RLM: A General Model for Trust Representation and Aggregation

Xiaofeng Wang, *Member, IEEE,* Ling Liu, *Senior Member, IEEE* and Jinshu Su, *Member, IEEE*

Abstract—Reputation-based trust systems provide important capability in open and service-oriented computing environments. Most existing trust models fail to assess the variance of a reputation prediction. Moreover, the summation method, widely used for reputation feedback aggregation, is vulnerable to malicious feedbacks. This paper presents a general trust model, called RLM, for a more comprehensive and robust reputation evaluation. Concretely, we define a comprehensive reputation evaluation method based on two attributes: reputation value and reputation prediction variance. The reputation predication variance serves as a quality measure of the reputation value computed based on aggregation of feedbacks. For feedback aggregation, we propose the novel Kalman aggregation method, which can inherently support robust trust evaluation. To defend against malicious and coordinated feedbacks, we design the Expectation Maximization algorithm to autonomously mitigate the influence of a malicious feedback, and further apply the hypothesis test method to resist malicious feedbacks precisely. Through theoretical analysis, we demonstrate the robustness of the RLM design against adulating and defaming attacks, two popular types of feedback attacks. Our experiments show that the RLM model can effectively capture the reputation's evolution and outperform the popular summation based trust models in terms of both accuracy and attack resilience. Concretely, under the attack of collusive malicious feedbacks, RLM offers higher robustness for the reputation prediction and a lower false positive rate for the malicious feedback detection.

Index Terms—trust model, accuracy assessment, malicious feedback, robustness.

1 INTRODUCTION

The rapid growth of Internet and ubiquitous connectivity has spurred the development of various collaborative computing systems such as service-oriented computing (SOC), Peer-to-Peer (P2P) and online community systems. In these applications, the service consumer usually knows little about the service providers, which often makes the consumer accept the risk of working with some providers without prior interaction or experience[1]. To mitigate the potential risks of the consumers, reputation-based trust systems [1,2] are deployed as a popular approach to predict how much the service provider can be trusted. The reputation value plays a pivotal role in aggregating, filtering, and ordering information for consumers to select service providers, and it can act as an incentive for service providers to improve their Quality-of-Service. Over the past few years, many reputation (social trust) models have been proposed for different applications such as: social web services [12,13,24], decentralized overlay networks and applications [5,14], multi-agent systems [9,10,11] and recommender systems [4,21,25].

Reputation is a statistical value about the trust probability derived from the behaviour history. Usually, the reputation is based on the interactions carried out directly between providers and the evaluator (personal experience) and the recommendations made by other consumers (feedback) [1]. From the personal experience's perspective, most existing work used the simple average [23], the Bayesian [8,9] or the belief models [10,11] to quantify the trust as some statistical values. However, they ignore another important attribute of the predicted statistical value, namely the prediction variance (or prediction accuracy), which depicts how much the trust prediction may deviate from the real one. For example, a service provider has a service success probability of 0.9. But due to the incomplete personal experience, a customer quantifies the provider's trust as 0.7. By using existing trust models, the customer can neither assess the accuracy of the reputation prediction made by her nor assess the trust values recommended by other in order to use them in her recommendation. Hence, it is hard for a consumer to decide how much to rely on the prediction of the feedbacks made by others to make her own trust decision. Moreover, when the customer recommends this trust prediction to others as a feedback, she cannot give reliable suggestion about how to aggregate the feedback so that others can minimize the variance of their trust evaluation.

To aggregate feedbacks recommended by others, the summation method is widely applied in reputation systems, such as eBay [23] and Eigentrust [15]. However, several have shown that it is easy to manipulate summation based feedback aggregation by malicious nodes for their personal profits [3,16]. A malicious node can falsely improve its own reputation or degrade the reputations of others. As a measure to defend malicious feedbacks for

- *X. Wang and J. Su are with the School of Computer, National University of Defense Technology, Changsha, 410073, Hunan, China. E-mail: (xf_wang,sjs)@nudt.edu.cn.*
- *L. Liu is with the College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332. E-mail: lingliu@cc.gatech.edu.*

the summation method, most existing work weighted the feedbacks by considering their credibility, such as the trust value based credibility used in Eigentrust [15] and the personalized similarity based credibility used in PeerTrust [5]. However, these credibility techniques usually need accurate trust knowledge of the system [5,15,16] or manually tuned intuitive parameters [9,11], which are often unrealistic or impractical in a real world application. We believe that the feedback credibility based techniques lack of the robustness to resist malicious feedbacks.

