determine the blocking ratio (BR) as the proportion of the number of the IPs with malicious feedback ratings and all IPs in the same sample interval, i.e., BR = θ/n where the better the algorithm is, the larger the blocking ratio is.
After we place the malicious IPs, the remote user- vice broker (RSB) will be responsible for getting out the malicious clients who rated those Web services. Lastly, the certification and authorization module (AAM) of the RSB will block these malicious users. This level is comparatively straightforward and is not the focus of this report. By our proposed prevention.

Algorithm 2 Blocking malicious feedback ratings

```
Input: ip, IP address elements
Output: BR, the blocking ratio θ =
0;
for i=1 to n do
    for j=1 to k do
        if Vector(hash[j](ip[i]))=0 then
            θ++;
            RSB.get(ip[i]);
            AAM.block(ip[i]);
        end if
    end for
end for
BR ← θ/n;
Return BR;
```

Dodge, once an attacker has been found, we can dismiss the feedback ratings that are associated with the attacker or the victim by discriminating the IP addresses. With the aid of the RSBs, our reputation system can shield against the malicious feedback ratings from the reputation measurement of each Web service.

V. THEORETICAL ANALYSIS AND LIMITATION

In this segment, we present a theoretical analysis of the proposed reputation measurement approach and a- likes feedback ratings prevention scheme, and dis- cuss the limitation of our coming. First we show the muscles of joy

Efficiency of the proposed reputation measurement approach. Then, we study the proposed prevention scheme on the false positive probability and success probability. Finally, the limitation is discussed.

a. Efficiency of measurement approach

As described above in Section 3, we use the CUSUM and PCC to solve the reputation measurement prob lem. For the proposed measurement approach, to actually measure the reputation of each Web service, the feedback ratings associated with the Web service will be computed by detecting malicious feedback ratings and adjusting the subjective feedback ratings. In the context of the malicious feedback rating detection, for each Web service, the CUSUM monitors a set of n feedback rating sample.

If $f_i \geq h$, then take action where $h > 0$ is the pre- specified CUSUM threshold. Note that if a sample i follows malicious feedback ratings, then the expect- ed score $E(z(y_i))$ should be positive so that f_i will eventually rise above threshold h. Moreover, $E(z(y_i))$ should be negative when the samples follow benign feedback ratings. We justify the choice of $z(y_i)$ in Section 3.1.

The CUSUM is adequate for identifying any abrupt change of benign feedback rating traffic to malicious feedback rating traffic. To understand this, note that if these feedback ratings are benign, $z(y_i)$ is negative (in the expected sense) and the corresponding f_i will stay around the zero value, regardless of how long the benign feedback rating traffic has been observed. However, when the benign feedback rating traffic turns to malicious feedback rating traffic, f_i increases and eventually surpasses the threshold h. Hence, the CUSUM prevents a malicious user from suppressing f_i with a long history of benign feedback rating traffic. This ensures that the CUSUM detects malicious feedback ratings in a timely manner. Of course, it is not a good approach when the number of feedback ratings is very little or none, e.g., for newly deployed Web services.

In the context of the feedback rating adjustment, the PCC is used to shield the influence of differ- ent preferences of users and adjust their subjective feedback ratings. The PCC adopts Eq.14 to obtain a set of similar users. Then the feedback similarity between two users can be computed by Eq.16. Based on the feedback similarity values, the feedback ratings with different preferences could be adjusted. Final- ly, the accurate reputation score can be computed. Hence, based on the above analysis, our proposed approach can measure the reputation of each Web subjective feedback ratings fill the Web service environment.

b. False positive probability of prevention scheme

From [26], we can demonstrate that the proposed prevention scheme can block malicious feedback ratings with a very low false positive probability (all the k bits are found to be 1 but the IP with a malicious feedback rating is not a member of S) by setting k = Ln 2m/n.. Hence, the false positive probability of the prevention system is really low. For example, when $k = 10$, the minimal *of* 0.1%. Hence, our proposed prevention scheme can block malicious feedback rating with a very small false positive probability, which indirectly improves the accuracy of reputation measurement foreachWebservice.

