Reputation Evaluation with Malicious Feedback Prevention Using a HITS-Based Model

Okba Tibermacine, Chouki Tibermacine, Mohamed Lamine Kerdoudi

Reputation Evaluation with Malicious Feedback Prevention Using a HITS-Based Model

Okba Tibermacine
Computer Science Department
Biskra University, Algeria
Email: o.tibermacine@univ-biskra.dz

Chouki Tibermacine
LIRMM, CNRS
and Montpellier University, France
Email: Chouki.Tibermacine@lirmm.fr

Mohamed Lamine Kerdoudi
Computer Science Department
Biskra University, Algeria
Email: l.kerdoudi@univ-biskra.dz

***Abstract*—The reputation of web services is calculated by aggregating user feedback ratings. Though reputation is a subjective metric, it can be considered as a good indicator about services Quality of Experience, and henceforth, it can be used for recommending services in an open ecosystem. In this work, we propose a three-phase process for evaluating web service reputation by aggregating user feedback ratings. The relationship between users and services is modeled as a bipartite graph where an adapted HITS (*Hypertext Induced Topic Search*) algorithm is employed to distinguish between honest and malicious users in Phase I. Then, this model is used to evaluate, in Phase III, the reputation of web services from user ratings after punishing malicious users in Phase II. An experiment on a dataset of real Web services was conducted to validate the effectiveness of the proposed model in evaluating Web service reputation.**

***Keywords*-Web services, Reputation, HITS algorithm, User credibility evaluation.**

I. Introduction

The proliferation of services on the Web makes the selection of Web services a difficult task for building service-oriented applications. Therefore, it is crucial to provide effective recommendation and selection techniques that recommend satisfactory and trustworthy services. In the literature, reputation is a widely-used mechanism that allows the recommendation of optimal services from users' point of view. That is, reputation is seen as a collective measurement of the opinion of a community of users regarding their actual experiences with web services [1], [2]. Reputation reflects reliability, trustworthiness and credibility of web services and their providers [3], which consider it as an important factor for service selection and recommendation [4], [5].

Many reputation assessment models have been proposed in the recent years (e.g. [3], [6], [7]). In these models, reputation is calculated using all feedback ratings provided by users. However, these users can act maliciously, as in any open online system, and provide malicious feedback ratings. Obviously, malicious ratings lead to the presence of a bias towards positive or negative feedback ratings which influence drastically the evaluation of reputation and hence on the performance of the whole recommendation system.

In this paper, we propose a process for the detection of possible malicious users and the evaluation of service reputation after malicious users neutralization. We model the interactions between users and services as a bipartite graph. The process applies at two rounds an adapted HITS Algorithm [8] (HITS stands for *Hypertext Induced Topic Search*) to evaluate the credibility values of users and the reputation values of services. The choice of this algorithm is motivated by its successful application in Web page ranking and graph-based analysis. The goal of the first Phase is the evaluation of users credibility values based on the majority voting model. An analysis of the dispersion of these credibility values allows the identification of a threshold that separates honest users from potential malicious users. The process neutralizes the effect of malicious users by excluding their feedback ratings from the next Phase evaluation of service reputation. Finally, the process reconstructs a refined model without malicious users and applies the HITS algorithm for a final evaluation of service reputation.

The main contributions of this paper are the following:

- Proposing a HITS-based reputation evaluation process that enables: i) the detection of malicious users based on the majority voting model, and ii) the assessment of service reputation after the exclusion of malicious users feedback ratings.
- Conducting an experiment on a set of real-world web services to evaluate the proposed process against a selection of similar methods.

It is worth to mention that in some works of the literature HITS and graph-analysis algorithms have been successfully used for evaluating the reputation of services, but this algorithm, at the best of our knowledge, was not used for the detection of malicious users. This is further explained in the Related Work Section.

The remaining of this paper is organized as follows: Section II exposes our model of the problem that we are tackling. Section III presents the proposed process to detect possible malicious users and to evaluate the reputation of services. Section IV describes the evaluation of the process. Before concluding and presenting some perspectives in Section VI, we discuss the related work in Section V.

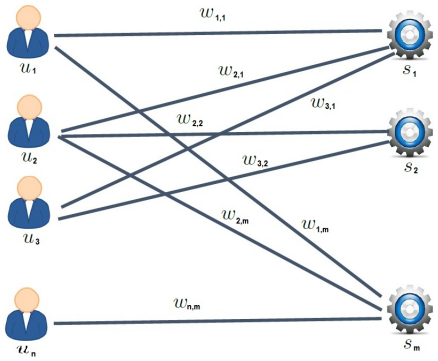

Figure 1. A bipartite graph that represents the target problem

II. System Model

We represent users, services, and their relationship as a bipartite graph $G = (U, S, W)$, where,

- U and S are two classes of vertex, where, $U \cap S = \phi$.
- U represents a set of users and S represents a set of services.
- W is the edge set. The edges are weighted and represent a feedback rating.
- Each edge connects a vertex in U to one in S.

