Dynamic Trust Model for Reliable Transactions in Multi-agent Systems

Anupam Das*, M. Mahfuzul Islam† *IEEE member* and Golam Sorwar‡
*†Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh.
Email: *anupamdas@cse.buet.ac.bd,†mahfuzul.islam@ieee.org
‡School of Commerce and Management, Multimedia and Information Technology Unit,
Southern Cross University, Hogbin Drive, Coffs Harbour, NSW 2457, Australia.
Email: ‡golam.sorwar@scu.edu.au

***Abstract*—Most network applications such as pervasive computing, grid computing, P2P etc. can be viewed as multi-agent systems which are open, anonymous and dynamic in nature. Due to such nature multi-agent systems present potential threats in ensuring secured communication. One of the key feasible ways to minimize threats is to evaluate the trust and reputation of the interacting agents. Many trust models have done so, but they fail to properly evaluate trust when malicious agents behave in an unpredictable way. Not only that they are also ineffective in providing quick response to a malicious agent's oscillating behavior i.e., they are unable to assess the true nature of an opportunistic agent. To cope with such strategically altering behavior this paper presents a dynamic trust model where we analyze the different factors related to the trust of an agent and then propose a comprehensive mathematical model for evaluating trust. Simulations show that our model compared to other existing models can effectively cope with the strategic behavioral change of an agent and at the same time efficiently isolate the malicious agents.**

***Index Terms*—Multi-agent System, trust model, reputation model, credibility, malicious behavior.**

I. Introduction

In a multi-agent system, agents interact with each other to achieve a definite goal that they cannot achieve alone and such systems include P2P, grid computing, MANETs and so on. Multi-agent Systems (MASs) are increasingly becoming popular in carrying valuable and secured data over the network. Nevertheless, the open and dynamic nature of MAS has made it a challenge for researchers to operate it in a secured environment for information transaction. Malicious agents are always seeking ways of exploiting any existing weakness in the network. Researchers have long been utilizing trust theory from social network to construct trust models for effectively suppressing the malicious behavior of participating agents. Trust issues have become more and more popular since traditional network security approaches such as the use of firewall, access control and authorized certification cannot predict the agent's behavior from the viewpoint of trust.

Most of the existing global trust models [1]–[8] can successfully isolate malicious agents when the agents behave in a predictable way. However, these models suffer greatly when agents start to show dynamic personality. These models also fail to quickly adapt to the strategic alternations made by malicious agents. Moreover, some of the models show little effect in dealing with more complex attacks such as unfair rating and collusion.

With this research problem in mind, we propose a dynamic trust model which can effectively detect sudden strategic alteration in malicious behavior. Our model considers a variety of factors in determining the trust of an agent such as recent and historical trend, decay of trust and so on. We have also used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing trust.

The remaining part of the paper is organized as follows. Section II reviews the existing related works. In section III we formally introduce our dynamic trust model. We describe a way to filter out unknown recommenders in section IV. We present the simulations and comparative studies in section V. In section VI we describe how our model resists the common threats confronted in a multi-agent system. Finally we conclude our paper in section VII.

II. Related Work

In this section we will describe some of the most recent and popular research works done on trust model. We will discuss the key ideas of - PeerTrust [3], FCTrust [5], SFTrust [4], recommendation trust model provided by *Wang et al.* [9], trust model provided by *Li et al.* [7] and trust model proposed by *Wen et al.* [6].

