An Iterative Opinion Aggregation Algorithm for Reputation System Voting Schemes in MANETs and Peer-2-Peer Networks

Filippos D. Koravos\textsuperscript{1,2}  
filippos@inf.uth.gr

\textsuperscript{1}University of Thessaly  
Dept. of Communication & Computer Engineering  
Glavani 37 & 28th October  
38221, Volos, Greece

Leandros Tassiulas\textsuperscript{1,2}  
leandros@inf.uth.gr

\textsuperscript{2}Center for Research and Technology - Hellas  
Informatics and Telematics Institute  
6th klm Harilaou - Thermi Road  
57001, Thessaloniki, Greece

Abstract—Distributed systems and open networks gain more attention everyday, offering possibilities so far unattainable with classic networking technologies. Their main strength, openness and absence of centralized management and control, is also their main drawback, rendering them vulnerable to malicious intent. To this end, Reputation Systems have been used, providing the means to rate the network’s nodes according to their behavior and hopefully isolate nodes demonstrating misbehavior. In Reputation Systems an important issue is how to process, or aggregate, the various opinions provided by the network nodes about the others. So far, most such technologies use majority voting schemes, which are simplistic by nature and cannot easily produce clear results when opinion diversity is large. The goal of this paper is to provide a simple, yet efficient algorithm to enhance the Voting Schemes employed by Reputation Systems. The scheme can be used on top of existing Reputation Systems with few modifications.

Index Terms—Peer-2-Peer Networks, Reputation Systems, Trust, Voting Schemes, Wireless Ad-Hoc Networks

I. INTRODUCTION

Voting Schemes have attracted a lot of interest lately, mainly due to their application on Reputation and Trust Management Systems. The proliferation of distributed and decentralized systems, such as Wireless Ad Hoc networks Peer-2-Peer networks, has given rise to new possibilities and novel applications, ranging from rapidly deployed infrastructure-less wireless local area networks to world-wide file sharing or media delivery networks. Given their open nature, such networks are often targeted by mischievous users seeking to either gain a greater share of the network’s resources or to disrupt its normal operation, a fact which calls for new mechanisms to discourage malicious behavior. Reputation Systems are such mechanisms, as they allow nodes (or peers in the context of Peer-2-Peer networks) to evaluate the quality of others when deciding whether they should interact with them or not. The term interaction may refer to all sorts of communication, such as routing messages through them, exchanging files or asking for a service in general.

Reputation Systems essentially consist of two distinct mechanisms: One which quantifies the intuitive notion of trust or quality, a measure of the nodes’ behavior, forming the opinion of a node about the others, and a mechanism to aggregate those opinions when a node asks for the others to provide theirs about an unknown node. For the latter, Voting Schemes are commonly used. So far various formulations of Voting Schemes have been proposed in the literature, however most of them seem to employ only a simple majority voting rule or a simple weighted voting technique to determine the quality of the unknown node.

This work proposes a different approach, whereby nodes are asked to provide either a group of nodes they consider suitable to perform a task or, along with the group itself, a ranking of those nodes in the group, rather than their opinion about a specific node. Our aim is to investigate the implications, in terms of opinion aggregation, of asking other nodes to provide a group of nodes suitable for performing the task, instead of asking for other nodes to provide their opinion for all nodes individually. Naturally, there is the need to devise new aggregation rules for group selection, which is the most important contribution of this paper: We propose an iterative algorithm that improves the result of the simple majority voting technique.

This work has been financially supported by the Greek Ministry of National Education and Religious Affairs (E.P.E.A.E.K.-“HRAKLEITOS” grants), the Greek General Secretariat’s for Research and Technology Grants for Collaborations between Greece and Great Britain, and by the European Commission funded projects WIP (IST-4-027402) and NET-REFOUND (IST-034413)
is most probably the best known Reputation System and is a variation of Pagerank [2], the ranking system used to order the results of a search in Google. In Eigentrust peers ask for their acquaintances to provide an opinion for a given peer. These opinions (votes) are weighted by the trust that the peer asking for the votes places on those who provide the vote. The authors assume some sort of initial opinion that each peer has about the other. Also, they assume initial trust, which is a necessary condition for the algorithm to converge. Eigentrust makes use of the aforementioned simple weighted voting rule to aggregate the local trust values.