Figure 1 depicts an example of such graphs. Where, u_i ($i \in \{1, 2, ..., n\}$) represents a given service user, and s_j ($j \in \{1, 2, ..., m\}$) represents a service. The edge weight w_{ij} represents the feedback ratings attributed by user u_i to service s_j.

The goal of this work is to estimate for each user $u_i \in U$ its credibility value c_i, such that $c_i \in [0, 1]$. And then, we estimate for each service $s_j \in S$ its reputation value r_j such that $r_j \in [0, 1]$ which is an aggregation of all feedback ratings in W_j s.t. $W_j = \{w_{ij} | \quad \forall u_i \in U \quad and \quad c_i \geq \theta\}$. θ is a threshold above which user u_i with credibility value c_i is considered as a honest user.

It is worth mentioning that the aggregation function can be a simple union of feedback ratings or a process of evaluation that considers a number of factors (e.g. User credibility, Time sensitivity and Community maliciousness density) to assess the reputation [9]. For the sake of simplicity, the aggregation function in this work includes only feedback ratings and user's credibility to show that a two-phase Hits evaluation is able to fairly assess the credibility of users and then the reputation values of services.

III. Reputation evaluation process

We first present the overall process and then we detail each of its phases.

A. Overview

Unfair feedback ratings influence drastically on the evaluation of service reputation, especially if theses ratings are attributed by malicious users [3]. A possible solution for fairly evaluate service reputation consists in identifying malicious users, and then neutralizing their negative effects by neglecting their feedback ratings during the assessment of reputation. The process that we propose for services reputation evaluation goes through three phases as depicted in Figure 2.

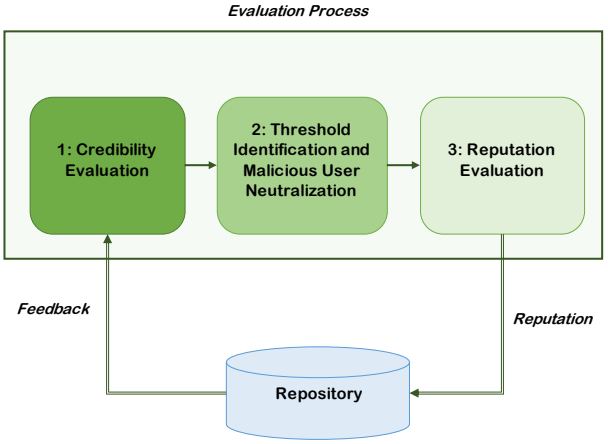

Figure 2. Evaluation process

Phase 1 is a HITS-based evaluation of user credibility where we apply an adapted version of the HITS algorithm to evaluate theirs credibility values. These values enable, in phase 2, the identification of two classes of users; Honest and malicious users.

Phase 2 is statistical-based identification of the threshold which is used to separate honest and malicious users. Feedback ratings attributed by malicious users are discarded from the set of feedback ratings. This set is used, in phase 3, to evaluate the reputation of services.

Phase 3 is a second round of service reputation evaluation using the results obtained in selected in phase 2.

The algorithms of each step of the process are given in the following subsections.

B. Phase 1: Credibility Evaluation

The HITS algorithm is a link analysis algorithm, proposed by Kleinberg [8], to rank web pages based on link structure among them. The algorithm defines two main concepts; *Authorities* and *Hubs*. An authority is a Web page with a lot of inbound links, which means that it is a web page which is highly reliable and holds informative data and many other web pages index it. A hub page is a web page that serves as an organizer of information on a given topic and points to many good authority pages on a particular subject. HITS assumes that an authority is referenced by many hubs while a hub references many authorities.