Moreover, the Bloom Filter of our approach stores each IP address with hash function. When malicious attack is found in the I-the sample interval, we merely discard the IP addresses of the sample interval. Because the number of feedback ratings within a sample intervals is very few, if the discarded feedback rating is right, the influence is also very limited for reputation measurement.

c. Success probability of prevention scheme

From [26], a malicious feedback rating over a Web ser- vice can be jammed with high probability by querying whether its IP is a member of S, assuming that the RSB could efficiently work. Hence, the proposed pre- Vientiane scheme can block malicious feedback ratings with high success probability.

Moreover, it can support different development environments of reputation systems with special false positive probability constraints. The Bloom filter guarantees no false negative, and in an ideal case, the success probability could reach 100%. But the proposed prevention scheme can not block malicious feedback rating with 100% probability because of these existing factors such as dynamic IP addresses, the low intensity of malicious feedback ratings and so on. Hence, the validation demonstrated that our proposed prevention scheme can block the malicious feedback ratings with very high probability.

d. Limitations of our proposed approach

The detection scheme of our plan of attack may give way when the volume of malicious feedback ratings is really depressed. The higher the volume of malicious feedback ratings is, the better the detection performance of our proposed attack.

The adjustment scheme of our approach is not desirable for a new service or the service used rarely, since the number of feedback ratings and

Users are really depressed. Of course, if there is not an ad- equate service community with massive services invoked, it cannot also play.

Notice that we propose a malicious rating prevention, only the active IP addresses and distributed rating attack using different IP addresses cannot be recognized and blocked in this theme.

When the operation of a service sudden changes from serious/tough to bad/good, if users move over very bad/good feedback ratings, this AP- approach will sustain a false positive (note that if the performance changes from fair to bad/good, this approach is till effective).

- Thither is a tradeoff between measurement accuracy and computation load. Thither is a heavy overload because of the complex computation in rating adjustment phase.

VI. PERFORMANCE EVALUATION

This section uses experiments to evaluate the guaran- tees of our proposed attack. We use a real world Web service QoS dataset and a feedback rating dataset in the experimentation. We also choose to apply simulation to generate feedback ratings because it enables us to study large-scale malicious and subjective feedback ratings of the reputationmeasurementsof Web services in service recommendations.

a. Experiment Setup

For the experiments on the divergence, we employ an actual feedback rating dataset1. The dataset consists of data from a real online dating service (Libimseti) [28]. Overall the dataset contains 194,439 users, who provided 11,767,448 feedback rating. Valuations are on a 1-10 scale, where ”10” is the best (integer feedback ratings only). But users who supplied at least 20 feedback ratings are included.

It is worth mentioning that because of the current limited availability of feedback rating data, many existing reputation systems [16,29-30] used simulation data for performance evaluation. In the simulation data, The simulated malicious and subjective feedback can reflect the real situations by going down the magnitude (e.g., 1,2,..., 10) of subjective feedback ratings and the density (e.g., 10%, 20%,..., 100%) of malicious feedback ratings [4, 5, 7, 10, 16]. Hence, in our experiments, we also use simulation to generate malicious and biased feedback ratings to assess the proposed AP- approach, as follows.

Malicious and biased feedback ratings are generated- Ed synthetics, which lets us to control the charac- teristics of the feedback ratings. Hence, to look into the performance of the reputation measurement for different feedback ratings, we simulated 500 services and 500 users. These users reported their feedback