In this paper, we present the Robust Linear Markov (RLM) model for a more comprehensive and robust reputation evaluation, which significantly extend our earlier work [28]. The main contributions of our RLM model are three folds.

First, in contrast to existing feedback based reputation trust models, our RLM model represents the reputation trust by two attributes: reputation value and reputation prediction variance. The model is tracked by a linear hidden Markov process, so that a more comprehensive and accurate reputation can be evaluated. The assessment of the reputation prediction variance can help to achieve a better local decision making as well as a more intelligent third-party reputation aggregation.

Second, we propose the Kalman aggregation method for feedback aggregation instead of using the intuitive summation method. Our Kalman aggregation method can adjust the influence of a malicious feedback by the parameter of estimated feedback variance, which is used to support our robust trust evaluation techniques.

Third but not the least, to defend against the random/coordinated malicious feedback attacks defined in section 3.2, we design and demonstrate a robust two-phase calibration method for our RLM trust model. First, we introduce the Expectation Maximization (EM) algorithm to autonomously calibrate the model parameters to mitigate the influence of a malicious feedback. Then, we enhance the model with the hypothesis test method, which can resist malicious feedbacks more effectively with a confidence level. We provide theoretical analysis to demonstrate the robustness of our design.

To our best knowledge, RLM is the first trust model that can enable an evaluator to assess the accuracy of a reputation prediction made by itself. Unlike the summation aggregation method, our Kalman feedback aggregation can inherently support robustness techniques based on the inference theory. Moreover, the proposed model calibration method can resist malicious feedbacks autonomously and precisely. In the paper, we give both theoretical proof and experiments to demonstrate the validation, accuracy and robustness of the RLM model. With a firm basis in the statistics inference theory, our RLM trust model supplies a new way to construct a robust reputation system for distributed and open service-oriented environments.

The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 formulates the comprehensive trust evaluation problem and possible attacks. Section 4 describes our RLM trust model and Kalman feedback aggregation. Section 5 introduces the robust model calibration using EM and hypothesis test methods. Experimental results are presented in Section 6, followed by the conclusion in Section 7.

2 RELATED WORK

In open service-oriented environments, reputation based trust systems can determine how much an unknown service provider can be trusted in future interactions. Usually, the reputation/trust value can be modeled by two parts: the direct trust value from the evaluator and the feedbacks from others [1]. To measure the direct trust, Song et al. [7] used the fuzzy logic to compute the reputation score, which is the trust index's numerical value derived from some rules. The Bayesian reputation [8] computes the trust value according to the beta probability density functions (PDF). The posteriori reputation value is decided by $\alpha + 1/\alpha + \beta + 2$, where α and β are two parameters denoting the number of positive and negative results. Wang and Singh [10] modelled the reputation as a three dimension belief (b, d, u), representing the probabilities of positive, negative and uncertain outcomes. All these models quantify the trust as some predicted probability values. However, they ignore the prediction variance, which is one of the two attributes of a statistical prediction (i.e. [19]). Hence, these trust models cannot assess the accuracy of a reputation prediction made by itself. In contrast, the reputation prediction variance is considered in our RLM model to give a more comprehensive and accurate reputation evaluation, and both the reputation value and its prediction variance are tracked by our reputation filter.

To aggregate reputation feedbacks, the summation method [5,6,15] is widely used. The simplest summation method is to sum the number of positive ratings and negative ratings separately like eBay [23]. Combined with different system architectures, the summation method can have different forms. For example, in P2P systems, the Eigentrust used the trust value to weight a peer's feedback, and then they got the global reputation summation in a matrix notation. In the Bayesian reputation system, a feedback comprises the number of positive outcomes r and the number of negative outcomes s. The feedback is aggregated by adding r and s to the totalized positive and negative outcomes α and β respectively. Hence, we can say that the essence of beta aggregation is also a summation method. Although the summation method is easy to aggregate feedbacks, it lacks the support for robustness to resist malicious feedbacks. However, our proposed Kalman feedback aggregation method can adjust the influence of a malicious feedback through the parameter of estimated feedback variance, which supplies a support to resist malicious feedbacks.

