Reputation-Based Dynamic Trust Evaluation Model for Multi-Agent Systems Based on Service Satisfaction

T. Chalama Reddy¹, R. Seshadri²

¹Department of CSE, Narayana College of Engineering, Nellore, India
²Computer Center, Sri Venkateswara University, Tirupathi, India

Abstract—Trust and security issues have become critically important with the fast expansion of multi-agent system. Many computer applications such as the E-business systems, Agent-to-Agent, Grid and Semantic Web can be viewed as multi-agent systems which are open, anonymous and dynamic in nature. By its nature of multi-agent systems, malicious agents are always seeking to spread malicious content in the multi-agent network. To address threats of untrustworthy agents and their malicious behavior in the multi-agent system, many trust/reputation-based models have given solutions, but they fail to properly evaluate trust when malicious agents start to behave in an unpredictable way. Another aspect of trust evaluation models is becoming critical for sustaining good quality of service, because different trust evaluation models use different standards in its opinion. In this study, we propose a reputation-based dynamic trust evaluation model to help evaluating the trustworthiness and predicting the feature behavior of agents. There are two main features of our model. First, we use trust parameter amount of satisfaction as reputation value to represent the quality of service offered by an agent. The amount of satisfaction; a service requestor agent has about a service provider agent. The amount of satisfaction is a major parameter to measure the trustworthiness of service provider. Second, we suggest suitable trust mechanisms that use personal experience or the experiences of others, possibly combined to help agent in evaluating the trustworthiness of another agent by considering major trust parameters such as the amount of satisfaction of transaction, the size of transaction, trust Decays with time of transaction, context of transaction, successful rate of transactions, and the feedback provided by the other entities in the past. Simulation results show that the model can evaluate accurately the agents’ trust degree and effectively control of malicious agents attack. When the malicious agents have a higher proportion in the system, it can effectively detect sudden strategic alteration in malicious behavior, and still keep high success rates. The experimental results illustrate the effectiveness of the proposed approach.

Keywords—reputation-based trust, satisfaction function, malicious agents, trust parameters and mechanisms, service attributes quality of service (QoS).

I. INTRODUCTION

With the development of computer and communication technology, the network environment has changed from the relatively closed network to open network environment which has become into a publicly accessible by a large numbers of dynamic users. A multi-agent system is one in which many agents run concurrently, communicating among them and working toward either individual goals or a common objective [9]. Many computer applications such as the Agent-to-Agent[6], E-business systems, Grid[8] and Semantic Web[7] can be viewed as multi-agent systems, as individual components in each system are both autonomous and flexible in their actions. Such characteristics of multi-agent system make it an ideal environment for attackers/malicious agents to spread malicious content. One of the solutions to minimize the threats is to evaluate the trust and reputation of interacting agents. Reputation and trust is crucial for success of these agents.

Due to existence of fraud and unreliable services, agents faces how to identify and choose efficient service with increasing choice of chance. With these research problems in mind, we propose reputation-based dynamic trust evaluation model that explicitly define reputation as the QoS provided by agent. Every time, agent P has received a service from Q (i.e., a transaction from Q to P), P updates the reputation value of Q based on the perceived QoS of this transaction. In reality, the QoS measured in each transaction with agent Q could be different, but it is realistic to assume that agent usually consistently provides its service around some quality level within long time window. Our reputation model tries to make calculated reputation values come together in a short time which reflects the average level of QoS provided by each agent. Obviously, it is not efficient to update reputation values only through direct transactions.

Reputation systems provide a technique for building trust through social control without trusting third parties. Reputation based trust models are basically divided into two categories based on the way information is aggregated from an evaluator’s perspective [16].
They are “Direct/Local experience model” and "Indirect/Global reputation model” where direct experience is derived from direct encounters or observations (firsthand experience) and indirect reputation is derived from inferences based on information gathered indirectly (secondhand evidence such as by word-of-mouth). So, in case of global reputation models [17] an agent aggregates feedback from all the agents who have ever interacted with the target agent to obtain a better decision. However, global reputation models are much more complex to manage than local experience models as malicious agents have the opportunity to provide false feedbacks. To address threats of untrustworthy agents and their malicious behavior in the multi-agent system, we consider decentralized reputation-based dynamic trust mechanism. In this proposed model, we consider both direct and indirect experiences to judge another agent of multi-agent network to predict their likely behavior of transactions, thus reducing the risk of transaction failure to avoid losses.