The same idea is applicable to the evaluation of user credibility and service reputation; a user rates many services,

Algorithm 1: A HITS-based algorithm for user credibility evaluation

Input: Feedback rating set: W;
User set: $U = \{U_1, U_2, .., U_n\}$;
Service set: $S = \{S_1, S_2, .., S_m\}$;
Output: Credibility values : $C = \{C_1, C_2, .., C_n\}$;
Reputation values : $R = \{R_1, R_2, .., R_m\}$;
begin
/* *Initialization* */
$C_i = 1 \quad \forall i \in \{1, 2, .., n\}$;
$R_j = 0 \quad \forall j \in \{1, 2, .., m\}$;
repeat
/* *Updating Service reputation values* */
foreach $j \in \{1, 2, .., m\}$ **do**
Update R_j by using Equation 1
/* *Updating User credibility values* */
foreach $i \in \{1, 2, .., n\}$ **do**
Update C_i by using Equation 2
until *Convergence or the Maximum number of iterations is reached*;
return C, R;

while a service receives feedback ratings from many users. By applying this idea on the bipartite graph G introduced in Section II, the reputation R_j of service S_j is given by Equation 1.

$$R_j = \frac{\sum_{u_i \in U_j} c_i \times w_{ij}}{|U_j|} \tag{1}$$

Where U_j denotes the subset of users that have assigned a feedback rating for service S_j, and $|U_j|$ is its cardinality. In the first iteration C_i initialized with 1.

After the second iteration, the credibility C_i of user $u_i \in U$ is given by the Equation 2.

$$C_i = \frac{\sum_{s_j \in S_i} \frac{\min(w_{ij}, R_j)}{\max(w_{ij}, R_j)}}{|S_i|} \tag{2}$$

Where S_i denotes the subset of services that are rated by user i. The credibility of user u_i is the average of all the ratios between the feedback rating given by the user for each service and the reputation of that service which was calculated in the previous iteration. If the feedback w_{ij} is close to what the majority have assigned to the web service, represented by the reputation R_j, the ratio increases and consequently the credibility of the user improves.

Algorithm 1 evaluates the reputation of services (using Eq. 1) considering initially that all users are honest (with credibility values all equal to one). Then, it updates the credibility of each user based on the comparison between its feedback and the reputation of the service using Eq.2.

Algorithm 2: A HITS-based algorithm for service reputation evaluation

Input: Feedback rating set: $W = \{W_1, W_2, .., W_n\}$;
User set: $U = \{U_1, U_2, .., U_n\}$;
service set: $S = \{S_1, S_2, .., S_m\}$;
Credibility values : $C = \{C_1, C_2, .., C_n\}$;
Credibility Threshold : θ ;
Output: Reputation values : $R = \{R_1, R_2, .., R_m\}$;
begin
/* Malicious user Neutralization */
foreach $i \in 1, 2, .., n$ **do**
if $C_i \geq \theta$ **then**
$W'.add(W_i)$;
$W = W'$;
reConstruct_UserSet(W', U); /* Update S */
reConstruct_ServiceSet(W', S); /* Update U */
/*** Apply HITS to evaluate Service reputation ***/
$C_i = 1 \quad \forall i \in \{1, 2, .., n\}$;
$R_j = 0 \quad \forall j \in \{1, 2, .., m\}$;
repeat
/* *Updating service reputation values* */
foreach $j \in \{1, 2, .., m\}$ **do**
Update R_j by using Equation 1
/* *Updating user credibility values* */
foreach $i \in \{1, 2, .., n\}$ **do**
Update C_i by using Equation 2
until *Convergence or the Maximum number of iterations is reached*;
return R;

Then, the algorithm starts another iteration of evaluating the reputation of services based on the updated credibility values and after that evaluating credibility values based on the new reputation values. The algorithm stops the evaluation when the values of reputation and credibility are stabilized or when a maximum number of iterations is reached.

C. Phase 2: Malicious Users Identification and Neutralization

The first round of applying HITS algorithm (phase one) allows an initial evaluation of service reputation and user credibility. In the second phase of the evaluation process, we analyze the credibility of users to identify potential malicious users. The iterations of the HITS algorithm evaluate the credibility of users based on the majority voting model which leads to group similar users together with very close credibility values. Analyzing the dispersion of these values

can identify one group or two groups of users (i.e. Honest and/or Malicious users). The identification of these groups goes through the following steps:

1) Sort credibility values (Vector C).
2) Compute the mean μ of these values s.t.
$$\mu = \frac{\sum_{i=1}^{N} c_i}{N}$$
3) Compute the standard deviation σ s.t. $\sigma = \sqrt{\sigma^2}$ where,
$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}, \qquad s.t. \quad x_i = c_i$$
4) Find the largest gap (i.e. distance) between each two adjacent values. Let's denote this gap as $G_{k,l} \in R^+$ where k and l are the adjacent credibility values.
5) If the largest gap is less than or equal to the standard deviation i.e. $G_{k,l} \leq \sigma$ then all the credibility values are close to each other which means that users are similar and this group represents honest users. Otherwise (the case where the gap is lager than the standard deviation i.e. $G_{k,l} > \sigma$), a threshold θ between two groups of users can be identified as the average between the adjacent credibility values with the largest gap ($\theta = \frac{k+l}{2}$). θ is the threshold that separates between the two groups. In this case, the group with the lowest arithmetic mean is the group of malicious users.