In the aggregation of feedbacks, one fundamental problem is how to cope with the shilling attack [16] where malicious nodes submit dishonest feedback to boost their own ratings or bad-mouth legal nodes. Most existing work considered the credibility of a feedback to detect malicious feedbacks, and they are compared in the literature [27]. A simple solution for measuring the credibility of a node's feedback is to use the node's reputation value, which is used in EigenTrust [15] and PowerTrust [6]. However it is possible that a node may maintain a good reputation by providing high quality services, but send malicious feedbacks to its competitors. The credibility can also be measured by using personalized similarity (PSM) [5,16], where peer w uses a personalized similarity between itself and another peer v to weight the feedbacks from peer v. The disadvantage of PSM is that the peer w needs to have the wide trust knowledge about peer v's rating on some special peers, which is sometimes an unrealistic precondition. For other credibility methods, Yu and Singh [11] proposed the Weighted Majority Algorithm (WMA) and Whitby et al. [9] used the quantile detection method to filter out unfair ratings. Both these two methods need manually tuned intuitive parameters without guarantee of any quantitative confidence. In contrast, we employ the EM algorithm to get a robust parameter calibration, so that our RLM trust model can autonomously run without requiring the system trust knowledge or manual actions. Moreover, our hypothesis test method can filter out a malicious feedback precisely with a specific confidence level.

3 Comprehensive Trust and Attacks

A reputation-based trust system usually comprises two components: the underlying architecture, which concerns of how to distribute and collect the feedbacks, and the trust model, which describes the representation and aggregation of reputation-based trusts. This paper focuses on the design of a comprehensive and robust general trust model. In this section, we first formulate the problem of building a comprehensive trust model, which takes into account the accuracy evaluation of trust predictions. Then, we discuss the possible feedback attacks to the trust model.

3.1 Comprehensive Trust Formulation

We argue that to get a comprehensive trust prediction, trust models need to provide the local assessment of trust prediction accuracy. Since the reputation value is essentially a statistical value derived from the observation samples (reputation feedbacks), we model the reputation in a statistical form. Assuming that the real reputation of a node is denoted by R, which is not known by the trust evaluators. In a comprehensive model, we try to predict the actual reputation value by trust evaluation, denoted as a two dimension tuple, namely $rep = \{\langle R\rangle, P\}$, where $\langle R\rangle$ is the predicted reputation value, and P is the estimated reputation prediction variance, which is an estimation about the square error between $\langle R\rangle$ and the real reputation R. The attribute P is an evaluation about the accuracy of the predicted reputation value $\langle R\rangle$, which can be understood as the evaluator's confidence in $\langle R\rangle$. Hence, the lower the estimated prediction variance P is, the more confidence will the evaluator have in the predicted reputation value $\langle R\rangle$.

Upon obtaining the prediction $\langle R\rangle$ and its estimated prediction variance P for a node, the node can send the tuple rep to others as a reputation feedback. Hence, a feedback can be denoted as $f = \{z, c\}$, where z (coming from $\langle R\rangle$) is the feedback reputation value, and c (coming from P) is the suggested feedback variance, which indicates how accurate the feedback reputation value z is, and serves as a hint to others about how to intelligently aggregate the feedback reputation value. A bigger suggested feedback variance c means that the recommender has less confidence in z. Hence the aggregator should reduce the influence of the feedback in his reputation aggregation.

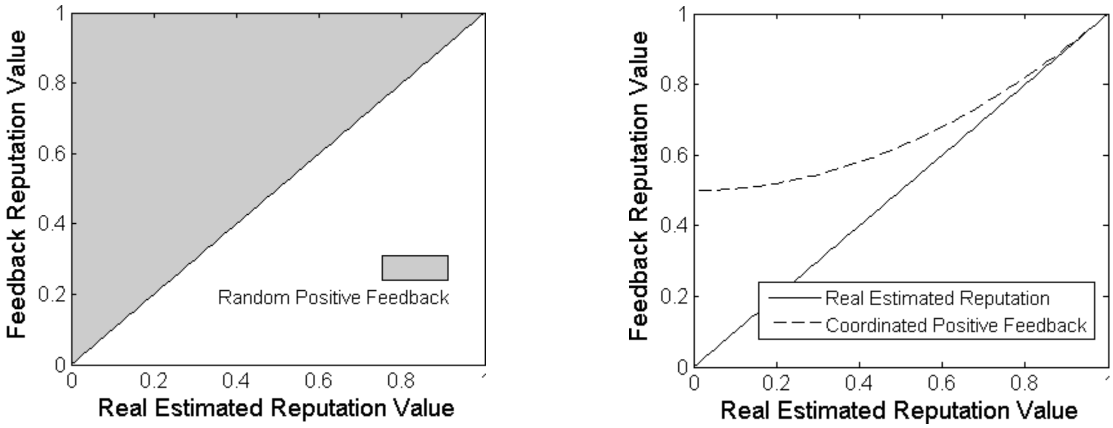
We dedicate Section 4 to the comprehensive trust evaluation problem. Concretely, we use a linear hidden Markov process to track the evolution of trust state, and propose the Kalman aggregation method for feedback aggregation instead of using the intuitive summation method.