This paper focuses on reputation and trust management where multi–agents work. The proposed system is reputation-based dynamic trust evaluation model which can effectively detect sudden strategic alteration in malicious behavior. The rest of this paper is organized as follows: study and analysis of the related work in section 2. The protocol we propose is described in Sections 3. Results of the simulation experiments testing the protocol’s effectiveness are presented in Section 4. Section 5 concludes the paper with a discussion of the future work.

II. RELATED WORK

There are a few existing online reputation systems such as the feedback of e-bay, yahoo auction, and auction universe. Most of these systems use the single trust parameter of feedbacks as reputation measure, which often fails to handle the trustworthiness of users effectively.

One of the earliest works in this area is the protocol by Aberer and Despotovic [1] which aims to identify dishonest agents by a complaint-based system. A shortcoming of this protocol is that it maintains only the negative feedbacks, providing no means for a trustworthy agent to be distinguished from a newcomer. The trust evaluation is also rather simplistic, classifying every agent either as trustworthy or untrustworthy. Moreover, maintenance of a P-Grid, architecture is required on top of the existing P2P structure.

EigenTrust model puts forward the distributed computing method based on DHT[9], using the trust algorithm that calculates global trust value by direct trust value with trust transfer characteristics, but this model exists convergence problem, and have the higher communication cost and relative value of global reputation, it makes cannot directly judge whether the node is credible from the global reputation value. Zhou etc. [10] improve the Eigen Trust model for the aspects of determination of credible agents and the speed of trust iteration convergence, and puts forward the Power Trust model. In resistance of the malicious agents aspect, this model stronger than Eigen Trust algorithm, but in the calculation of trust value, it doesn’t consider the influence of the trust value on trading volume, and has not to make punishment for malicious behavior.

Agent Trust model [11, 12] uses the confidence factor to synthesize local reputation and global reputation, allow for many factors which could influence measurement of trust, and can deal with false evaluation very well, but Agent Trust model does not give measure method of trust factor and method for determining confidence factor.

Trust has been addressed at different levels by many researchers. In [10] behavior based trust model is proposed in which trust is computed by taking into account the direct trust, reputation and decay function. The various parameters like number of transactions for the different types of resources are not considered.

Jøsang etc. propose the concept of multiple-valued logic[17, 18] based on the subjective logic, with Dirichlet multidimensional probability distribution [19] as the foundation, allowing to have different grades evaluation, which can be used for the calculation of reputation value, provide more flexible platform for design of reputation system. But the model only uses direct evaluation result to calculate trust degree, without considering recommendation trust, and how to identify malicious evaluation and give it punishment.

Today, the most trusted models evaluate level of trust about service provider agents. In the human society, when people choose and buy a commodity, the first concern is whether the function of the goods to meet their needs, and then consider other properties of the goods, such as price, the brand value, appearance, usability, after-sale service and so on, finally purchase the items according to preferences oneself and comprehensive evaluation of each property.
Purchased items, if the properties of the goods are consistent with vendors claimed, the seller is considered as trustworthy.

In view of the above questions, we propose a reputation-based dynamic trust evaluation model that evaluates multiple service attributes provided by service provider based on personal interest, and combines with the trust value, decide whether to trade finally; after the transaction, service requester calculates amount of satisfaction according to actual QoS and service provider's own QoS claimed to judge the credibility of service provider, then gives corresponding rewards and punishment, and trust update.

III. PROPOSED REPUTATION-BASED DYNAMIC TRUST EVALUATION MODEL

The main problem behind of reputation-based trust evaluation model is that while gathering reputation values from other agent in multi-agent network, some malicious agents may collude to provide fraud recommendations for increasing or decreasing someone’s reputation. To solve this problem, we propose a reputation-based dynamic trust evaluation model that measures the trustworthiness of agents when propagating reputation values. It should be noticed that the trust mechanism is used to detect malicious agents that report fraud reputation values, not for detecting agents that provide fraud services. The agents providing fraud services can be detected when providing low QoS. This trust evaluation model predicts what they have behavior in future transactions through collection ,analysis of historical information of agent behavior, thus could select an agent which has a highest trust value, and reduce the risk of transaction failure to avoid losses. In this model, we take into account some of trust parameters to determine suitable trust mechanisms that use personal experience or the experiences of others, possibly combined to make trust decision about an agent. We have used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing the trust of agents.

In this model, we use decentralized trust approach in which each agent maintains a table with one entry for each agent in this system. There is no central data base. So each agent in this network updates its records according to result of every transaction that does. The table 1 shows trust table for Sr1 in which one record for each agent.