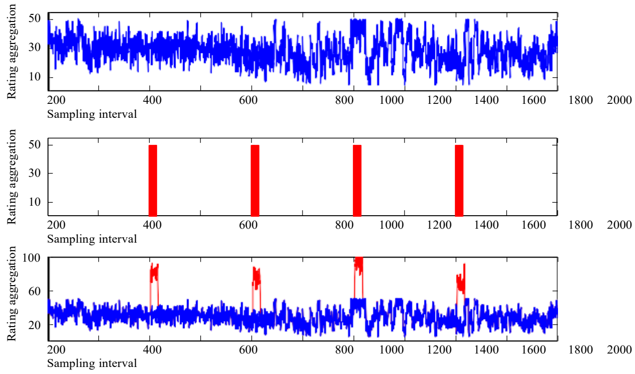


Fig. 4. An example: the synthesis of original feedback

Ratings with two characters, i.e., biased feedback ratings and malicious feedback ratings. Every feedback rating is likewise fixed to an integer feedback rating from 1 to 10. The malicious feedback ratings contain Malicious positive feedback ratings and malicious negative feedback ratings. In parliamentary law to facilitate experimental comparison with other accesses in a same experi- mental environment, as exhibited in Fig. 4, we choose a part of the feedback rating traffic in which a sampling interval (a specific point of time) contains 5 feedback rating and where feedback rating aggregation (y- axis) denotes the summation of 5 feedback rating. In Fig. 4 (a), the background feedback rating traffic is recorded. We think that there are simply a few malicious feed- back ratings in the Libimseti dataset because it has no business benefit or benefit conflicts in the network dating site2. In Fig. 4 (b), only (positive) malicious feedback ratings that are from the simulated malicious users are registered. As shown schematically in Fig. 4 (c), the original feedback ratings with malicious feedback ratings are generated synthetically, which allow us to look into the functioning of our approach.

Unless otherwise marked, the parameters of the CUSUM algorithm are set to (λ = 0.5, α = 0.7, and he = 0.7). In comparisons, all of the test cases and the runtime environment are the same. Each experimental result is accumulated at an average after each approach runs 10 times.

We take our experimental results from a PC with an Intel Core2 2.0GHz CPU and 2.0GB of RAM. The automobile is running Windows XP SP3, Mat- lab 7.6 and Java 1.4.8. We compare our approach with the reputation measurement approaches in [7] and [10], with deference to the divergence of the reputation measurement and the dependability of the composition service. The approach in [7] takes the client, the service, the normalized transaction feedback rating, and the set of optional attributes to make a service invocation history record that is employed to evaluate the reputation. Established on a compounding of a perceptual experience

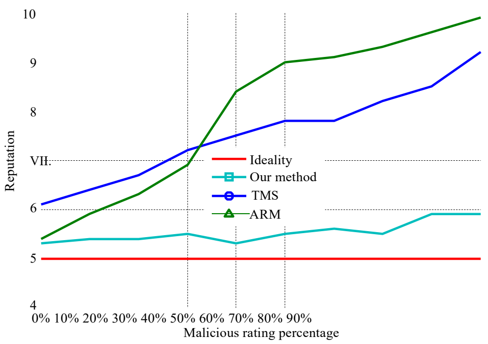


Fig. 5. Positive malicious feedback rating percentages.

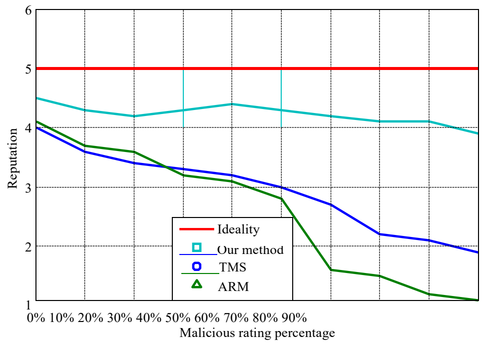


Fig. 6. Negative malicious feedback rating percent- ages.

Office and a disconfirmation function, the approach in [10] designed a feedback rating computation model, and then embraced the simple exponential smoothing approach to compute reputation scores. For example purposes, OA represents our approach, TMS represents the approach in [7] and ARM represents the approach in [10].

6.2 Experiment on Deviation

In this experiment, we compare our approach with TMS and ARM with respect to the deviation of reputa- tion measurement under a malicious feedback rating condition and normal feedback rating conditions.

Definition 1. Deviation: We define the deviation of the reputation measurement for each individual service as the deviation between the ideal reputation (all of the feedback ratings are objective, honest, and benign) and the actual reputation (feedback ratings are subjective or malicious).

6.2.1 Malicious feedback ratings

In this experiment, we vary malicious feedback rat- images from 0% to 90%, with a step of 10% with 100 random independent services. The 100 services are reckoned on an abstract service for a more objective measurement. As presented in Figs. 5-6, the measurement

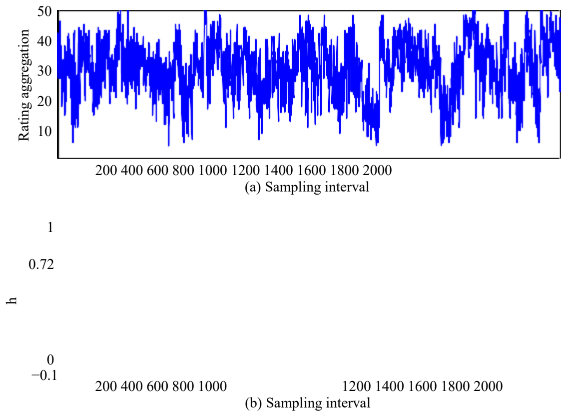


Fig. 7. Benign feedback rating detection. The parame- ter *h* represents the threshold value.