3.2 Malicious feedback attack model

Attackers in a reputation system can either work alone or launch attacks by colluding with one another. A collusive attack can be implemented by disparate attackers or a single attacker acquiring multiple identities through a Sybil attack [26]. Typically, the effect of a single attacker is relatively small, but collusive attackers usually have much more severe influence on the reputation system. They can cooperate to issue high volumes of malicious feedbacks, which are more difficult to defend against. Hence in this paper, we are primarily concerned with the collusive reputation attack which has large number of malicious feedbacks.

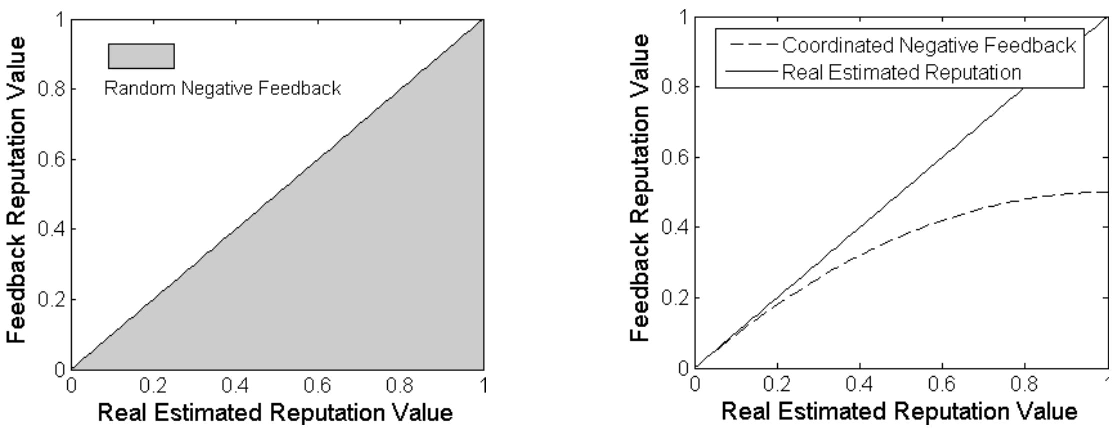
In this paper we can classify the malicious feedback into two types: adulating feedbacks and defaming feedbacks. In adulating feedback attacks, attackers try to falsely improve the reputation of their own or their partners. One basic form of the attack occurs when the multiple colluding attackers send unfairly positive feedbacks about each other. The adulating feedback reputation value can be modeled in two ways:

1) Random positive feedback: the feedback reputation value is a random value (as shown in Fig.1(a)) between 1 and the predicted reputation value set by the attacker. In such attacks, colluding attackers send random feedback reputation values about the target separately without coherence.
2) Coordinated positive feedback: the feedback reputation value has a deterministic relationship with



(a) Random Positive Feedback (b) Coordinated Positive Feedback

Fig. 1. Adulating Feedback Model



(a) Random Negative Feedback (b) Coordinated Negative Feedback

Fig. 2. Defaming Feedback Model

the desired reputation value predicted by the attacker. In such attacks, all the participating attackers seek to send feedback values coherent to each other. Consider the example shown in Fig.1(b), given the estimated reputation value x, the reputation value for the coordinated positive feedback is $(x^2 + 1)/2$.

In contrast to adulating feedback attacks, defaming feedback attacks try to degrade the reputation of others. Similarly, they can be modeled in two ways:

1) Random negative feedback: the feedback reputation value is a random value (as shown in Fig.2(a)) between 0 and the actual reputation value predicted by using the trust model.
2) Coordinated negative feedback: all the participating attackers seek to send coherent feedback reputation values, which are smaller than the actual reputation values predicted by using the model. As an example shown in Fig.2(b), given the real estimated reputation value x, the reputation value for the coordinated negative feedback is $(-x^2 + 2x)/2$.