PeerTrust [3] computes the trustworthiness of an agent as the normalized feedback weighted against the credibility of the feedback originators. The limitation of PeerTrust is that it has to retrieve all the transactions within the recent time window (which may contain a large number of transactions) to compute the trust of an agent. So the process is expensive from both computational and spatial point of view. In computing the global trust of any agent, the recommendation model provided by *Wang et al.* [9] combines both local trust calculated from the agent's own experience and recommendation provided by other agents in the system. This model uses average successful transaction as local trust which fails to reflect the trend and relative time significance of a transaction. FCTrust [5] uses transaction density and similarity measure to define the credibility of any recommender as opposed to [1], [10] which use the global trust as the weight of the quality of feedbacks. However, FCTrust stores all the transaction performed within a time frame which might be a significantly long period and this adds in storage requirement. SFTrust [4] is a double trust metric model. It separates service trust from feedback trust with the view to take full advantage of all the agents' service abilities even in the presence of fluctuating feedbacks. But in computing recommendation trust an agent can forward recommendation request to their neighbors who can then forward the request to their neighbors and so on. Aggregating information through this kind of local broadcasting (limited by the TTL field) can be really time consuming. The trust model provided by *Li et al.* [7] addresses different aspects in determining the trust of an agent such as recent trust, historical trust, expected trust and confidence in trust of other agents. But there are a number of limitations in their approach like- the model considered either historical trust or recent trust in computing the expected trust but not both and this fails to reflect an agents true nature. *Wen et al.* [6] proposed a trust model which addresses both direct and referral credibility. In case of referral credibility it uses the transitive law of trust. However, its main flaw lies in the assumption that all routes to the desired agent have equal cost/weight which is not right as the longer the path the more possibility that a malicious agent lies in the path.

The models defined in [1]–[5], [7]–[10] do not address a critical aspect of trust which is the decay of trust value with time. Since the network present today is highly dynamic and unpredictable, trust values should decay with the elapse of time. Some models [6], [11] have their own decay functions. However, their decay functions fail to properly simulate a realistic decay process i.e., their decay functions have a high declination rate in the initial stage and a low declination rate at the latter stage. Realistically it should be the reverse that is the declination rate should be smaller in the earlier stage while it should be higher in the latter stage and we have incorporated such decay function.

III. Our Trust Model

The main objective of this paper is to provide a dynamic trust model for effectively evaluating the trust of agents even in the presence of highly oscillating malicious agents. A number of parameters have been considered in our trust model for computing the trust of an agent. Now, many of these parameters have been previously discussed in [3]–[7], [9] but none these models can fully cope with the strategic adaptations made by the malicious agents. The mathematical and logical definitions used for these parameters also cannot reflect the true scenarios faced in real life. Moreover none of the models have considered all the parameters that we have considered. In the following sections we will redefine the mathematical expressions used for some of the parameters and at the same time define some new ones and then finally combine all of the parameters to present our new trust model. For the following sections we assume that agent ***a*** (called evaluator) needs to calculate the trustworthiness of agent ***o*** (called the target agent).

A. Satisfaction

Satisfaction function keeps record of the satisfaction level of all the transactions an agent makes with another agent. However, instead of storing all of the transaction history we have defined a exponential averaging update function to store the value of satisfaction. This greatly reduces the storage overhead and at the same time assigns time relative significance to the transactions. Let, $S_t^i(a,o)$ represent the amount of satisfaction agent a has upon agent o based on its service up to *i* transactions in the *t-th* time interval. The satisfaction update function is defined as follows-

$$S_t^i(a,o) = \alpha S_{cur} + (1-\alpha) S_t^{i-1}(a,o)\ . \tag{1}$$

$S_t^0(a,o) = S_{t-1}^{last}(a,o)$ and $S_0^0(a,o) = 0$. Here, S_{cur} represents the satisfaction value for the most recent transaction and we have used a binary feedback system where an agent rates other agents with either 1 or 0 based on whether the transaction is satisfactory or not.

$$S_{cur} = \begin{cases} 0, & \text{if recent transaction is not satisfactory} \\ 1, & \text{if recent transaction is satisfactory} \end{cases} \tag{2}$$

The weight α changes based on the accumulated deviation $\xi_t^i(a,o)$.

$$\begin{aligned} \alpha &= threshold + c\frac{\delta_t^i(a,o)}{1+\xi_t^i(a,o)} && (3)\\ \delta_t^i(a,o) &= |S_t^{i-1}(a,o) - S_{cur}| && (4)\\ \xi_t^i(a,o) &= c\delta_t^i(a,o) + (1-c)\xi_t^{i-1}(a,o) && (5) \end{aligned}$$

$\xi_t^0(a,o) = \xi_{t-1}^{last}(a,o)$ and $\xi_0^0(a,o) = 0$. Here c in some user define constant factor. Initial value of α is set to 1 and *threshold* is set to 0.25.