This phase ends with the elimination of all malicious users whose credibility values belong to the group with the lowest arithmetic mean. This task (described in the first part of Algorithm 2) ends with the removal of all malicious users form the set of users U and consequently the elimination of all edges between theses users and the ranked services.

D. Phase 3: Final Evaluation of Service Reputation

After the identification of possible malicious users and the neutralization of their effects by excluding their feedback in Phase 2, a new model is constructed from the updated user and service sets. A second round of HITS algorithm allows a final evaluation of service reputation.

Algorithm 2 defines the process of excluding possible malicious users in the beginning, and constructing the refined model to evaluate the reputation of services from feedback ratings of the remaining users. The algorithm updates iteratively both new credibility values and service reputation values until the stabilization of values or the maximum number of iterations is reached. The output of this phase is the set of final service reputation scores.

IV. EVALUATION

To evaluate the proposed HITS-based reputation evaluation process, we conducted an experiment on a set of real Web services. We compare our results (results with and without malicious user neutralization) with three reputation evaluation methods from the literature.

A. Performance Comparison

We compare the performance of the proposed evaluation method with the the following reputation assessment methods:

1) The Average Method (labeled in this experiment as: **Average**): The average algorithm takes the mean of all explicit feedback ratings as a reputation value. It is widely used in commercial services like Amazon [10].
2) The approach used by Wang *et al* [3] (labeled in this experiment as: **Cusum**), where the reputation score $q(s_j)$ of a service s_j is assessed as follows:
$$q(s_j) = \frac{1}{n}\sum_{i=1}^{n} r_i$$
where, r_i represents the $i-th$ feedback rating, and n represents the number of feedback ratings.
The approach assesses reputation value using only pure feedback ratings (fair ratings or adjusted malicious ratings), because the approach applies a malicious feedback ratings prevention scheme based on the Cumulative Sum Method (CUSUM). The CUSUM monitors n feedback ratings sample interval. For each sample interval, they assign a score $Z(y_i)$ which is assessed as follows:
$$Z(y_i) = \frac{\mu_1 - \mu_0}{\sigma^2}(y_i - \frac{\mu_1 - \mu_0}{2})$$
where, rating feedback sample intervals are represented by $\{y_1, .., y_n\}$ and the variable $y_j (y_j = \sum_{i=1}^{m} r_j)$ $(i \leq j \leq n)$ $(m = 1, 2, ...)$. μ_0 and μ_1 are the mean feedback rating traffic before and after the change. When a sample interval is available, the CUSUM f_i is updated as follows:
$$f_i = \max(f_{i-1} + Z(y_i), 0)$$
if $f_i \geq h$ then a positive shift occurs in the $n-th$ sample which means that there is an abnormal detection point (presence of malicious feedback rating). In our implementation of this scheme, we set h to 0.7 (this is based on the authors experiment settings).
3) The approach proposed by Mekouar *et al*. [11] (labeled in this experiment as: **TrustWS**), where the reputation of a web service is assessed as the difference between positive and negative feedback ratings divided by the sum of both. Reputation is set to 0 when the sum of feedback ratings equates to 0. This approach do not include the credibility of users for reputation assessment.

These three methods are compared with the initial results (labled **HITSW**) obtained in phase one without the neutralization of malicious users, and the final results (labled **HITSN**) obtained in phase three after the neutralization of malicious users.

B. Evaluation Metrics

In order to measure the quality of the evaluation provided by our process in comparison with other methods, we have used two metrics: Mean Absolute Error (MAE) and Root-Mean-Squared-Error (RMSE).MAE is a quantity that measures how close are the estimations (predictions) to the eventual outcomes. MAE is defined in Equation 3. RMSE quantifies the difference between predictions and eventual outcomes. RMSE gives a relatively high weight to larger errors. It is defined in Equation 4.

$$MAE = \frac{\sum_{i=1}^{n} |R_i - \hat{R}_i|}{n} \tag{3}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{R}_i - R_i)^2}{n}} \tag{4}$$

In Equations 3 and 4:

- n represents the number of tested services.
- R_i denotes the actual reputation (reputation which is calculated by aggregating simulated feedback ratings).
- $\hat{R}_i$ denotes the estimated (predicted) reputation calculated by the proposed method (or selected methods from the literature)

We note that MAE gives equal weights to all the individual differences. But, when large errors are particularly undesirable, RMSE is more useful, because it gives high weights to large errors [12]. In addition to the MAE and RMSE metrics, we have used the Precision and Recall metrics to evaluate the identification of malicious users.