In Figs. 5-6, the Ideality line presents the ideal rep- utation. Although, in reality, it is impossible to obtain an ideal reputation for each service. With the Ideality line, we can effectively judge the operation of the three attacks.

In other words, the better the approach is, the smaller the deviation is.

From Fig. 5, we can see that the deviation of our approach is 5.53 on average, but the others are 7.54 (TMS) and 7.98 (ARM), respectively. When the posse- itive malicious feedback rating percentage increases, the deviations of TMS and ARM become larger. These relationships exaggerate the actual reputation value of the service and deceive or mislead users. Luckily, our glide slope is not tender to the positive male- course feedback ratings. With an increasing figure of positive malicious feedback ratings, it however delivers respectable performance.

From Fig. 6, we can see that the divergence of our AP- proach is 4.23 on average, while the deviations of TMS and ARM are 2.54 and 2.59, respectively. Specifically, when the negative malicious feedback rating percent- age is more than 50%, the measured reputations of the TMS and ARM will sharply diminish. Understandably, the measured reputation scores by TMS and ARM are inaccurate, which disguises the actual reputation of the service and creates the re-evaluated service fail to contend with existing services for market share. In contrast, our approach still works well despite the existing negative malicious feedback ratings.

In summary, from Figs. 5-6, with different numbers of malicious feedback ratings, the deviation of our approach is much smaller than those of the other approaches.

6.2.2 Benign feedback ratings

In the experiment, we apply our approach to the original feedback ratings without adding any malicious attack. The CUSUM algorithm is used to analyze the

(a) Shows the original feedback ratings, in which a sampling interval contains 5 feedback rating, and feedback rating aggregation (y-axis) denotes the summation of 5 feedback rating. Fig. 7 (b) shows the results, where all h values are mostly zeros and always a lot smaller than the doorway. Hence, no false alarms are described, which evidences that our proposed approach does not cause any effect on the accuracy of the reputation system under benign conditions.

6.3 Experiment on Optimality

In practical application, an important purpose of a reputation measurement is to help service recommendation systems find the optimal services under reputation attribute constraints. Nevertheless, because of the axis- tence of malicious and biased feedback ratings, the reputation score of a Web service often cannot reflect a service provider's real performance, which prevents users from customizing the best services according to their QoS requirements. Hence, in this segment, we compare the optimality of the composition service to further value our approach (The composition service that is unitary of the primary research issues of Service Computing is a service aggregating smaller and fine-grained services [6,8]).

For experiments on optimality, we employ an actual QoS dataset named WS-DREAM3 from [2]. The WS- DREAM data set takes approximately 1.5 million Web service invocation records of 150 users in 24 states. Values of three QoS attributes (i.e., Response Time, Response Data Size, and Failure Probability) are piled up from these 150 users on 10,258 Web services.

Established along the measured reputation scores of the three overtures, the QoS attributes can be expanded into four attributes (Response Time, Response Data Size, Failure Probability, and Reputation). Hence, we can get the best services under reputation attribute constraints with Mixed Integer Programming [31].

In our work, the overall utility [31] is the Agra- gation of the three QoS attributes (Response Time, Response Data Size, and Failure Probability) of the composition service under reputation constraints. To facilitate a comparison, we select ”RUX” to represent the overall utility under reputation constrains, where the reputation scores are measured using our approach. Likewise, ”RUY” and ”RUZ” represent the overall utility, for which the reputation scores are measured using TMS and ARM, respectively. We use ”OUT” to present the optimal solution of the Composi- ton service; in other words, the overall utility is only the collection of the three QoS attributes (Response Time, Response Data Size, and Failure Probability).