Each record consists of several fields such as service requester, service provider, time of previous transaction, and amount of service satisfaction of current transaction, direct trust value of previous transaction, direct trust value current of transaction, indirect trust value of current transaction, and expected trust value of current transaction. Table1: shows trust table of service requester /service recommender.

<table>
<thead>
<tr>
<th>Service requester</th>
<th>Service provider</th>
<th>Time of last transaction</th>
<th>Amount of service satisfaction</th>
<th>Direct trust value of previous transaction</th>
<th>Direct trust value of present transaction</th>
<th>Indirect trust value of present transaction</th>
<th>Expected trust value of current transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sr1</td>
<td>Sp1</td>
<td>0</td>
<td>0.783</td>
<td>0.786</td>
<td>0.789</td>
<td>0.696</td>
<td>0.789</td>
</tr>
<tr>
<td>Sr2</td>
<td>Sp2</td>
<td>1</td>
<td>0.678</td>
<td>0.803</td>
<td>0.823</td>
<td>0.703</td>
<td>0.697</td>
</tr>
<tr>
<td>Sr3</td>
<td>Sp10</td>
<td>10</td>
<td>0.689</td>
<td>0.815</td>
<td>0.835</td>
<td>0.615</td>
<td>0.698</td>
</tr>
<tr>
<td>Sr5</td>
<td>Sp11</td>
<td>11</td>
<td>0.875</td>
<td>0.841</td>
<td>0.861</td>
<td>0.741</td>
<td>0.879</td>
</tr>
</tbody>
</table>

3.1 Reputation and Trust

To address threats of untrustworthy agents and their malicious behavior in the multi-agent system, we have considered reputation–based dynamic trust mechanism. This reputation system provides a technique for building trust through social control without trusting third parties. Reputation is the opinion of one agent about the other. Trust is a derivation of the reputation of an agent. The biggest difference between trust and reputation is that reputation is an objective concept, and trust is a subjective concept. Trust can increase or decrease with further experience (interactions or observation). New experiences are more important than old ones since old experiences may become obsolete or irrelevant with time passing. The trust value ranges from 0, which means complete trust, to 1, which means complete trust. There are two different approaches to handle the trust. The first approach is a centralized trust approach that has a solid authentication set of rules where decisions are made on some digital or logical rules called policy based trust mechanism. Second approach is a decentralized trust approach where reputation assembled and shared in distributed environment called reputation based trust mechanism. In this paper, we used the second approach.

3.2: Trust parameters

The agent’s relationships are established in pair wise interactions during which certain information or service is exchanged.
To help agent in evaluating the trustworthiness of another agent, the problem is to essentially determine a suitable trust mechanism or function that computes a trust value from information or input data that is relevant to trust decision a service requester agent is going to make. In order to determine a suitable trust mechanism, we need to determine the trust parameters. We have identified five important trust parameters such as the amount of satisfaction, number of transactions, and credibility of feedback, transaction context factor and trust Decays with time to assess the trust of agent.

Related definitions
Agent: refers to a service requester or service provider or service recommender.
Service requester: A service requester is an agent which requests for service.
Service provider: A service provider is an agent that provides the service.
Service recommender: A service recommender is an agent who recommends the service to service requester.

Let \( X = \{ x_1, x_2, \ldots, x_N \} \) be a set of \( N \) agents in the system. In an interactive, each agent may play one of three kinds of roles: Service Requester agent (evaluator), Service Provider agent, and Service Recommender agent.

In this model, evaluator is an agent who evaluates the trustworthiness of other agents based on social interactions. Here we assume that agent \( x_i \) (called the trustee agent) needs to calculate the trustworthiness of another agent \( x_j \) (called the target agent).

I. The amount of satisfaction:

Amount of satisfaction is an ambiguous and abstract concept. The state of satisfaction depends on number of factors which consolidate as psychological, economic and physical factors. The amount of satisfaction reflects how well this agent has fulfilled its part of the service agreement. In this paper, we introduce satisfaction function, \( \text{Sat}_i(x, x_j) \) that measures the degree of amount of quality of service (QoS) satisfaction that a service requestor (truster) \( x_i \) fulfills its request by given service provider agent \( x_j \) (called the trustee agent).