C. Data Collection

The used web services in this experimentation are extracted from WSDream [13]and QWS [14] datasets. WSDream dataset holds a set of 5825 web services each with two QoS values (response time and throughput) that are given by 339 users in different locations. QWS dataset holds 365 web services with 9 QoS metrics including: Response time, Availability, Throughput, Successability, Reliability, Compliance, Best Practices, Latency, and Documentation.

We have used in our experiment a single set of services that constitutes the intersection of the two datasets (services that belong to the two datasets). Indeed, we have matched the services from the first dataset with the services in the second dataset by comparing their URIs, names, and WSDL file sizes. The obtained set is composed of 409 services (53 services are redundant but with different endpoints and QoS metrics), for each service, we have used the values of response time and throughput that are provided by the 339 users from the WSDream dataset. For the seven other QoS metrics (availability, documentation,etc.), we have used the values provided by the QWS dataset. These values are static for all users. It allows us to maximize the number of QoS metrics with different response time and throughput values.

D. Feedback Rating Simulation

Due to the limited number of feedback rating data, in the selected web services and in order to evaluate our process in a reliable way, we have enriched the collected data with additional feedback rating values. To do so, as several web service reputation management approaches (e.g. [6], [15], [16]), we have simulated the interactions between a set of 339 users and the 409 web services. At the end of the interaction, for each user, we generate feedback values for its consumed Web service.

However, each service has an actual performance level used to quantify quality perception. In our experiment, it represents on a scale of 10 how good is the overall quality provided by the service. We denote the actual performance level as *PerfVal*. It can be calculated based on a utility function of the delivered service [17]. Our utility function is based on the root mean square. The latter is a measure of the magnitude of the scaled QoS metrics. Thus, *PerfVal* of service s_i is assessed as follows:

$$PerfVal(s_i) = 10 \times \sqrt{\frac{\sum_{j=1}^{k} Scal(Q_{i,j})^2}{k}} \tag{5}$$

where, k is the number of used QoS metrics (9 in our experiment). $Scal(Q_{i,j})$ is the scaling function, which is defined by Equation 6, if the quality is positive (i.e., the higher is the value the higher is the quality), and by 1 minus the same formula otherwise.

$$Scal(Q_{i,j}) = \frac{Q_{i,j} - Min(Q_j)}{Max(Q_i) - Min(Q_j)} \tag{6}$$

$Min(Q_j)$ and $Max(Q_j)$ are respectively the minimum and maximum recorded values of the quality Q_j.

We have two kinds of users: honest and malicious users. In practice, honest users rate a service based on its *PerfVal* within the interval $[Max(0, \textit{PerfVal} - 2), Min(\textit{PerfVal} + 2, 10)]$ [6]. For instance, if *PerfVal*=7, honest feedback ratings could be 5, 6, 7, 8, and 9. The deviation with ± 2 from *PerfVal* represents the natural variation between user opinions. We take this assumption in our simulation. For the honest users, we generate randomly values in this interval, and the malicious users randomly rate the same service outside the interval (always on a scale of 10). For the previous example, malicious feedback ratings could be 0, 1, 2, 3, 4, and 10.

Moreover, Whitby et al. [18] and Malik et al. [6] claim that high maliciousness densities are unrealistic in real world applications. Thus, we use in this simulation different malicious user densities that varies in the interval [5% - 40 %].

We have conducted 10 rounds of simulation to calculate the reputation of the selected services with our HITS-based

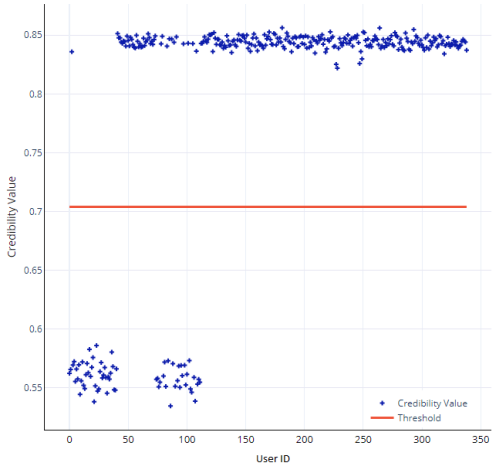

Figure 3. A sample of credibility evaluation and Threshold identification with a density of malicious users that equals to 25%.