In [3] and [4] Jiang and Baras follow a similar approach. These two papers in essence confirm the results of Eigentrust, whereby the local trust value matrix must be irreducible and aperiodic for the algorithm to converge. The authors introduce the notion of headers, which are pre-trusted nodes that only vote for other nodes they fully trust. These headers are actually the same pre-trusted peers that Eigentrust deems required for the algorithm to function. Needless to say, in their work opinions are aggregated in a similar fashion to Eigentrust. The convergence of the algorithm, given the constraints regarding the local trust value matrix, is provided by the Perron-Frobenius theorem. A very good analysis on how this theorem is applied to such problems is presented in [5], where, among others, Pagerank is used as an example. Another important issue with Eigentrust is its round-based operation, i.e. after each round of interactions (file transfers), new trust vectors are computed and spread across the network. This implies that there is a large computational and communication overhead involved, as pointed out in [6].

Reputation Systems are also used in sensor networks for similar purposes. In [7] the authors use a Reputation System to rate the nodes of a sensor network according to the accuracy of the localization information they provide to others. The employed voting scheme is a simple majority voting rule. Other Reputation Systems for sensor networks do not even employ a voting mechanism, the nodes only trust the received opinion according to the trust they place on the reporting node. For example, in [8] both first hand observations and received opinions update the trust value of a node according to the Beta distribution. Actually, there have been a few other papers [9],[10] that use the Beta Distribution. Their basic idea is that as more observations (either first or second hand) arrive at a node, it is able to better predict whether a given node is going to misbehave or not. This probability is the reputation rating of a node about another. As mentioned earlier, these approaches do not use Voting Schemes to aggregate the various received (second hand) opinions, but rather they only accept them with an appropriate weight. However, these opinions may be discarded if they deviate too much from the opinion of the node requesting it (a deviation test is performed in [10]), which in turn may lead to much useful information going unexploited.

Voting Schemes have also been of central importance in social and economics sciences. In this context the basic aim is to analyze the aggregation rules, so as to make them elect candidates in an order that is as close as possible to their true ranking. While the voting rule itself is very important, the fundamental difference between such approaches and ours is that our aim is to find the best possible result, given the outcome of the voting algorithm used. A good paper describing the basic ideas of the social choice problem and presenting the most common voting rules is [11].

As mentioned earlier on, the voting mechanisms found in previous works are simple and minimal. Weighted average, commonly used in voting mechanisms of Reputation Systems, may lead to inconsistent and wrong reputation related decisions if there are diverse opinions. Our proposed scheme uses an Iterative Process to try and pinpoint the best possible nodes (or peers) from a voting process where the voters do not have a clear view of the best nodes, but rather a noisy perception of the true ranking. It has to be noted that the proposed voting mechanism can also vote for a group of nodes, which is useful in scenarios where a node seeks for a group of other nodes to perform a task on its behalf.

### III. PROBLEM FORMULATION

The goal of this paper is to analyze the effects of choosing a group of representatives (nodes or peers) to perform a task, for example, provide a file. Some advantages of electing a group of nodes have been briefly outlined in the introduction. Our approach differs from the classic Voting Schemes employed by Reputation Systems in the sense that a node asks from others to provide a group of nodes suitable for the requested service (for example a file download), instead of querying the others on the suitability of individual nodes before requesting a service from the nodes in question. This enables requests for services that need to be performed by a group and not individual nodes. These services could include emerging real-time services, such as collaborative video delivery. It is also very interesting to note that since the most reputable nodes are expected to be overloaded in classic Reputation Systems, choosing the trust-wise best nodes may actually mean receiving worse service. On the contrary, requesting a service from a group of nodes can balance the load among the providers of this service. In essence, what we are looking for is for the group best suited to perform the task, which of course depends on many factors, trust being one and load being another. Simply put, our formulation allows nodes to submit their opinion an a per-case basis, generalizing the notion of quality and taking it out of the strict boundaries of trust.

The following paragraphs describe the basic formulation of the scheme, the employed noise models and the proposed algorithm.