Definition 2. Optimality: We define the optimality of the composition service as the ratio of the overall utility and the optimality result with the tracing:

$$optimality = 100\% \times RUi/OUT, i = X, Y, Z, \qquad (7)$$

Where the better the glide slope, the larger the optimality. The optimality results of the three approaches are presented in Table 2. The number of QoS attributes is set to 4, and the number of QoS constraints is set to 1. The number of service candidates per service class is from 10 to 50, with steps of 10, and the number of service classes is set to 5. We vary the number of similar users (K) from 2 to 10, with steps of 2.

From Table 2, we can see that with the different number of similar users, the optimality of OA is the heaviest. Its optimality is 91.4% on average, while those of TMS and ARM are only 72.4% and 72.1%, respectively. Compared with TMS, most effects of OA are larger than 90%, while all of the effects of TMS are smaller than 90%. Compared with ARM, the results of OA are more important. Hence, the performance of OA is the best among the three overtures. So, with OA, the service, selection algorithm can obtain the optimal inspection and repairs. As a consequence, our approach can significantly better the functioning of the service selection for the service composition system in open service environments.

TABLE 2

The optimality of the composition service. The parameter K in the table represents the number of similar users.

K	Method	The number of service candidates				
		10	20	30	40	50
2	OA	**88.2%**	**89.5%**	**89.3%**	**84.2%**	**85.9%**
	TMS	79.2%	77.5%	72.6%	69.8%	69.9%
	ARM	72.5%	70.5%	70.4%	72.7%	77.4%
4	OA	**90.9%**	**91.4%**	**89.7%**	**90.8%**	**91.0%**
	TMS	68.8%	66.9%	68.3%	65.6%	63.5%
	ARM	74.7%	74.8%	73.4%	71.9%	72.4%
6	OA	**91.6%**	**91.0%**	**92.8%**	**89.4%**	**89.9%**
	TMS	77.2%	78.4%	76.3%	71.5%	61.8%
	ARM	80.2%	78.5%	66.7%	72.3%	79.1%
8	OA	**93.4%**	**94.5%**	**91.2%**	**93.0%**	**95.4%**
	TMS	72.9%	72.1%	72.6%	71.5%	65.8%
	ARM	79.3%	79.8%	74.5%	66.7%	69.8%
10	OA	**94.5%**	**91.5%**	**96.5%**	**95.0%**	**93.7%**
	TMS	72.2%	73.8%	61.3%	59.4%	65.9%
	ARM	66.8%	71.4%	64.9%	60.3%	61.7%

6.4 Experiment on success ratio

In a service recommendation system, another significant goal is to recommend reliable services for users. Nevertheless, because of the bankruptcy of the reputation measurement schemes, the selected service often dives- outs from the user's expectations, which may lead to service composition failure in practical applications. Therefore, the purpose of this experimentation is to compare the success ratio of our proposed approach with other accesses, with deference to the number of close-to- end QoS constraints. For this role, we touched on the number of service candidates per service class to 100 services, and we varied the number of QoS constraints (NQC) from 1 to 3, i.e., NQC=1,2,3. Furthermore, we

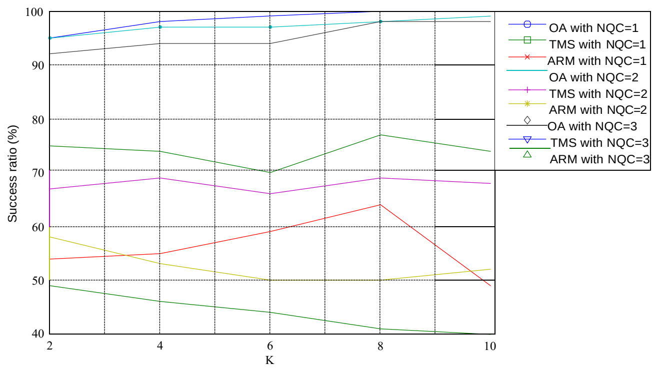


Fig. 8. Comparison of success ratios. The parameter K in the figure symbolizes the number of similar users. The parameter NQC in the image symbolizes the name- bar of QoS constraints.

Also get to the definition of the success ratio where the higher the success ratio of one approach is, the better its performance.

Definition 3. Success Ratio (SR) is how much the ratio of users' QoS constraints (Ci) to the monitored aggregated QoS values (Ui).