B. Direct Trust

Direct trust also known as local trust represents the portion of the trust that an agent computes from its own experience with the target agent. Let, $DT_t^i(a,o)$ represent the direct trust that agent *a* has upon agent *o* up to *i* transactions in the *t-th* interval. We have used the satisfaction measure along with the credibility of the target agent to define direct trust.

$$DT_t^i(a,o) = S_t^i(a,o) * Cre_t^i(a,o) \tag{6}$$

where $Cre_t^i(a,o)$ represents the feedback credibility of agent o from agent a's perspective. Here credibility is being considered as a malicious agent may intentionally try to behave in a honest manner towards certain agents while it may be acting maliciously towards other agents in the system.

C. Indirect Trust

Indirect trust is computed from the experience of other agents in the system. When there is no or even little interaction with the target agent, an agent has to depend on the experience gained by other agents to make a fruitful decision. Indirect trust is computed by considering recommendation from other agents along with the credibility of the recommender. Let, $IT_t^i(a,o)$ represent the indirect trust that agent a computes about agent o.

$$IT_t^i(a,o) = \frac{\sum_{x \in W-\{a\}} Cre_t^i(a,x) * DT_t^i(x,o)}{\sum_{x \in W-\{a\}} Cre_t^i(a,x)} \tag{7}$$

Here $W = IS(o)$, represents the set of agents who have ever interacted with agent o. If $|W - \{a\}| = 0$ then we set $IT_t^i(a,o) = 0$.

D. Global Trust

Global trust reflects only the expected behaviors. We have defined global trust as a weighted combination of direct and indirect trust. Direct trust is given higher weight as more and more interactions occur with the target agent, that is, the evaluator becomes more confident about its own experience than taking recommendation from others. Let, $GT_t^i(a,o)$ represent the recent trust that agent a has upon agent o.

$$GT_t^i(a,o) = \beta DT_t^i(a,o) + (1-\beta) IT_t^i(a,o) \tag{8}$$

where β represents the weight of direct trust which can be dynamically calculated as follows-

$$\begin{aligned} \beta &= \frac{I_t(a,o)}{I_t(a,o) + M_t(a,o)} \\ M_t(a,o) &= \frac{\sum_{x \in W-\{a\}} I_t(x,o)}{|W - \{a\}|} \end{aligned}$$

Here $W = IS(o)$, represents the set of agents who have ever interacted with agent o and $I_t(a,o)$ represents the number of interactions agent a has conducted with agent o in the *t-th* interval. So, $M_t(a,o)$ represents the mean number of interactions that other agents (agents other than a) have conducted with agent o. If $|W - \{a\}| = 0$ then we set $M_t(a,o) = 0$ and if $I_t(a,o) + M_t(a,o) = 0$ then we set $\beta = 0.5$ (default value).

E. Feedback Credibility

To compute feedback credibility (for brevity we use the term credibility) we first determine personalized difference rating $D_t^i(a,o)$ between agent a and agent o. Then we use $D_t^i(a,o)$ to define the dynamic feedback credibility update function. Let, $PS(a)$ represent the set of agents with whom agent a has made interaction. Then $CIS(a,o) = PS(a) \bigcap PS(o)$ denotes the set of agents with whom both agent a and agent o have interacted.

$$D_t^i(a,o) = \sqrt{\frac{\sum_{x \in CIS(a,o)} (S_t^i(a,x) - S_t^i(o,x))^2}{|CIS(a,o)|}} \tag{9}$$

To measure the feedback credibility between agent a and agent o ($Cre_t^i(a,o)$), agent a compares $D_t^i(a,o)$ with the credibility deviation constant(τ) and then updates credibility according to the following function-

$$Cre_t^i(a,o) = \begin{cases} Cre_t^{i-1}(a,o) + \frac{1 - Cre_t^{i-1}(a,o)}{\mu}, \\ \qquad\qquad \text{if } D_t^i(a,o) < \tau \\ Cre_t^{i-1}(a,o) - \frac{Cre_t^{i-1}(a,o)}{\psi}, \text{else} \end{cases} \tag{10}$$

where μ represents rewarding factor and ψ represents punishment factor and both of them can be changed dynamically depending on the system. In equation (10) we see that when the difference rating is less than τ credibility increases and otherwise it decreases. Here $\mu, \psi \in \Re$ and we must make sure that that $\mu > \psi$ because the degree of punishment should be greater than that of incentive. By doing so, we incorporate the principle of "slow rise and quick decline". If however, $|CIS(a,o)| = 0$ then we set $D_t^i(a,o) = 0$ and $Cre_t^i(a,o) = 0.5$ (default value).