To defend against the random/coordinated malicious feedback attacks, we develop a robust two-phase calibration method for RLM model in Section 5. In the first phase, we use the Expectation Maximization (EM) algorithm to autonomously calibrate the model parameters, which can mitigate the influence of a malicious feedback. In the second phase, we further enhance the model with the hypothesis test method, which can be more resilient to malicious feedbacks with a confidence level.

4 RLM Trust Model

To maintain the reputation for a node, we assume that the evaluator can receive feedbacks about the node continually through feedback sessions. All the feedbacks are assumed to be real time recommendations. The feedback received at session k is denoted as $f_k = \{z_k, c_k\}$, z_k and c_k represent the feedback reputation value and suggested feedback variance respectively. After each reception of a feedback f_k, the evaluator tries to predict the real time reputation R_k of the node, and evaluate the prediction variance P_k. Ideally, the reputation feedback value should equal to the real reputation. But due to the incomplete knowledge of the recommender and transient fluctuations of the service quality, the feedback reputation value usually has a deviation from the real reputation. Because many independent sources contribute to this deviation, it is reasonable to model the deviation as a zero mean Gaussian noise $Normal(0, Q_k)$. Hence, we can model the relation between the feedback reputation value and the real reputation value as:

$$z_k = R_k + q_k \text{ and } q_k \sim Normal(0, Q_k) \quad (1)$$

where Q_k is the estimated feedback variance, which is a parameter estimated locally by the reputation aggregator for a feedback. A bigger Q_k means a bigger reputation prediction variance estimated by the aggregator for the feedback f_k. Hence, the feedback reputation value z_k should have a smaller influence on the reputation aggregation. This will be demonstrated in the next section. It should be noted that the estimated feedback variance Q_k is different from the suggested feedback variance c_k. Although they are both estimated prediction variance about the feedback reputation value, c_k is suggested by the recommender, which can be honest or malicious, and Q_k is a new local evaluation made by the aggregator. Intuitively, a bigger (resp. smaller) suggested c_k will result in a bigger (resp. smaller) Q_k estimated by the aggregator, which will be demonstrated later in Theorem 4.

For a normal node, we assume that its reputation follows a stochastic process. In the statistical inference theory, the reputation prediction problem belongs to the infinite impulse response filter problem, which is to predict a new reputation value based on the feedback samples and previous reputation values. For the infinite impulse response filter, linear autoregressive (AR) model is widely used, which is reported to have a good prediction performance [19]. Hence, we also use the linear autoregressive model to define the reputation space evolution, and the nonlinear evolution can be treated with locally weighted methods in a similar fashion [18]. As the first approximation, the reputation R_k can be modeled as a first order linear AR model:

$$R_k = A_k R_{k-1} + w_k \text{ and } w_k \sim Normal(0, W_k) \quad (2)$$

where A_k is the reputation state transfer factor, and W_k is the variance for the state transfer noise. These two parameters need to be dynamically estimated by the reputation aggregator.

Equations 1 and 2 define a linear space model for the reputation. This linear model forms a hidden Markov problem as illustrated in Fig.3 (a Markov process with unknown state parameter R_k). The square nodes are targeted attributes of the reputation evaluation, double squares are observed reputation feedbacks, and circular nodes are dynamic parameters to be tuned. Our goal is to obtain the reputation value R_k and its estimated prediction variance P_k from this model, which will be introduced in the next section by using our Kalman feedback aggregation method. All the dynamic parameters in the model such as A_k, Q_k and W_k, will be tuned to cope with malicious feedback, which will be introduced in section 5.

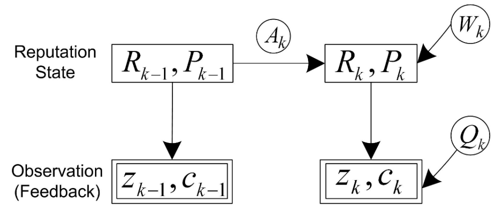


Fig. 3. Graphical RLM Model

4.1 Kalman Feedback Aggregation

In RLM model, the reputation's state evolution can be tracked in the aggregation of reputation feedbacks. The Kalman Filter (KF) is an optimal linear estimator for linear Gaussian systems, and it can give the least mean squared prediction of the system state [17]. Because of the linear properties of our RLM trust model, we change the typical Kalman filter to aggregate RLM reputation feedback. Our Kalman feedback aggregation can simultaneously track the evolution of the reputation value and its prediction variance. Moreover, it can adjust the influence of a feedback by the estimated feedback variance, which can support further robustness techniques to counter the malicious feedback.