#### A. Node Quality

To begin with our formulation, let us assume that there are $N$ nodes aiming to select $M << N$ nodes to perform a task, such as downloading a file. Let us also assume that there is an objective quality of the nodes as potential representatives to perform this task. The objective quality shall be denoted as $q_i$, while the vector $Q = [q_1, q_2, \ldots, q_N]$ is the objective quality vector. The goal is to select $M$ nodes $i_1, i_2, \ldots, i_M$ so as to maximize

$$\sum_{m=1}^{M} q_{i_m}$$

(1)
The obvious solution is to select the $M$ highest quality nodes. However, an important aspect of the proposed model is that the objective qualities are not known by the nodes, as is the case with various reputation and ranking systems, such as PageRank. The objective qualities can be seen as equivalent to the global trust values of Eigentrust. However, the purpose of the proposed scheme is not to provide a method for nodes’ local opinions to converge to the global trust values, but an efficient voting procedure that aggregates the votes (opinions) submitted by others to derive more accurate local opinions. How these values are further used and processed depends on the Reputation System.

As mentioned earlier, it is very important to note that network nodes might be completely unaware of the quality of another node. That is, there usually exists high diversity in opinion about a node, or group of nodes, especially in the early stages of the networks’ deployment. Therefore, the problem that has to be addressed is to identify the voting mechanisms that are more effective in maximizing the quality of the selection given the fact that users might have noisy or distorted view of the quality function.

B. The Voting Mechanisms

In the proposed scheme, two voting mechanisms are considered:

- A “Majority Voting” mechanism, whereby each node votes for $L$ representatives ($L \leq M$), giving them one vote each.
- A “Ranking Voting” mechanism, whereby each node not only votes for $L$ representatives ($L \leq M$), but also rates them by giving $L$ votes to the best, $L - 1$ to the second best and so on.

Obviously there are a lot of other voting mechanisms that can be employed, however these two were selected on the basis that the Majority Voting mechanism does not ask nodes to rank the alternatives, but rather to indicate which ones they consider suited for the given task. On the contrary, the Ranking Voting mechanism requires nodes to provide a relative ranking of the voted representatives, giving them a weight according to their rank. In fact, this mechanism resembles the Borda rule [11], whereby each voter assigns $m - 1$ votes to the best representative, $m - 2$ to the second best and so on. Hence, these two voting mechanisms capture the difference between submitting rankings and submitting only votes of confidence to the alternatives.

C. Node Ignorance and Noise Models

An important issue of the proposed scheme is how to model the nodes’ ignorance. As mentioned in the previous paragraphs, each node has a noisy perception of the other nodes’ real quality. Let us assume that node $i$ perceives a noisy version of the quality function and denote the quality of node $i$ as perceived by node $j$ by $q_{ij}$. Therefore, the perceived quality vector for node $j$ is $Q_p = [q_{1j}, q_{2j}, \ldots, q_{Nj}]$. The perceived quality shall be

$$q_{ij} = q_i + \alpha g_j n_{ij}$$

In equation 2, $g_j$ is the ignorance factor of node $j$ and $n_{ij}$ is white noise, while $\alpha$ is a multiplier aimed at adjusting the noise magnitude. Node $i$ votes for the nodes with the largest $q_{ij}s$. Equation 2 is for the first class of nodes, called Normal Nodes. We shall define a second one later on, termed Expert Nodes, which follows a slightly different noise model.

Node ignorance, as presented here is in essence a measure of the confidence a node has about its opinions on others. Other works usually assume that nodes completely trust themselves. In this work the nodes’ competence to assess the quality of others comes into play, expressed through the ignorance factor. It is also important to note that, as is the case with the objective quality, we consider the ignorance factor a quantity unknown to nodes. Another way to view the ignorance factor is to consider it as a measure of the voting quality of the nodes. Indeed, the more ignorant a node is, the more difficult it is for it to cast a ballot that is a good estimate of the true ranking. In the model presented earlier the objective quality and the ignorance factor are considered independent of each other. That is, a good quality node is not necessarily a good voter and vice versa.

At first sight, considering the two qualities independent seems a natural choice: a very good node may not necessarily know which other nodes are best suited for a given task. However, it should also be noted that, intuitively, a good node for a given task may also be able to identify other nodes also well-suited for that task, the same way a good professional can better judge the quality of other professionals. Obviously, this extra knowledge stems from long-term experience, expressed through the interactions it had with others. Hence, we also introduce a second class of nodes, called Expert Nodes. Their noise model is defined by the following equation:

$$q_{ij} = q_i + \alpha(1 - q_j)g_j n_{ij}$$

The expert nodes model partially captures the intuition that the opinion of good nodes is less noisy and thus more accurate and important. From another point of view, this model expresses more knowledgeable and experienced nodes.