F. Decay model

Due to the highly dynamic nature of agents, trust should have the property of declination with lapse of time. If an agent has not interacted with the network for a long period, the evaluation of its trust should degrade gradually. Since in our model direct trust depends on satisfaction and indirect trust is calculated from the direct trust of recommenders we apply a decay function on satisfaction metric. The decay functions are given as follows-

$$\widehat{S}_t^i(a,o) = S_t^i(a,o) e^{-\lambda \Delta t} \tag{11}$$

$$\Delta t = t_{current} - t_{previous} \tag{12}$$

where $\widehat{S}_t^i(a,o)$ represents the value of satisfaction after decay. Here, λ is the decay constant and its controls how quickly the value will diminish to zero. Δt represents the interval between the current interaction and the last interaction. So as Δt increases a trustworthy agent becomes untrustworthy in the network. By including the time decay model we establish the principle "the more recent the transaction the more reliable it is".

G. Deviation Reliability

Deviation reliability is a measure of how much deviation we are willing to tolerate. Malicious agents sometimes strategically oscillate between raising their trust and milking the reputation which seriously affects the performance of the network. So, some kind of measurement is required to handle such scenario. Deviation reliability handles such trust fluctuation. To record the sudden misuse of trust by agents, we introduce the component *accumulated misused trust* (denoted as $AT_t^i(a,o)$).

$$AT_t^i(a,o) = \begin{cases} AT_t^{i-1}(a,o) + \frac{GT_t^i(a,o) - GT_t^{i-1}(a,o)}{\omega}, \\ \qquad \text{if } GT_t^i(a,o) - GT_t^{i-1}(a,o) > \varphi \\ AT_t^{i-1}(a,o) + GT_t^{i-1}(a,o) - GT_t^i(a,o), \\ \qquad \text{if } GT_t^i(a,o) - GT_t^{i-1}(a,o) < -\varphi \\ AT_t^{i-1}(a,o), \qquad \text{otherwise} \end{cases} \tag{13}$$

where φ represents the tolerated margin of error in the evaluation of trust and ω ($\omega > 1$) represents the punishment factor for sudden rise in trust. From equation (13) we see that we are considering both sudden rise and fall of trust by agents whereas Li *et al.* [7] considered only sudden fall of trust. However, we are penalizing lesser for sudden rise since we are encouraging agents to raise their trust through benevolent interactions. Initial value of accumulated misused trust $AT_0^0(a,o) = 0$.

Deviation reliability uses the accumulated misused trust metric to measure the deviation in agent behavior. Deviation reliability (denoted $DR_t^i(a,o)$) is defined by the following equation-

$$DR_t^i(a,o) = \begin{cases} 0, & \text{if } AT_t^i(a,o) > maxAT \\ \cos(\frac{\Pi}{2} * \frac{AT_t^i(a,o)}{maxAT}), & \text{otherwise} \end{cases} \tag{14}$$

where $maxAT$ represents the maximum tolerable misused trust. Initial value of deviation reliability $DR_t^i(a,o) = 1$.

H. Final Trust Metric

The trust metric is the actual value used in prioritizing all agents. It is computed from expected trust and confidence. Let, $T_t^i(a,o)$ represent the final trust value agent *a* places upon agent *o*.

$$T_t^i(a,o) = GT_t^i(a,o) * DR_t^i(a,o) \tag{15}$$

An agent will use equation (15) to select the target agent with the highest T value.

IV. Filtering Out Unknown Recommenders

In this section we discuss a methodology to filter out dishonest recommenders from the recommendation membership set. In computing indirect trust (IT) about a certain target agent we consult other agents that have interacted with the target agent, but malicious agents often provide false feedback to obscure the true nature of the target agent. This type of false feedback can lead to wrong interpretation about an agent's capability and hence hinder effective communication. In equation 7 we have taken feedback from every agent who has ever interacted with the target agent. However, the evaluating agent might not have interacted with many of these recommenders, as a result, the evaluating agent has no way of knowing how much credible they are and thus has to take the risk of relying on unknown recommenders. So we propose Algorithm 1 for filtering out dishonest recommenders.