Shows the comparison of the success ratios among the approaches, where the parameter n is ready as an = 100. With different NQC, the success ratio of our plan of attack is a lot more eminent than those of the other two approaches. The overall success ratio of our approach is 96.9% on average, while those of the other two approaches are 66.6% (TMS) and 50.9% (ARM), respectively. These experimental results show that our approach effectively reduces the influence of male- cows and unfair feedback ratings on the success ratio of writing services.

6.5 Works on the Parameters

In this part, we examine the upshot of the parameters of our proposed attack on the optimality and success ratio results. As shown schematically in Fig. 9, the parameters contain the *fine* parameter α, the EWMA parameter λ, the CUSUM threshold h, and the number of similar users K. In our experiments, the number of QoS constraints is 1, and the number of service candidates per service class is 30.

6.5.1 *The gist of the fn parameter α*

Fig. 9 (a) depicts the effect of the α of the fun for our reputation approach. To demonstrate its impact clearly, we vary the value of α from 0.1 to 1 with a step value of 0.1. We set λ = 0.7, h = 0.7, and K = 10 and (4) the best performance of the approach is for values of λ in the interval [0.4,0.7].

6.5.3 *Effect of the CUSUM threshold h*

Fig. 9 (c) shows the effect of the CUSUM threshold h of our reputation approach where we vary the value of h from 0.1 to 1 with a step value

In the experiment. The image indicates the falling out: (1) The optimality is significantly trimmed down when the value of α is increased from 0.7 to 1. This notice suggests that the optimality will be cut down when the most probable percentage of the mean rate after the natural event of mass malicious feedback ratings has occurred increases; (2) the success ratio is not substantially influenced by the value of the α; and (3) the best performance of the approach is for values of α in the interval [0.4, 0.7].

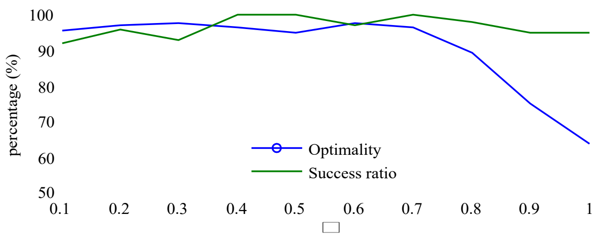
6.5.2 Effect of the EWMA parameter λ

Fig. 9 (b) depicts the result of the EWMA parameter λ for our reputation measurement approach. To demonstrate its impact clearly, we vary the value of λ from 0.1 to 1 with a step value of 0.1. We set α = 0.5, h = 0.7, and K = 10 in the experiment. Similar to before, the material body is obtained by getting hold of the norm of 10 runs. The image indicates the following: (1) the optimality is increased when the value of λ is increased from 0.1 to 0.5. Still, it is significantly trimmed down when the value of λ is increased from 0.6 to 1; (2) the success ratio is gradually increased at the early stage and is unwavering at the previous phase. This observation indicates that the success ratio will be steady when the current feedback ratings play a larger role than the historic

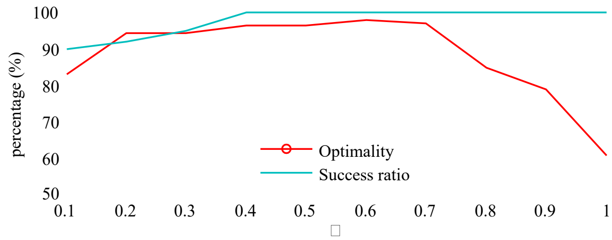
of 0.1. We set α = 0.5, λ = 0.7, and K . The figure is obtained by taking hold of the average of 10 runs. Fig. 9 (c) indicates the following: (1) the optimality is increased when the value of his increased from 0.1 to 0.6. How- ever, it is significantly trimmed down when the value of his increased from 0.7 to 1; (2) the success ratio is steady at the former stage and is slimmed when the value of h changes from 0.9 to 1. This notice suggests that the success ratio will be reduced because, with the increasing threshold value, the approach cannot filter malicious feedback ratings effectively; and (3) the best execution of the approach is for the values of h in the interval [0.4, 0.8].