Before service of every transaction, the service requestor \( x_i \) computes QoS based on various attributes (such as resource running speed, reliability, easily use etc.) values of QoS provided by service provider \( x_j \), according to formula.

\[
\text{Sat}_i(x, x_j) = \sum_{m=1}^{M} w_m \ q_{attrm} x_j
\]

Where \( q_{attr1}, q_{attr2}, \ldots, q_{attrM} \) are various service attribute values of QoS service provider claimed to provide and its values must be within range from 0 to 1. Service providers can update QoS attribute value themselves according to their immediate service performance and the feedback of the service requester.

Where \( w_1, w_2, \ldots, w_M \) are Weight factors based on importance of service attribute. Their values must be within range from 0 to 1. The sum of all weight factors must be 1, and it is defined as follows:

\[
0 \leq W_i \leq 1, \quad \sum_{i=0}^{M} W_i = 1 \quad \text{then } W_i \text{ is called for the weight factor of } i\text{-kind attribute.}
\]

After completion of the \( n \)-transactions in \( t \)-th time interval, the service requestor \( x_i \) computes QOS based on various attribute value of QoS provided by service provider \( x_j \), according to following formula:

\[
Q_{x_i,x_j}(t) = \sum_{m=0}^{M} W_m \ q_{attrm} x_i, x_j
\]

Where \( q_{attr1}, q_{attr2}, \ldots, q_{attrM} \) represents various attribute value of QoS actually provided by service provider \( x_j \) to service requester \( x_i \) and its values must be within range from 0 to 1 and \( w_1, w_2, \ldots, w_M \) are same as in before transaction.

The difference value of the QoS, i.e, service requester \( x_i \) actually received after the end of \( n \)-th transaction in \( t \)-th time interval and service provider \( x_j \) claimed to provide, is defined as according to following formula,

\[
\text{diff}_Q(Q_{x_i}^{\text{self},t}, Q_{x_i}^{x_j,x_i}) = \frac{Q_{x_i}^{x_j,x_i}-Q_{x_i}^{\text{self},t}}{Q_{x_i}^{x_j,x_i}}
\]

We have used the difference value of QoS to define the amount of satisfaction as follows

\[
\text{Sat}_i(x_i, x_j) = \text{diff}_Q(Q_{x_i}^{\text{self},t}, Q_{x_i}^{x_j,x_i})
\]

The value of \( \text{Sat}_i(x_i, x_j) \) exceeds a certain threshold (\( E \)), we think that the \( x_j \) is unbelievable and reduce the value of its reputation. Otherwise, the \( x_j \) declared true characteristics of its service, the \( x_j \) is believable, its trust value should be increased. According to following formula, the service requester judges whether it is satisfied for its own requirement or not.
ii. Number of transactions

With the need of transactions, i.e. some agents have a higher transaction frequency than other agents; the trustworthiness of agent is not captured fairly when a simple aggregation of feedbacks is used to model the trustworthiness of agents without taking into account the number of transactions. When the number of successful transactions increases between two agents, the higher degree of mutual trust will be built between two agents. This can prevent some malicious agents as honest in the beginning of transactions between two agents. For this reason, we introduce number of successful transaction function, σ which is defined as follows:

\[ \sigma_n(x_i, x_j) = \sqrt{\text{size}} \]  

Where s and n represent successful and total number of transactions between two entities \((x_i,x_j)\) respectively.

iii. Credibility of feedback

Feedback credibility is used to measure the degree of accuracy of the feedback information that the recommending agent provides to the evaluator. Normally it is assumed that good agents always provide true feedback and malicious agents provide false feedback. However, this is not always the real scenario as good agents might provide false feedbacks to their competitors and malicious agents might occasionally provide true feedbacks to hide their real nature. So feedback credibility is needed to determine the reliability of the feedback. During trust evaluation, feedbacks provided by agents with higher credibility are trust worthier, and are therefore weighted more than those from agents with lower credibility.

iv. Transaction Context Factor

Transaction context is another important factor when aggregating the feedbacks from each transaction as transactions may differ from one another even within the same business community.

For example, if a community is business sense, the size/value of a transaction is an important context that should be incorporated in the trust metric to weight the feedback for that transaction. It can act as a defense against some of the subtle malicious attacks, such as a seller develops a good reputation by being honest for small transactions and tries to make a profit by being dishonest for large transactions.