We assume that the network consists of both normal nodes and expert nodes. As will be made clear later on, expert nodes are in no way different to the other nodes, except that they have a more clear view of the true ranking. They do not have to be special nodes, being for example fully trusted by the others, or play a more important role, nor do they have to coincide with cluster heads or other special-purpose network entities. Given the required experience for a node to become expert, we only assume the existence of a small percentage of experts.

D. The Iterative Voting Algorithm

As stated in the previous paragraphs, the aim of this work is to present a mechanism that has the property of identifying a group of nodes to perform a task that is as close to the objective best as possible, especially in cases where there is a lot of diversity in opinion. In other words our goal is to identify the best possible group through a set of different and often contradicting opinions. Let us assume that each node submits a ballot with $M$ nodes (its votes). The ballot is derived using the algorithms presented in section III-B, however it is
obvious that the ballot can be derived using any other voting mechanism. Let us also denote by \( w_i \) the votes that each node receives. The \( M \) nodes with the highest \( w_i \)'s are selected. Now, let us assume the following \( N \times N \) matrix, called the vote matrix, which represents how the nodes voted. Its elements \( v_{ij} \) denote the votes that node \( j \) gets from node \( i \). Hence, the sum of each column represents the number of votes that each node received. It is obvious that the votes \( w_i \) are given by equation (4).

Let us denote the vote matrix as \( V \). In an effort to best exploit the information contained in \( V \), we propose refeeding the initial votes \( w_i \) into the vote matrix. Hence, our proposed algorithm is the following iterative process:

\[
\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} \mathbf{V},
\]

where \( i \) is the iteration. Our goal is to investigate under which circumstances this heuristic algorithm has the ability to extract more information from a given vote matrix and select a group as close as possible to the objective best.

IV. EXPERIMENTAL EVALUATION

A. Assumptions and considered metrics

Most approaches model trust and/or quality as a quantity in the range \([0, 1]\) and hence, for reasons of compatibility, the range of the Objective Quality and the Ignorance Factor is also \([0, 1]\). An important note is that from equations 2 and 3 it is quite obvious that the perceived qualities can fall outside the aforementioned range. However, for consistency reasons these perceived qualities are bounded to fall into the range \([0, 1]\). The noise is assumed to be white Gaussian noise \( N(0,1) \). Strictly speaking, there may be correlation among opinions (right or wrong), however not only this is partially captured by the ignorance factor, but also cases of extreme correlation will be evaluated in the liar scenarios sub-section. Regarding the aforementioned ranges, various combinations were tested, only to find out that the ranges have no effect on the scheme, except of course in extreme cases which have no practical value. As far as the convergence is concerned, the algorithm converges pretty quickly, most of the times after three or four iterations, and therefore it was not considered necessary to measure it as a performance metric.

Given our original goal, which is to test the functionality of the algorithm especially in cases of extreme opinion diversity, the basic metric we consider is the quality of the elected groups. To measure the performance of the scheme we compare the quality of the elected group (or single node) after the application of the Iterative Voting Process against the group (or single node) initially elected at the first voting round and against the randomly selected group (or single node). Obviously the maximum quality is given by the simple multiplication \( \text{ElectedGroupSize} \times \text{MaximumNodeQuality} \).

As far as noise is concerned, we consider two basic scenarios: A normal (or low noise) mode for \( \alpha = 1.0 \), whereby the noise component of the nodes' perceived quality is one order of magnitude lower than the objective quality component, and a high noise mode for \( \alpha = 10.0 \), whereby the the noise component is of the same order of magnitude as the objective quality component. Obviously, other values for \( \alpha \) could be used, however values above 10 distort the opinions to the extend that voting becomes practically random.

An important issue has to do with the qualities themselves: The proposed algorithm focuses solely on aggregating the opinions (i.e. the perceived qualities). Therefore, how these qualities are derived or how they evolve in time is of no importance to our scheme.

Our test includes both the evaluation of group selection scenarios as well as the scenario of selecting one node to perform a given task. This is again for reasons of compatibility with Reputation Systems which use opinion aggregation mechanisms to estimate the trustworthiness of a given node. Obviously, when voting for one node Majority Voting and Ranking Voting are exactly the same.