Algorithm 1 Compute Recommender Set(a,o)

```
Input: Evaluating agent $a$ and target agent $o$
Output: Set of Recommenders $W$
set $W = PS(a) \bigcap IS(o)$
if $W = \emptyset$ then
  set $W = IS(o) - \{a\}$
end if
if $W \neq \emptyset$ then
  for each $x \in W$ do
    if $Cre(a,x) < \gamma$ then
      $W = W - \{x\}$
      #discarding unknown witnesses
    end if
  end for
end if
Return $W$
```

V. Experimental Evaluation

This section evaluates our model's performance and shows its feasibility and effectiveness. We have carried out our experiments to achieve two main objectives. Firstly, we assess how quickly it adapts to strategically oscillating behavior. Secondly, we demonstrates the effectiveness of our model compared to other existing trust models under different scenarios.

A. Simulation Setup

We have developed our simulation in java language using JBuilder and the discrete event simulation toolkit SimJava [12]. This section describes the general simulation setup, including the environment setting, agent's behavioral pattern and performance evaluation index.

Our simulated environment contains N agents where N is set to 1000 in almost all the experiments. The agents are 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. The percentage of malicious agents in the network is controlled by the parameter *mal*. In our simulation we have studied mainly three types of malicious behavioral pattern namely-noncollusive, collusive and strategic alteration. In noncollusive setting malicious agents deceive other agents and provide false feedback. The collusive setting is similar to the noncollusive setting 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. We have used the parameter *collusion* to denote the percentage of malicious agents forming collusive groups. In the strategically altering setting a malicious agent may occasionally decide to cooperate in order to fool the system. We use the parameter *mres* to model the rate of dishonest feedback by a malicious agent. We use the parameter *res_rate* to determine the rate of agents that respond to a transaction request.

Table I summarizes the the different parameters related to the environment setting and trust computing. The table also lists the default value of the different parameters used. These default values have been empirically tuned.

For comparing the performance of our trust model with other existing trust models we use the evaluation criterion called successful transaction rate (STR). STR is described

TABLE I
SIMULATION PARAMETERS SETTINGS

	Parameter	Default value
Environment Setting	N	1000
	mal	40%
	mres	100%
	res_rate	5%
	collusion	0%
Trust Computation Setting	α	1
	threshold	0.25
	c	0.9
	β	0.5
	μ	20
	ψ	4
	τ	0.25
	λ	0.1
	γ	0.8
	φ	0.25
	ω	2
	$maxAT$	10

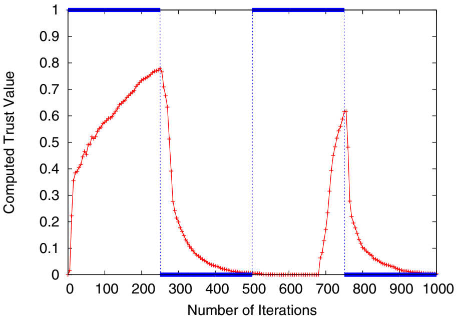

Fig. 1. Effectiveness of our model against dynamic personality.

as the ratio of the number of successful transactions to the total number of transactions. We determine STR against the variation of *mal*, *mres* and *collusion*. All the experimental results are averaged over 30 runs of the experiment.

B. Handling Dynamic Personality of agents

Up until now, we have considered more or less fixed personality of collusion. The objective of this experiment is to show how our trust model handles dynamic personality. Here we simulate the pattern where a malicious agent first builds up its reputation and then milks the built reputation and finally again tries to build its reputation i.e., the agent oscillates between building and milking reputation. For testing such scenario we simulate an environment which contains all good collusion except for only one malicious agent with dynamic personality. Fig. 1 shows the computed trust value of the malicious agent under altering behavioral pattern. We see that our trust model quickly responds to the sudden fall of performance by the malicious agent once it builds up its reputation giving it less scope of utilizing its built reputation. Once the trust value diminishes to a low value it requires significant number of consecutive good services for its trust value to rise again i.e., it must give proof of its cooperative nature. From Fig. 1 we see that in spite of the good nature of the malicious agent in the third slot its trust value rises very late and even in that case it does not rise to the previous value.