The transaction context parameter \(C_n(x_i, x_j)\) expresses the context of the \(n\)-transactions in \(t\)-th time interval, for example, the size or value (small or large) of transaction. The transaction context parameter is defined as:

\[ C_n(x_i, x_j) = e^{-\frac{1}{\text{size}}} \]  

\((\text{size} > 0)\) is used to adjust the amount of the transaction that can influence on the direct trust.

v. Trust Decays with time

If an agent does not interact with the network for a long period, the evaluation of its trust should degrade gradually. So, we have to apply a decay function on direct and indirect trust. The decay function is given as follows:

\[ \text{Dec}_n(x_i, x_j) = e^{-\lambda \Delta t} \]

\(\Delta t = t_c - t_p\), where \(t_c\) is current transaction time and \(t_p\) is previous transaction time.

Here \(\lambda\) is the decay constant and it controls how quickly the value will diminish to zero. \(\Delta t\) represents the interval between the current interaction and the last interaction.

3.3: Trust Mechanisms

In order to determine a suitable trust mechanism, we need to determine the trust parameters such as the amount of satisfaction, number of transactions, and credibility of feedback, transaction context factor, and trust decays with time. Based on these trust parameters, we have determined trust mechanisms such as direct trust, indirect trust and expected trust for evaluating trustworthiness of agent.

3.3.1 Direct trust mechanism

Direct trust also known as local trust represents the portion of trust that an agent computes from its own experience about the target agent. In other words, it keeps record of the direct trust level of all the transactions an agent makes with another agent. However, instead of storing all of the transaction history we have defined an expected trust for evaluating trustworthiness of agent.
This greatly reduces the storage overhead and at the same time assigns time relative weight to the transactions. Let DT\_n\_t(x_i, x_j) represent the direct trust that agent x\_i has upon agent x\_j up to n transactions in the t-th time interval.

The following formula shows the direct trust degree after completions of n-transactions in t-th time interval, the x\_k over the x\_j :

\[
DT\_n\_t(x_i, x_j) =\begin{cases} \sigma_n \text{Sat}_n(x_i, x_j) \times C_n(x_i, x_j) \times \text{Dec}_n(x_i, x_j) & \text{if } n=1 \\ \alpha DT\_n\_t\_1(x_i, x_j) + \sigma_n (1 - \alpha) \text{Sat}_n(x_i, x_j) \times C_n(x_i, x_j) \times \text{Dec}_n(x_i, x_j) & \text{if } n \geq 2 \end{cases}
\]

The parameter \( \alpha \) \((0 \leq \alpha \leq 1)\) is a historical factor, it expresses the proportion of the previous trust value when calculate the direct trust degree. To assess the direct trust of agent x\_i, the x\_j have to consider some of important trust parameters such as the amount of satisfaction Sat\_n(x_i, x_j), number of transactions \( \sigma_n \), transaction context C\_n(x_i, x_j), and decay characteristics with time Dec\_n(x_i, x_j), and the direct trust degree in the history DT\_n\_t\_1(x_i, x_j).

\[3.3.2: \text{Indirect Trust mechanism:}\]

Indirect trust also referred as recommendation is computed from the experience of other agents. The evaluating agent then aggregates recommendations from other trusted agents along with the feedback credibility of the recommenders.

Let, IT\_n\_t(x_i, x_j) represent the indirect trust that agent x\_i computes about agent x\_j.

\[IT\_n\_t(x_i, x_j) = \sum_{k=1}^{n} FC\_n(x_i, x_k) \times DT\_n\_t(x_k, x_j) \times Dec\_n(x_k, x_j)/N\]

Where X represents the set of agents who have ever interacted with agent x\_j, and N expresses the number of the recommended agents. FC\_n(x_i, x_k) expresses the feedback credibility, the x\_i over the x\_k. DT\_n\_t(x_k, x_j) expresses the direct trust degree, the x\_k over the x\_j. If an agent, x\_k does not interact with the agent x\_i a long period, the evaluation of its trust should degrade gradually.

From the equation, we see that indirect trust is influenced by weighted average of recommendations from different recommenders along with feedback credibility of the recommenders and decay with time. The recommendation made by a recommender or asking friends is its own experience, i.e., its own direct trust

\[3.3.3: \text{Expected Trust mechanism}\]

The expected trust that reflects expected performance of target agent depends on two aspects: One is judging the information of the agent’s recent transactions itself. The second is judging the information of historical transactions. It is quantitative expression of trust, refers to compressive trust value which synthesizes the direct trust and indirect trust. Direct trust is given higher weight as the evaluating agent performs more and more interactions with the target agent, i.e., the evaluator becomes more confident about its own experience than taking recommendation from others.