B. Basic Evaluation of the System

Our basic evaluation scenario assumes the existence of a percentage of expert nodes, that can distinguish the good ones for a task from the bad ones, based on their own quality. It is quite important to test the performance of the scheme against the percentage of Expert Nodes, which will depict how many experts are needed for the proposed algorithm to perform adequately. Figure 1 plots the quality of the Elected Node against increasing percentage of expert nodes and group size, for the low noise mode (\( \alpha = 1.0 \)), while Figure 2 depicts the high noise scenario (\( \alpha = 10.0 \)).
Indeed, the scheme is quite robust against liars. To evaluate it, we define two types of colluding liars:

- **Random Liars.** These colluding liars choose to vote for the same node or group of nodes, regardless of its quality. They are called random liars simply because the group or single node they support is chosen at random.
- **Expert Liars.** The same way we assume the existence of a group of expert liars, who know the true ranking of the group of nodes and vote for the worst nodes or groups. It has to be said that this is an extreme scenario, however it serves well the purpose of displaying the robustness of the scheme.

The following experimental evaluation will depict up to which point the scheme can tolerate liars. It should be stressed that liars are a very severe problem, since the openness of Wireless Ad-Hoc Networks, Peer-2-Peer and autonomic networks in general allows for users to disconnect and reconnect using different identities, therefore providing a malicious user the ease to lie without consequences. This is often referred to as the **Sybil Identity** problem. To probe further on this issue, the interested reader can look into [13] and [14].

In all cases we assume that the percentage of expert nodes is 10%, while the scenario is a high noise one, that is \( \alpha = 10 \) for the honest nodes. We make this choice to show that our proposed algorithm can withstand large percentages of colluding liars even when opinions of honest nodes are distorted. We assume that the noise magnitude of expert nodes is on average half the magnitude of the normal nodes', which means that expert nodes have also quite noisy perception. Needless to say, liars make their choice regardless of noise.

First of all the case of Random Liars shall be evaluated. Since the colluding liars support a random node or group, it is expected that as their percentage increases the voting procedure shall produce on average a result very close to the random choice. Figure 7 depicts the scenario of voting for one node out of 200. It is clear that the Iterative Voting Process shows very good robustness against colluding liars, as it greatly enhances the result of the initial Majority Voting. As expected, for really large percentages of colluding liars the performance of the Iterative Process drops, since the information the
experts and the honest-but-noisy normal nodes have is not enough to estimate the true ranking through voting. Same conclusions can be drawn from the next Figure (8), which depicts the scenario of voting for a group of 5 nodes out of 300. All other parameters are kept the same as in the previous case. It is quite interesting to note that Ranking Voting shows a more stable behavior compared to Majority Voting. Majority Voting may yield better results for relatively low percentages of liars, however there is a sudden drop in its performance, whereas Rankings exhibit a more stable performance. This is a property of Majority Voting schemes in general, especially in cases where there are many alternatives and high opinion diversity: a 35% to 40% of colluding voters is enough to completely control the outcome. This is also mentioned in Eigentrust [1], where the authors claim that since their scheme uses majority voting, “large percentages of colluding malicious users will be able to influence the assignment of global trust values values in the network”, to put it in the authors’ own words.

Ranking Voting, on the contrary, exhibits this more stable behavior due to the inherently richer information that the ballots convey. Even in cases of high opinion diversity there is still enough information in the honest voters’ ballots to counter the effect of the colluding liars, hence the result degrades gradually as the percentage of colluding liars increases. The Iterative Voting Process is relatively steady for both Majority Voting and Ranking Voting. Again it is clear that the Iterative Process shows good performance for even quite large percentages of liars, exploiting the information contained in the ballots to the fullest possible extent.

It is even more interesting to evaluate the robustness of the scheme against Expert Liars. As stated earlier, this is a quite extreme scenario, however if the scheme is robust against such liars, it can handle most real-case scenarios. Figure 9 shows the schemes performance when voting for one node out of 200. It has to be noted that as the percentage of expert liars increases, the quality of the initially voted node or group closes zero, which is expected since expert liars vote for the worst nodes(s). All other parameters have the same values, e.g., we assume high noise (α = 10), 10% of the nodes are experts and their noise has on average half the magnitude of the other nodes’ noise. It is clear that the up to 25% of expert liars the initial voting selects a slightly better node than choosing at random, while the Iterative Voting Process improves the selection even further. However, for larger percentages Majority Voting completely fails. Instead, the Iterative Voting Process manages to withstand a bit larger Expert Liar percentages. The results for voting for a group are similar, as depicted in Figure 10. As in the case of random liars, it is evident that when Ranking Voting is used the performance is more steady and in this case the Iterative Voting Process yields better results for larger percentages of expert liars.