C. Comparison with other Trust Models

In this section we will demonstrate the superiority of our trust model over other existing trust models. In these experiments an agent first computes and compares the trust values of the responding agents and chooses the agent with the highest trust value for interaction. A transaction is successful if the participating agent is cooperative i.e., if it is a good agent. In all the experiments we compute STR as the evaluation criterion against different scenarios. We perform a total of 100 iterations where in each iteration each agent initiates one transaction. Transactions initiated by malicious agents have been discarded from the calculation of STR. Since the responders to a transaction request is generated at random we take the mean of 30 experiments for each scenario. We compare our model with SFTrust [4], FCTrust [5], P2P recommendation Trust model (for short we will use Reco-Trust) [9], trust model of users' behavior (for short we will use User-Trust) [6], PeerTrust [3] and Dynamic trust model for Multi-agent systems (for short we will use MAS-Trust) [7].

First, we calculate STR against the variation of percentage of malicious agents *mal* while setting *mres* to 100% and *collusion* to 0%. As from Fig. 2 we see that both our trust model and PeerTrust show superiority over the remaining trust models as the amount of malicious agents in the network increase beyond 40%. Networks today especially the internet hold great threats as it teems with malicious agents. So, in such networks both our trust model and PeerTrust would provide the best performance.

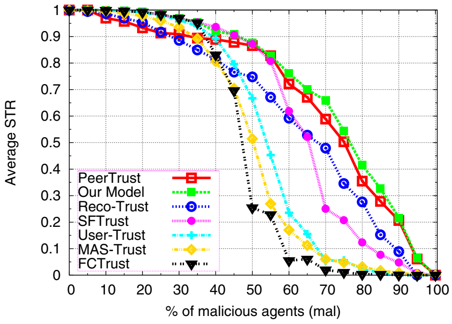

Fig. 2. Comparing our model with other existing trust models in terms of average STR against *mal*.

In the next experiment we want to observe the impact of collusion on STR. So, for this experiment we set *mal* to 60% because as the number of malicious agents increases their collusive impact becomes greater and we also set *mres* to 100%. Fig. 3 represents the computed STR against *collusion*. Due to the experimental randomness, the gradient of the curves may vary from experiment to experiment. From Fig. 3 we see that SFTrust, FCTrust, MAS-Trust and User-Trust suffer greatly as the collusive group size increase. On the other hand both our trust model and PeerTrust remain intact to collusive attack. The main reason behind this is the feedback credibility measure which filters out false feedbacks. Here false high ratings come from agents (namely malicious agents) with low credibility as a result they have no impact on STR.

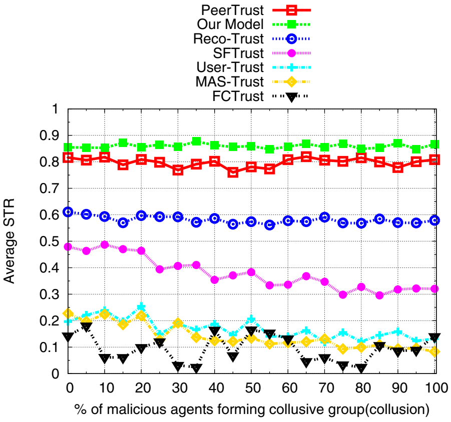

Fig. 3. Comparing our model with other existing trust models in terms of average STR against *collusion*.

In the third experiment we will analysis the impact of *mres* on STR. Malicious agents tend to fool other agents by oscillating between good and malicious nature. In this experiment we set *mal* to 60% while *collusion* is set to 0%. Fig. 4 represents the computed STR against *mres*. From the figures we see that our model out performs all the trust model significantly and PeerTrust suffers the most. This is because our model keeps track of sudden rise and fall of trust by agents and penalizes any agent showing frequent trust fluctuation. Thus, our model can successfully resist strategically altering behavior by malicious agents.