6.5.4 *Essence of the PCC parameter K*

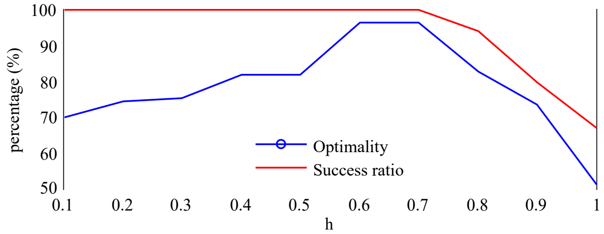
Fig. 9 (d) indicates the force of the PCC parameter K in which the measurement experiments are conducted, which vary the value of K from 2 to 10 with a step value of 2. We set α = 0.5, λ = 0.7, and h = 0.7 in the experiment. Fig. 9(d) shows that the optimality and the success ratio are increased when the value of K is increased from 2 to 10.



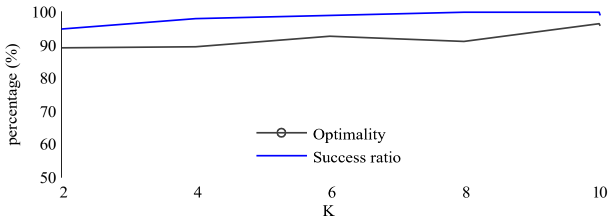
(a) Effect of the f_n parameter α



(a) Effect of the EWMA parameter λ



(c) Effect of the CUSUM threshold h



(d) Effect of the PCC parameter K

Fig. 9. Effect of warming up on different light bulb parameters. The parameter α represents the increasing mean rate of the change feedback rating traffic after a malicious attempt. The parameter λ is a constant that specifies the depth of computer storage of the EWMA, i.e., it specifies the rate at which "older" data record into the computation of the EWMA. The parameter h is the alarm threshold for detecting malicious attacks. The parameter K represents the number of similar users that uses the same Web service.

This notice shows that the higher the value of K is, the better the public presentation of the approach is, i.e., the more objective the reputation score is.

VIII. CONCLUSION

The proposed reputation measurement approach utilizes malicious feedback rating detection and feedback similarity computation to measure the reputation of Web services. The efficiency of our proposed attack is assessed and validated by the theoretical analysis and extensive experiments. The experimental results indicate that our suggested approach can achieve a trustworthy reputation measurement of Web services and greatly improve the service recommendation process. The proposed prevention scheme can identify the IP addresses with the offending feedback ratings and stop them using a standard Bloom filter. The theoretical analysis indicates the efficiency of the proposed prevention scheme in blocking malicious feedback ratings within the Web service recommendation system. Our on-going research includes investigating the parameters of sampling interval according to the number of feedback ratings, the number of sampling, duration and storage space, and constructing a common malicious feedback rating prevention scheme for Web service recommendation systems.

REFERENCES

[1] X. Chen, X. Liu, Z. Huang, and H. Sun. RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation, *In Proceedings of the 8th IEEE Interna- tional Conference on Web Services (ICWS'10)*, pages 9-16, 2010.

[2] Z. Zheng and M. R. Lyu. Collaborative reliability prediction of service-oriented systems. *In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)*, pages 35-44, 2010.

[3] E. M. Maximilien and M. P. Singh. Conceptual model of web service reputation. *SIGMOD Record*: 31(4): 36-41, 2002.

[4] Z. Malik and A. Bouguettaya. Evaluating rater credibility for reputation assessment of web services. *In Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE'07)*, pages 38-49, 2007.

[5] Z. Xu, P. Martin, W. Powley, and F. Zulkernine. Reputation- enhanced QoS-based web services discovery. *In Proceedings of the IEEE International Conference on Web Services (ICWS'07)*, pages 249-256, 2007.

[6] D. Ardagna and B. Pernici. Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*, 33(6): 369-384, 2007.

[7] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K. Nahrstedt. A trust management framework for service-oriented environments. *In Proceedings of the 18th international conference on World Wide Web (WWW'09)*, pages 891-900, 2009.

[8] S. Nepal, Z. Malik, and A. Bouguettaya. Reputation Propaga- tion in Composite Services. *In Proceedings of the IEEE Interna- tional Conference on Web Services (ICWS'09)*, pages 295-302, 2009.

[9] R. Jurca, B. Faltings, and W. Binder. Reliable QoS monitoring based on client feedback. *In Proceedings of the 16th international conference on World Wide Web (WWW'07)*, pages 1003-1012, 2007.