Those Figures clearly depict the tolerance that the scheme displays against colluding liars. Even in the Expert Liars scenario, the scheme can provide a better result than random choice for up to 50% - 55% of expert liars for group selection scenario, and up to 35% for the single node selection scenario. The results are much better for the random liars scenario (more than 70% tolerance), which is most common. What's also very interesting is that the Iterative Voting Process with Ranking Voting is more robust for large percentages…
of Expert Liars than the Iterative Voting Process with Majority Voting, as is the case with Random Liars. This can be attributed again to the fact that rankings convey more information than majority voting ballots and in the case of large liar percentages, where clear opinions are sparse, this information is better exploited by the Iterative Voting Process. Hence, for large percentages of colluding liars the Iterative Voting Process makes better use of Ranking Voting than of Majority Voting.

V. RATIONALE BEHIND FORMULATION AND SCHEME ADVANTAGES

Our basic aim is to work with rankings, not only because of their importance in Voting Schemes, but also because quality differences are most of the times subjective and difficult to measure: In general, even if two nodes $a$ and $b$ provide the same quality value for a third node $c$, it does not necessarily mean that $a$ and $b$ share the exact same opinion about $c$. This makes the reputation and trust and reputation evidence subjective, uncertain and incomplete, as mentioned in [3], and places even more importance on rankings, since they have the property of bypassing the subjectiveness of reputation and trust measures. To put it another way, majority and ranking voting provide an abstraction from actual qualities, which in essence means that nodes are given the opportunity to use whatever mechanism they wish in order to derive the qualities. This is completely in line with our basic target to select a group of nodes best suited to perform a given task, with “best-suited” being scenario-dependent.

The noise models and the expert node formulation provide a very important advantage: Expert Nodes are not special nodes performing any special task. Other works, most notably [1] and [3], assume the existence of pre-trusted nodes or headers respectively, which are special nodes: others know them and are required to place at least some trust on them. On the contrary, we place no special characteristics on the Experts, other than the more clear view of the true ranking. This provides Expert Nodes anonymity: any node can be an expert at any time, as long as it has a better view of the true ranking. It does not have to broadcast this fact nor behave any differently. In contrast, pre-trusted peers and headers are well known, hence, they can be targeted by malicious users. Our scheme does not require for the Experts to be known, thus protecting them to a certain extent from malicious intent as only if an Expert gets compromised can malicious nodes learn about the identity of other experts.

VI. CONCLUSIONS AND FUTURE WORK

This work presents an alternative Voting Mechanism, suitable for Wireless Ad-Hoc Networks and Peer-2-Peer networks, that employs an Iterative Process to distinguish the best possible representatives out of a set of diverse opinions (ballots). The scheme converges really fast and, especially in cases of large opinion diversity, it has the ability to improve the result of the initial voting mechanism. The scheme can be used on top of any reputation system, as it does not have any special requirements on the format of the reputations.

However, we are currently examining a few extensions to the scheme. Although voting has essentially to do with rankings, the proposed formulation allows for the consideration also of the relative difference in quality of the nodes. Simply put, the modeling of the node quality allows nodes to consider how much better a node is from the others, which in turn makes it possible to cast ballots reflecting this difference in quality.

Another important difference of our scheme with classic social sciences Voting Schemes is that in Wireless Ad-Hoc Networks and Peer-2-Peer systems the voters and the candidates coincide. As pointed out in [12], when these groups coincide, the transitive effects of voting should be considered: intuitively, if a good node $a$ thinks highly of node $b$, then the opinion of $b$ on $c$ should have more importance. This has only been partially incorporated in our scheme through the expert node model. We are looking into ways to exploit these transitive effects.

Finally, we plan to deploy the scheme in a real reputation system and further test it, mainly to diagnose any possible performance problems.

REFERENCES