Let, ET\_n\_t(x_i, x_j) represent the expected trust degree of agent x\_j from agent x\_i’s perspective. The expected trust degree is calculated by the following equation:

\[ET\_n\_t(x_i, x_j) = \beta \times DT\_n\_t(x_i, x_j) + (1 - \beta) \times IT\_n\_t(x_i, x_j)\]

Where \( \beta \) represents the weight of direct trust which can be calculated as follows

\[\beta = \frac{1}{1 + \left( \frac{IT\_n\_t(x_i, x_j)}{DT\_n\_t(x_i, x_j)} \right)}\]

Here X = TS(x\_i), represents the set of agents who have ever interacted with agent x\_j and IT\_n\_t(x_i, x_j) represents the number of interactions agent x\_i has conducted with agent x\_j in the t-th interval. So, M\_n\_t(x\_i, x\_j) represents the mean number of interactions that other agents (agents other than x\_i) have conducted with agent x\_j. If |X - {x\_i}| = 0 then we set M\_n\_t(x\_i, x\_j) = 0 and if \( \Gamma\_n\_t(x\_i, x\_j) + M\_n\_t(x\_i, x\_j) = 0 \) then we set \( \beta = 0.5 \) (default value). We have weighted the interaction count (I) with feedback credibility of the recommenders in computing M\_n\_t(x\_i, x\_j).

We can see that if \( \Gamma\_n\_t(x\_i, x\_j) \) increases (compared to M\_n\_t(x\_i, x\_j)) the value of \( \beta \) also increases signifying that as the evaluator becomes more experienced, it tends to rely more on its own judgment. This is consistent the fact that people communicate with each other in society, as the same time it can resist the malicious agent false recommendation.
3.4 Application

In this paper, we consider a scenario in which customer selects service provider agent (bank) with highest trust value by using selection algorithm explained below. After that customer requests / applies to selected service provider for a business/personal loan by submitting his personal details and loan requirement details.

To fulfill the customer’s request, service provider agent gathers and verifies customer’s information at various government departments. After that service provider checks its loan approval policies with customer information sent by various government departments, customer’s credit score obtained from CIBIL (Credit Information Bureaus of India Limited), regulations prescribed by central bank(RBI), personal details of customer such as customer’s negativity, assets, salary etc. If serviced provider loan approval policies are satisfied with customer’s information, then service provider will send loan approval information and attribute values of QoS provided to customer who sent request for service. After that, customer calculates trust value of service provider, and then it decides whether the selected service provider is trustworthy or not. If customer thinks that this service provider can be trusted, it will send the willing message. The transaction process is explained in figure 1.

In this paper, we use the term agent to represent any individual, government department and public/private service providers. In an interactive, each agent may play one of three kinds of roles: customer (C), Service Provider (SP), and Government Department (GD).

Description of selection Algorithm 1: For selective scenario, we first compute the trust of agents who respond to a transaction request and then we select the agent with the highest trust value. In this proposed algorithm1, S is the set of responders (agents that respond to a transaction request) where $S \in X$. We then first seek to choose an agent from S based on a threshold value of trust ($Y$). We finally sort the responders in decreasing order of trust value and select the one with highest trust value.

Algorithm 1 Selection of service provider agent ($x_i, S$)

Input: $x_i$ is a evaluating agent and $S$ is the set of agents responding to a service request

Where \( \{x_i\}, S \in X \)

Output: Service providing agent $x_j$

definition for each $x \in S$ do

Compute Trust ($x_i, x$)

if Trust($x_i, x$) > $Y$ then

    $S \leftarrow S \cup \{x\}$

end if

end for

Sort agents in $S$ in decreasing order of trust value and return agent $x_j$ with the highest trust value.

Step 1: In this transaction process, customer asks all service providers for getting service. After that customer identify that service providing agents who respond to a transaction request is determined, and then the agent with the highest trust value is selected by using algorithm1. After that, customer sends service request to service providing agent who has highest trust value.

Step 2: when SP receives the customer service request, SP will check validation of customer’s request. After validation of customer’s request, SP will prepare and send its request message to various GDs for getting customer details.

Step 3: After validation of SP’s request, then the government department should check the trust value of the service provider. Based on trust value of SP, GDs decides whether they send the customer details to SP or not. If GDs is satisfied with trust value of SP, then they will send customer details to SP.

Step 4: When SP receives GDs’s responses, SP will validate them. Then SP should check its policy with customer details sent by GDs. If SP’s policy is satisfied with customer details, then SP will send various attribute values of QoS itself presently to the customer.
Step 5: When customer receives SP’s response, the customer computes QoS based on various attribute value of QoS provided by service provider, judges whether it is satisfied for its own requirement or not.