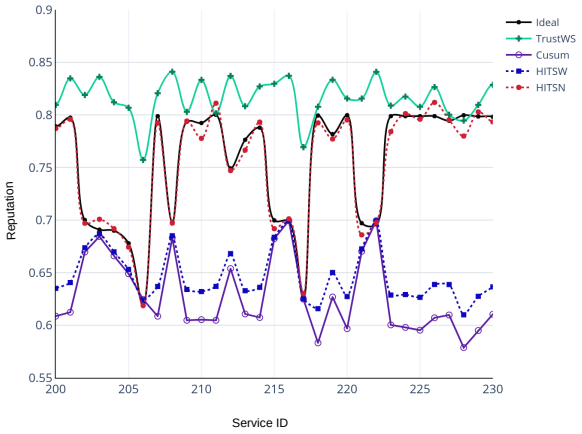

Figure 4. Comparison between the ideal reputation values and values obtained by the candidate methods for 30 services with maliciousness density quals to 25%.

process and with the candidate methods. The reputation is the average of the calculated reputation values of all rounds.

E. User Credibility Evaluation

Using the configuration described previously, we have run many simulation instances applying Algorithm 1. In each run, we varied the maliciousness density and we observed the results of evaluating user credibility.

Figure 3 depicts an instance of user credibility evaluation obtained by our HITS-based model, with 20% as maliciousness density (i.e. 20% of users are malicious in the system). As we can see, two clusters are automatically generated; the first with high credibility values (credibility ranges between 0.8 and 0.9) which represent honest users, and the second with low credibility values (credibility ranges between 0.45 and 0.58) which represents malicious users.

Moreover, by varying the malicious density between 5% to 40%, we found that precision and recall of identifying malicious users is always 100%, which means that our first HITS-based algorithm accurately identify malicious users.

F. Service Reputation Evaluation

We applied our process to evaluate the reputation of Web services using the generated feedback ratings. We run many instances varying the malicious density. For illustration we show the reputation values of 30 services. Figure 4 depicts a comparison between (1) the ideal reputation (aggregartion of PerfVal), (2) the reputation of the candidate methods (namely (**TurstWS**, **Cusum**)) and (3) the reputation values obtained by applying our process without Malicious neutralisation (**HITSW**), and the reputation values obtained by applying the neutralisation of malicious users (**HITSN** in red color). As we can see, the closest results to the ideal reputation is the results obtained by our process with malicious user neutralisation. This observation is remaining correct for all other services during all simulation runs with different malicious rates.

G. Results and Discussion

Table I summarizes a comparison between MAE and RMSE obtained by our model (HITSW (evaluation without neutralization of malicious users), and HITSN (evaluation with neutralization) and the other candidate methods. From this table, we can observe that our two-phase model gave smaller MEA and RMSE values, indicating better accuracy consistently, with precision of 99% ($Precision = 1 - MEA$). Moreover, the application of Algorithm 1 without excluding malicious users gives results close to those obtained by the average algorithm.

Another important observation is the steady evolution of RMSE with different densities of malicious users with **HITSN**. In contrast to all other methods, which are very sensitive to the change in maliciousness density, our method gives good performance scores with all densities, even with the highest one – with 40% of malicious users, RMSE is 3.32, which is very close to the lowest score (3.04). We can say that according to these results, we may safely use the proposed process to correctly evaluate the credibility of users and the reputation of services.

H. Threats to Validity

In our evaluation, we have used a dataset with a quite limited number of services. Using a larger dataset would give more accurate results. But unfortunately, we did not found any other datasets with real-world data. We tried to mitigate this threat by performing many simulations.

In a real world application, results using this process can be sensitive to the number of the feedback ratings which is known as the cold start problem. To overcome this threat we propose to generating hypothetical feedback ratings from

Table I
MEA AND RMSE RESULTS FOR THE CANDIDATE METHODS

Method	Average		TrustWS		CUSUM		HITSW		HITSN	
Mali-density	**MEA**	**RMSE**	**MEA**	**RMSE**	**MEA**	**RMSE**	**MEA**	**RMSE**	**MEA**	**RMSE**
5%	*0.142877*	3.857834	*1.693836*	34.61097	*0.65193963*	13.6513645	*0.108357*	3.357723	***0.075961***	**3.14275**
10%	*0.250164*	6.067272	*1.603579*	32.64668	*0.52265618*	11.318582	*0.178821*	4.4722	***0.078775***	**3.049035**
15%	*0.348422*	8.511924	*1.493836*	30.38836	*0.39974309*	9.5675793	*0.243923*	5.955679	***0.081487***	**3.203175**
20%	*0.463187*	11.36647	*1.410094*	28.72234	*0.33081475*	9.05587107	*0.33092*	8.004757	***0.084558***	**3.150615**
25%	*0.568663*	14.1054	*1.292422*	26.39261	*0.37330883*	9.06999431	*0.40957*	10.0917	***0.085657***	**3.160125**
30%	*0.683105*	16.88108	*1.173526*	24.25228	*0.47594905*	10.2670184	*0.502568*	12.4197	***0.084306***	**3.120733**
35%	*0.795552*	19.73738	*1.054889*	22.28034	*0.58678796*	12.2707987	*0.602304*	14.90803	***0.095006***	**3.265824**
40%	*0.915061*	22.66924	*0.938583*	20.55926	*0.70374139*	14.7922136	*0.711492*	17.58969	***0.093956***	**3.321841**

service QoS using simulations. This rating allows an initial reputation evaluation which will be replaced progressively when real feedback ratings will become available (received by real users). In addition, even that we used only two factors in the evaluation of the service reputation, it is still possible to include other factors such as time sensitivity in Equation 1.