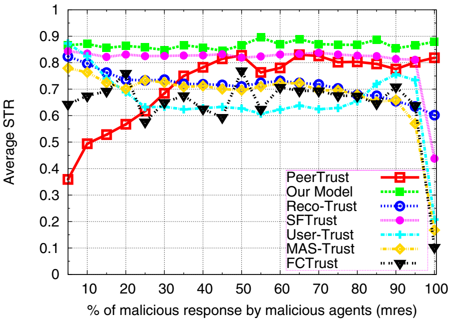

Fig. 4. Comparing our trust model with other trust models in terms of average STR against *mres* in presence of 60% malicious agents.

VI. Effectiveness against Threats

Our trust model provides a reputation based trust mechanism to resolve some of the threats and risks present in a multi-agent environment by enabling agents to choose reputed agents while avoiding untrustworthy agents. Here agents assist one another to avoid the malicious ones.

However, reputation-based trust mechanism also introduces vulnerabilities such as shilling attacks where adversaries attack the system by submitting false ratings to confuse the system. Shilling attack is often associated by collusion attack where malicious agents group up together and collaborate to raise each other's rating by making fake transactions. Our model prevents such threats by assigning feedback credibility to each feedback provider. By doing so, our model discards feedbacks given by malicious agents and thereby nullify collusion attack.

Another challenging threat that most trust models fail to handle it the strategic alteration of behavior by malicious agents. By cleverly alternating between good and malicious nature they try to remain undetected while causing damage. But our model keeps track of sudden rise and fall of trust and thereby can easily penalize such oscillating behavior.

VII. Conclusion

We have presented an innovative trust model which for the first time can effectively detect the strategically altering behavior of malicious agents and thus provide a more secured environment for communications. Our model can quickly detect sudden rise and fall of trust and thereby can easily penalize oscillating behavior. Analysis and simulation results show that compared with other existing trust models our model is more robust and effective in preventing attacks from opportunistic malicious agents.

References

[1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for Reputation Management in P2P Networks," in *Proceedings of the 12th international World Wide Web conference (WWW)*. Budapest, Hungary: ACM Press, 2003, pp. 640–651.

[2] Y. Wang and J. Vassileva, "Bayesian Network-Based Trust Model," in *Proceedings of IEEE/WIC International Conference on Web Intelligence (WI)*, Halifax, Canada, October 2003, pp. 372–378.

[3] L. Xiong and L. Li, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[4] Y. Zhang, S. Chen, and G. Yang, "SFTrust: A double Trust Metric Based Trust Model in Unstructured P2P Systems," in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing (ISPDP)*. IEEE Computer Soc. Press, 2009, pp. 1–7.

[5] J. Hu, Q. Wu, and B. Zhou, "FCTrust: A Robust and Efficient Feedback Credibility-Based Distributed P2P Trust Model," in *Proceedings of IEEE 9th International Conference for Young Computer Scientists (ICYCS)*. IEEE Computer Soc. Press, 2008, pp. 1963–1968.

[6] L. Wen, P. Lingdi, L. Kuijin, and C. Xiaoping, "Trust model of Users' behavior in Trustworthy Internet," in *Proceedings of IEEE WASE International Conference on Information Engineering (ICIE)*. IEEE Computer Soc. Press, 2009, pp. 403–406.

[7] B. Li, M. Xing, J. Zhu, and T. Che, "A Dynamic Trust Model for the Multi-agent Systems," in *Proceedings of IEEE International Symposiums on Information Processing (ISIP)*. IEEE Computer Soc. Press, 2008, pp. 500–504.

[8] Z. Runfang and H. Kai, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–476, 2007.

[9] X. Wang and L. Wang, "P2P Recommendation Trust Model," in *Proceedings of IEEE 8th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE Computer Soc. Press, 2008, pp. 591–595.

[10] D. Wen, W. Huaimin, J. Yan, and Z. Peng, "A Recommendation-Based Peer-to-Peer Trust Model," *Journal of Software*, vol. 15, no. 4, pp. 571–583, 2004.

[11] Y. Zhang, K. Wang, K. Li, W. Qu, and Y. Xiang, "A Time-decay Based P2P Trust Model," in *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2. IEEE Computer Soc. Press, 2009, pp. 235–238.

[12] R. McNab and F. Howell, "Using java for discrete event simulation," *Proceedings of the Twelfth UK Computer and Telecommunications Performance Engineering Workshop*, pp. 219–228, 1996.