If customer is satisfied, then customer again sends its service request to the selected service provider. Otherwise, customer selects next highest trust valued service provider agent and repeating procedure from step 1 until QoS is satisfied by service requester.

Step 6: After validation of customer’s willing message, SP will provide service to customer.

Step 7: When customer receives SP’s response, customer will validate and evaluate QoS of service provider. If the service does not match what it claimed to provide, then we say there is some purpose on agent $x_j$, we think that $x_j$ is unbelievable and reduce the value of its reputation. Otherwise, the $x_j$ declared true characteristics of its service, $x_j$ is believable, its its trust value should be increased. After completion of transaction, service requester will update its trust table.

IV. IMPLEMENTATION AND ANALYSIS

In our proposed trust model, statistical results have been obtained in order to evaluate the dynamic trust agents effectively even in presence of involving (oscillating) malicious behavior. The trustworthiness of a service provider is based on trust mechanisms such as direct trust, indirect and expected trust. We have defined various scenarios for our tests to evaluate our trust approach, and show its effectiveness, feasibility and benefits. Simulation setup, environmental setup, agent’s Behavioral Pattern setup and the tested scenarios, its results, and analysis are presented in the following sub sections. In all graphs the x-axis is the number of interactions and the y-axis is the correspondent expected trust value in each scenario.

4.1 Simulation Setup

The proposed model has been implemented and some statistics has been obtained in order to validate expected trust. From software point of view, we used a network simulation tool to create all agents in network to develop all trust calculations. From the hardware’s point of view we used Pentium-4(P4) 3.00 Ghz processor with 2GB RAM. The simulation tool use asynchronous interactions between machines and different scenarios were simulated according to specific policies in the network. Some assumptions were defined in the test environment in order to organize the tests.

4.2 Environment Settings

Our simulated environment contains N agents. N is set to 100 in almost all the experiment.

However, in one experiment we have varied the value of N to show the scalability of the trust model. It was evident from the experiment that the variation in N did not affect the performance of the trust model and as such N was set to 100 for most of the experiments. The agents are of mainly two types- good and malicious. Good agents cooperate in providing both good service and honest feedback. In contrast, malicious agents are opportunistic in the sense that they cheat whenever it is advantageous for them. Malicious agents provide both ineffective service and false feedback. Table-1, summarizes the different parameters related to the environmental setting and trust computation. These default values have been empirically tuned.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td># of agents in the system</td>
<td>100</td>
</tr>
<tr>
<td>Ps</td>
<td>% of sincere agents</td>
<td>0-100</td>
</tr>
<tr>
<td>Pm</td>
<td>% of malicious agents</td>
<td>0-100</td>
</tr>
<tr>
<td>M</td>
<td>% of agents who respond to a transaction request</td>
<td>5%</td>
</tr>
<tr>
<td>α</td>
<td>Successful transactions rate factor</td>
<td>0.6</td>
</tr>
<tr>
<td>α</td>
<td>Historical factor for direct trust</td>
<td>0.4</td>
</tr>
<tr>
<td>threshold</td>
<td>Minimum threshold value of amount of satisfaction</td>
<td>0.05</td>
</tr>
<tr>
<td>β</td>
<td>Weightage factor of Direct trust</td>
<td>0.03</td>
</tr>
<tr>
<td>c</td>
<td>Deviation factor for amount of satisfaction</td>
<td>0.05</td>
</tr>
<tr>
<td>ε</td>
<td>Decay constant or decay rate</td>
<td>range from 0 to 1</td>
</tr>
<tr>
<td>$\sum_{m=1}^{M} \hat{W}_m$</td>
<td>Service attribute value</td>
<td>range from 0 to 1</td>
</tr>
</tbody>
</table>

4.3 Agent’s Behavioral Pattern setting

The behavioral pattern of good agents is quite easy to simulate as they provide good service and honest feedback. However, it is challenging to simulate an agent’s malicious behavior realistically. We mainly study three behavioral patterns namely- non-collusive, collusive and strategically altering.

**Sincere agent:** always behaves accordingly to provide other agents with reliable services and fair evaluation after service.

**Simply malicious agent:** always cheats other agents with unreliable service during transaction and gives false feedback to other agents i.e., they rate good agents poorly while rating malicious agents highly.
Random Behavior malicious agent: behaves randomly. This means that it is not known for certain by the other members in the network whether a particular agent behaved accordingly or not.