Although the proposed process is essentially applied to the evaluation of web service reputation, it is possible to generalize the process to any kind of services or software APIs in general.

V. RELATED WORK

Trust and reputation management became a topic of interest in many research fields. Several papers (e.g. [19]) survey the existing academic and commercial reputation systems. Most of these works address the problem of aggregating many factors to fairly assess the reputation and trust of the interacting elements (e.g. services and users).

Among the similar approaches, Conner *et al.* [20] introduce a reputation-based trust management framework that makes a customized trust level assessment of client requests based on shared feedback ratings using caching mechanisms.

Mokarizadeh *et al.* in [21] propose to use service reputation scores for improving the quality of automatic service selection and composition. They assess the reputation of a service based on analysing relationships in a social network model. Though these approaches provide mechanisms for reputation evaluation, they do not provide any solution for detecting maliciouobtentions users.

Malik *et al.* in [6] propose a decentralized reputation system for web service orchestrations called `RateWeb`. The proposed system is based on a peer-to-peer (P2P) service model where each peer (service) is a consumer and a provider of services. Each peer can calculate and update the reputation of the other peers. Thus, the reputation value of a service can be evaluated in regards to a user perspective, which gives different values for the same service. This observation remains correct for user credibility evaluation. In our work, we refer to a majority voting model that collaboratively identifies malicious users and neutralizes their negative feedback ratings.

The authors in [22] have proposed a trust and reputation management system for peer-to-peer systems, and for web service environments (TrsutWs) in [11]. TrustWs enables to select web services based on the feedback gathered from past transactions. They assess the reputation of web services as the ratio of the difference between positive feedbacks (issued from satisfactory transactions) and negative feedbacks (issued from unsatisfactory transactions), and the sum of all feedbacks. However, this work do not provide any mechanism to deal with malicious users and subjective ratings, which weakens the reliability of the system.

Wang *et al.* propose a reputation measurement and malicious feedback rating prevention scheme employing "Bloom Filtering" to enhance the recommendation performance [3]. This allows to identify the IP addresses with offending feedback ratings and filter them out. They propose to detect malicious feedback ratings by using the Cumulative Sum Control Chart. After that, they reduce the effect of subjective user feedback preferences by employing the Pearson Correlation Coefficient. At the end, they measure the reputation of services based on the former ratings, and store the calculated reputation scores in the repositories to be used for service recommendation. In our work, we do not identify IP addresses of malicious users, but we iteratively refine the users' credibility score and service' ranking score until the stabilization of these values and obtaining of the final service reputation score.

Limam and Boutaba [17] proposed a framework for reputation-aware service selection and rating. It allows to automate the selection and the rating of software services. They introduced an algorithm that acts as a user-centric and reputation-aware service recommender. It enables to determine a services suitability to a particular users preferences in terms of quality and cost. In addition, the authors proposed a rating function which provides objective feedback on a delivered service without human intervention. They have also introduced a reputation derivation model that aggregates all of the feedback into an overall reputation score. We argue that with the existence of malicious and subjective feedback rating, their framework and model can not predict easily accurate feedback ratings in real web services environments.

The user credibility is widely addressed in the literature. In [23] an algorithm is proposed to adjust customer (for products) credibility values by feedback considering the rating consistency. They combine customer credibility together with originally assigned ratings. [24] have shown how the incorporation of user credibility into the recommendation

process can enhance the relevance of results recommended by the HeyStaks system. The authors in [25] have proposed six reputation-based algorithms, where the users' reputation is determined by the aggregated difference between the users' ratings and the corresponding objects' rankings.

Compared to all these works, our solution is based on the evaluation of user credibility and service reputation using only fair feedback rating based on a majority voting model. In addition, we propose an algorithm that identifies and neutralizes the effect of potential malicious users, which enables the improvement of evaluation accuracy.