To run the Kalman feedback aggregation, all the dynamic parameters (A_k, Q_k and W_k) in the model are assumed to be known. They will be tuned by our robust model calibration method in the next section. The Kalman aggregation method comprises two steps: the propagation step and the update step. Let R'_k denote the posteriori prediction of R_k, P'_k the posteriori estimation of P_k, and the symbol $\langle\rangle$ denote the prediction operator. Then, the corresponding equations for our Kalman feedback aggregation can be defined in Equations $3 \sim 7$, for $k = 1, \cdots N$.

Propagation Step

$$R'_k = A_k \langle R_{k-1} \rangle \quad (3)$$

$$P'_k = A_k^2 P_{k-1} + W_k \quad (4)$$

In the propagation step, the posteriori prediction of R_k and P_k are computed according to the RLM model. To run the Kalman feedback aggregation, we initialize the reputation value $\langle R_0 \rangle$ as 0.5, meaning we know nothing about the initial trust, and the prediction variance $P_0 = 0.01$ (a big variance value), meaning that we are not quite sure about the initial reputation prediction [19].

Update Step

$$S_k = P'_k + Q_k \quad (5)$$

$$\langle R_k \rangle = R'_k + \frac{P'_k}{S_k}(z_k - R'_k) \quad (6)$$

$$P_k = \frac{Q_k}{S_k} P'_k \quad (7)$$

In the update step, the feedbacks are aggregated to minimize the mean squared error of the reputation evaluation. Equation (5) computes the variance S_k of the residual prediction error. The final prediction of the reputation value R_k is updated by considering the deviation $(z_k - R'_k)$ and the ratio P'_k/S_k in Equation (6), and we can get the following theorem:

Theorem 1 Let f_1 and f_2 denote the reputation feedbacks to be aggregated. If f_1 has a bigger estimated feedback variance than f_2, namely $Q_1 > Q_2$, then f_1 will have smaller influence on the reputation value prediction $\langle R_1 \rangle$ than f_2.

Proof: Assuming that there are two feedbacks f_1 and f_2 to be aggregated, Q_1 and Q_2 are estimated feedback variance for f_1 and f_2 respectively, and $Q_1 > Q_2$. Let P_1 and P_2 refer to the estimated prediction variance before aggregating f_1 and f_2 respectively, then $P_1 = P_2$ since they all refer to the current state. From Equation (5), we can find that a bigger Q_1 will result in a bigger residual prediction variance S_1, and $S_1 > S_2$, then $P_1/S_1 < P_2/S_2$, leading to a slight update of the predicted reputation value $\langle R_1 \rangle$ in Equation (6).

Through Theorem 1, our Kalman feedback aggregation supplies a support to defend malicious feedbacks. A robust parameter calibration method should assign a big estimated feedback variance for a malicious feedback, so that the malicious feedback can only have a small influence on the reputation aggregation.

In Equation (7), the estimated prediction variance P_k is updated by the factor Q_k/S_k, and we can get the conclusion Theorem 2. It illustrates that when a evaluator aggregates a feedback with a big estimated feedback variance Q_k (denoting the variance estimated by the evaluator for the feedback reputation value), the estimated prediction variance of the new reputation prediction will increase, which means that the aggregator will be less confident about the predicted reputation value.

Theorem 2 Let f_1 and f_2 denote the aggregated reputation feedbacks with the same reputation transfer parameters ($A_1 = A_2, W_1 = W_2$). Let Q_1 and Q_2 denote the estimated feedback variances for f_1 and f_2, and P_1 and P_2 refer to the estimated prediction variance from the RLM model based on f_1 and f_2 respectively. The estimated prediction variance P_1 is bigger than P_2 for the new reputation prediction, if f_1 has a bigger estimated feedback variance than f_2, namely $Q_1 > Q_2$.

Proof: Assuming that there are two feedbacks f_1 and f_2 under a given reputation state, and they have the same reputation transfer parameters: the reputation state transfer factor A and transfer noise variance W. Given that feedback f_1 has a bigger estimated feedback variance Q_1 than f_2, namely $Q_1 > Q_2$, we want to show that $P_1 > P_2$. From Equation (5), we have $Q_1/S_1 > Q_2/S_2$, which leads to a bigger estimated prediction variance P_1 by Equation (7).