The collusive agents: is similar to the simply malicious agent with one additional feature that malicious agents form a collusive group and deterministically help each other by performing numerous fake transactions to boost their own rating. The aim of collusive agent is to other agents to believe the similar agents and refuse to service with legally agents.

Strategic malicious agents; may occasionally decide to cooperate in order to confuse the system. In this case, other agents are commonly fooled into thinking that the malicious agent is actually a good agent. These agents will give unreliable service to seek their illegal profits and make their credit value high within the system.

Scenario 1: All agents Behave Accordingly. In this test, all the recommended agents in the multi-agent network behave as expected. This means that they fulfill their requirements and perform their defined context correctly. Note that the entire agent acts following the same behavior, without changing any aspect of its functional context. In this case, the trust value of recommended agents is considered extremely trustworthy and it tends to stabilize in a value near 1 for both agent trust and Eigen trust model, thus avoiding blind trust. When there is no malicious agent in the network, the trust value of the agent reflects the individual behavior of the agents in the network. This is considered the ideal world. Figure 1 shows this result.

Scenario 2: Simply malicious agent: This agent always cheats other agents with unreliable service during transaction and gives false feedback to other agents i.e., they rate good agents poorly while rating malicious agents highly.

Scenario 3: Random Behavior of Agents. In this test all agents in the network behave randomly. This means that it is not known for certain by the other members whether a particular agent behaved accordingly or not. This test was set up in order to verify the results when nodes behave in a proper manner sometimes and then change their behavior with no particular reason. This can be considered the worst environment imaginable because it cannot be possible to predict if a node will or will not behave accordingly. This scenario is represented for both our trust and Eigen trust model in figure 3.
Scenario 4: strategically Behavior of Agents. In this test, some of agents behave strategically. This test simulates a coalition of agents in order to modify the service provider agent trust. Such behavior is considered as if the agents suffer some kind of attack or there are agents acting as black holes in the multi-agent system environment. When 20% of the agents start behaving in a malicious manner, the trust value of the node decreases and tends to stabilize in a value near 0.7 for our trust and near 0.5 for Eigen trust after 40 transactions. The analysis shows that the increase of the trust coefficients provided by the good agents overcomes the decrease of the coefficient of the malicious agents. Figure shows this result.

![Figure 4: In worst environment, strategically Behavior of Agents](image)

Scenario 5: Random Behavior of 60 Agents in network. In this test 60% of agents behave randomly, then the trust coefficient tends to stabilize in a value near 0.65 for our trust and near 0.35 for Eigen trust, as seen in Figure 5. When agents change their behavior, the trust value of the given agent decreases. When nodes behave randomly, the value of the given agent trust tends to 0.65. The reader may realize that the individual behavior of each member in the multi-agent network influences the trust value of the particular agent in the network. The results can also be considered satisfactory because all the agents are initiated at the same time in the network and interact with each other the same number of times.

![Figure 5: trust of a peer, when 60% nodes change their behavior](image)

V. CONCLUSION

In this paper, we proposed a Reputation-Based Dynamic Trust Evaluation Model that is used for effectively evaluating the dynamic trust entities even in presence of involving (oscillating) malicious behavior. In scenario 1, when all the recommended agents in the multi-agent network behave as expected, the trust value of recommended agents is considered extremely trustworthy and it tends to stabilize in a value near 1. In scenario 2, simply malicious agent always cheats other agents with unreliable service. When there is no malicious Agent, the Successful Transaction Rate could reach 100%. When the Malicious Agent reach to 50%, the Successful Transaction Rate still reach to above 70%, because it can effectively identify and restrain the Malicious Agent. In scenario 3, we observe that all agents in the network behave randomly. This test verifies the results when nodes behave in a proper manner sometimes and then change their behavior with no particular reason. In scenario 4, we observe that some of agents behave strategically. When 20% of the agents start behaving in a malicious manner, the trust value of the node decreases and tends to stabilize in a value near 0.7 for our trust and near 0.5 for Eigen trust after 40 transactions. In scenario5, we observe that 60% of agents behave randomly, and then the trust coefficient tends to stabilize in a value near 0.65 for our trust and near 0.35 for Eigen trust.

In this paper we have given a comprehensive mathematical definition of the different mechanisms related to computing trust. We also provide a model for combining all these mechanisms to evaluate trust. It is hoped that this research paper would help the bank management not only in improving the expected level of customer satisfaction but also strengthening the bond between the banks and their customers. Simulation results indicate that our model compared to other existing models can effectively cope with strategic behavioral change of malicious agents.

REFERENCES