VI. Conclusion

To overcome the effect of malicious users on the evaluation of Web service reputation, we proposed an adapted HITS-Based reputation process built on the implicit behaviors of service users. Firstly, the proposed process evaluates the credibility of users based on the majority voting concept using an updated HITS analysis. These credibility values enable the identification of malicious users which are excluded by the process in the second phase. The process uses the HITS-based model once again to evaluate service reputation values from feedback rating given by the remaining users. The convergence of the process was analyzed and experimented on real Web service datasets while comparing our results with a set of existing similar methods. We demonstrated that our approach provides more accurate results and is more efficient than the other methods.

As a future work, we plan to run more experiments to compare the proposed model with other methods. In addition, we plan to extend the model in order to allow the prediction of service reputation from its Quality of Service attributes, through which we can find a solution to the whitewashing and cold start problems.

References

[1] E. M. Maximilien and M. P. Singh, "Conceptual model of web service reputation," *Acm Sigmod Record*, vol. 31, no. 4, pp. 36–41, 2002.

[2] H. T. Nguyen, W. Zhao, and J. Yang, "A trust and reputation model based on bayesian network for web services," in *IEEE ICWS'10*. IEEE, 2010.

[3] S. Wang, Z. Zheng, Z. Wu, M. Lyu, and F. Yang, "Reputation measurement and malicious feedback rating prevention in web service recommendation systems," *IEEE TSC*, vol. PP, no. 99, pp. 1–1, 2014.

[4] Y. Wang and J. Vassileva, "Toward trust and reputation based web service selection: A survey," *ITSSA journal*, vol. 3, no. 2, pp. 118–132, 2007.

[5] X. Ye, J. Zheng, and B. Khoussainov, "A robust service recommendation scheme," in *IEEE SCC'13*, 2013.

[6] Z. Malik and A. Bouguettaya, "Rateweb: Reputation assessment for trust establishment among web services," *Journal on Very Large Data Bases*, vol. 18, no. 4, pp. 885–911, 2009.

[7] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *Journal of Parallel and Distributed Computing*, vol. 75, pp. 184–197, 2015.

[8] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, "The web as a graph: measurements, models, and methods," in *COCOON'99*. Springer, 1999.

[9] Z. Malik, I. Akbar, and A. Bouguettaya, "Web services reputation assessment using a hidden markov model," in *ICSOC'09*. Springer Berlin Heidelberg, 2009.

[10] X. Zhou, D. Lin, and T. Ishida, "Evaluating reputation of web services under rating scarcity," in *IEEE SCC'16*, 2016.

[11] L. Mekouar and Y. Iraqi, "Trustws: A trust management system for web services," in *ISWS'10*, 2010.

[12] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE TSC*, vol. 6, no. 3, pp. 289–299, 2013.

[13] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *IEEE ICWS'10*, 2010.

[14] E. Al-Masri and Q. H. Mahmoud, "Qos-based discovery and ranking of web services," in *ICCCN'07*. IEEE, 2007.

[15] H. T. Nguyen, J. Yang, and W. Zhao, "Bootstrapping trust and reputation for web services," in *IEEE 14th CEC'12*, 2012.

[16] O. Tibermacine, C. Tibermacine, and F. Cherif, "Estimating the reputation of newcomer web services using a regression-based method," *JSS*, vol. 145, pp. 112–124, 2018.

[17] N. Limam and R. Boutaba, "Assessing software service quality and trustworthiness at selection time," *IEEE Transactions on Software Engineering,*, vol. 36, no. 4, pp. 559–574, 2010.

[18] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," in *7th Int. Workshop on Trust in Agent Societies*, vol. 6, 2004.

[19] F. G. Mármol and M. Q. Kuhnen, "Reputation-based web service orchestration in cloud computing: A survey," *Concurrency and Computation: Practice and Experience*, 2013.

[20] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K. Nahrstedt, "A trust management framework for service-oriented environments," in *WWW'09*, 2009.

[21] S. Mokarizadeh, N. Dokoohaki, M. Matskin, and P. Küngas, "Trust and privacy enabled service composition using social experience," in *Soft. Services for e-World*. Springer, 2010.

[22] L. Mekouar, Y. Iraqi, and R. Boutaba, "Incorporating trust in network virtualization," in *2010 IEEE CIT'10*. IEEE, 2010.

[23] R. Zhang, M. Gao, X. He, and A. Zhou, "Learning user credibility for product ranking," *Knowledge and Information Systems*, vol. 46, no. 3, pp. 679–705, 2016.

[24] K. McNally, M. P. O'Mahony, and B. Smyth, "Modeling user and result reputation in collaborative web search," in *AICS'11, Trinity College Dublin*, 2011.

[25] R.-H. Li, J. Xu Yu, X. Huang, and H. Cheng, "Robust reputation-based ranking on bipartite rating networks," in *SIAM:SDM'12*, 2012.