[10] N. Limam and R. Boutaba. Assessing Software Service Quality and Trustworthiness at Selection Time. *IEEE Transactions on Software Engineering*, 36(4): 559-574, 2010.

[11] J. R. Douceur. The Sybil Attack. *In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS'01)*, pages 251-260, 2002.

[12] F. Li, F. Yang, K. Shuang, and S. Su. A Policy-Driven Dis- tributed Framework for Monitoring Quality of Web Services. *In Proceedings of the IEEE International Conference on Web Services (ICWS'08)*, pages 708-715, 2008.

[13] S. Wang, Z. Zheng, Q. Sun, H. Zou, and F. Yang. Evaluating feedback ratings for measuring reputation of web services. *In Proceedings of the IEEE International Conference on Services Computing (SCC'11)*, pages 192-199, 2011.

[14] Z. Wang and J. Cao. Committee-based Evaluation and Selec- tion of Grid Resources for QoS Improvement. *In Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (Grid'09)*, pages 138-144, 2009.

[15] R. Zhou, K. Hwang. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 18(4): 460- 473.

[16] X. Li and L. Ling. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7): 843-857, 2004.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P net- works. *In Proceedings of the 12th international conference on World Wide Web (WWW'03)*, pages 640-651, 2003.

[18] J. Caverlee, L. Liu, and S. Webb. Socialtrust: tamper-resilient trust establishment in online communities. *In Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries (JCDL'08)*, pages 04-14, 2008.

S. Ries and E. Aitenbichler. Limiting Sybil Attacks on Bayesian Trust Models in Open SOA Environments. *In Proceedings of the 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC-ATC'09*), pages 178-183, 2009.

[19] C. Dellarocas. Immunizing online reputation reporting sys- tems against unfair ratings and discriminatory behavior. *In Proceedings of the 2th ACM conference on Electronic Commerce (EC'00)*, pages 150-157, 2000.

[20] V. A. Siris and F. Papagalou. Application of anomaly detec- tion algorithms for detecting SYN flooding attacks. *Computer Communications*, 29(9): 1433-1442, 2006.

[21] R. Radharamanan, A. Galelli, D. T. Alex, and L. L. Perez. Sensitivity analysis on the CUSUM method. *International Journal of Production Economics*, 33(1): 89-95, 1994.

[22] Z. Yanzhen, G. Liang, L. Ge, X. Bing, and M. Hong. Rec- tifying prejudicial feedback ratings in reputation based trust management. *In Proceedings of the IEEE International Conference on Services Computing (SCC'07)*, pages 530-535, 2007.

[23] J. L. Hellerstein, F. Zhang, and P. Shahabuddin. A statistical approach to predictive detection. *Computer Networks*, 35(1): 77- 95, 2001.

[24] U. Shardanand and P. Maes. Social information filtering: al- gorithms for automating 'word of mouth'. *In Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)*, pages 210-217, 1995.

[25] Bloom BH. Space/Time Trade-offs In Hash Coding with Al- lowable Errors.*Communications of the ACM*,13(7):422-426, 1970.

[26] Dharmapurikar S, Krishnamurthy P, Taylor DE. Longest prefix matching using bloom filters. *IEEE/ACM Transactions on Net- working*, 14(2):397-409, 2006.

[27] L. Brozovsky and V. Petricek. Recommender System for On- line Dating Service. *In Proceedings of the 6th Conference Zalosti (Zalosti'07)*, pages 1-12, 2007.

[28] I. Maarouf, U. Baroudi, and A. R. Naseer. Efficient monitoring approach for reputation system-based trust-aware routing in wireless sensor networks. *IET Communications*, 3(5): 846-858, 2009.

[29] Z. Malik and A. Bouguettaya. Reputation Bootstrapping for Trust Establishment among Web Services. *IEEE Internet Com- puting*, 13(1): 40-47, 2009.

G.sailaja is presently pursuing M.Tech (CSE) Department of Computer Science Engineering from visakha Institute of Engineering andTechnology,Visakhapatnam.

A.Harikumar,M.TECH is working as an Assistant professor in Department of computer science and engineering in visakha institute of engineering and technology, Visakhapatnam.