5 Robust RLM Model Calibration

Before running the Kalman feedback aggregation, the parameters A_k, Q_k and W_k in RLM model need to be computed. More importantly, the RLM model needs to be robust to the malicious feedback defined in section 3.2. In this section, we first introduce the Expectation Maximization (EM) algorithm to autonomously give maximum likelihood estimation for these parameters locally. The EM algorithm can mitigate the influence of a malicious feedback that has incorrect feedback reputation value. Then, we further enhance the model with the hypothesis test method to resist malicious feedbacks that have both incorrect feedback reputation value and incorrect suggested feedback variance.

5.1 Parameter Calibration

To defend against malicious feedbacks, we need a robust and autonomous parameter calibration method for RLM model. In this subsection, we design the expectation maximization (EM) algorithm, which can give a maximum likelihood parameter estimation [20]. Moreover, our EM calibration algorithm can play as a preliminary measure to mitigate the influence of a malicious feedback.

For the parameters in RLM model, our goal is to choose values such that the likelihood of the estimated reputation $\log p(R_{1:N})$ is maximized. But due to the analytical issues, we can only have access to a lower bound of the measure [22], which can be formulated as:

$$\begin{aligned} \log p(R_{1:N}, \mathrm{z}_{1:\mathrm{N}}) &= \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \log p(\mathrm{z_i} \mid R_\mathrm{i}) \\ &+ \textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \log p(R_\mathrm{i} \mid R_{\mathrm{i}-1}) + \log p(R_0) \end{aligned} \tag{8}$$

We need to find the parameters that will maximize the above log-likelihood. However, as the sequence of reputation state R_k has not been observed, this maximization is not tractable directly, so we have to apply the EM algorithm. The EM algorithm transforms the maximization of the above likelihood function to iterations of successive two steps (expectation and maximization), where the reputation state sequence is assumed to be known. In the expectation step, EM computes an expectation of the log likelihood with respect to the current estimate of the reputation value. In the maximization step, EM computes the parameters which can maximize the expected log likelihood.

In our RLM model, one important characteristic is that the reputation feedback contains the attribute: suggested feedback variance, which implies how to aggregate the feedback so that a more accurate reputation prediction can be derived. To take into account the suggested feedback variance c_k, we extend the typical EM algorithm with an initialization step. Thus, after each new feedback $f_k = \{z_k, c_k\}$ becomes available, the EM algorithm will run an iteration that consists of three steps. The final EM equations are:

Initialization Step

$$Q_k = c_k, A_k = 1, W_k = W_{k-1}$$

Expectation Step

$$\sum\nolimits_k = W_k^{-1} + Q_k^{-1} \tag{9}$$

$$\langle R_k \rangle = (W_k^{-1} A_k \langle R_{k-1} \rangle + Q_k^{-1} z_k) / \sum\nolimits_k \tag{10}$$

Maximization Step

$$A_k = (\sum\nolimits_{i=1}^{k} \langle R_i \rangle \langle R_{i-1} \rangle) / (\sum\nolimits_{i=1}^{k} \langle R_{i-1} \rangle^2) \tag{11}$$

$$Q_k = \frac{1}{k} \sum\nolimits_{i=1}^{k} (z_i - \langle R_i \rangle)^2 \tag{12}$$

$$W_k = \frac{1}{k} \sum\nolimits_{i=1}^{k} (\langle R_i \rangle - A_i \langle R_{i-1} \rangle)^2 \tag{13}$$

In the initialization step, the estimated estimated feedback variance Q_k is set to be c_k, meaning that the evaluator has a belief in the suggested feedback variance at first. The reputation state transfer factor A_k is assumed to be 1, meaning that the reputation state does not change. The variance of the reputation transfer noise W_k is set to be the value used in the last aggregation iteration.

In the expectation step, to compute an expectation of the log likelihood, EM computes the expected reputation value $\langle R_k \rangle$ with respect to its conditional distribution. W_k^{-1} and Q_k^{-1} are used to weight the model derived reputation $A_k \langle R_{k-1} \rangle$ and the feedback reputation z_k respectively to calculate $\langle R_k \rangle$ in equation 10. Equation 9 computes $\sum_k = W_k^{-1} + Q_k^{-1}$, which is used as the denominator to get the expected reputation value in equation 10.

In the maximization step, all the dynamic parameters (A_k, Q_k and W_k) are updated to maximize the likelihood expectation. The maximization step can act as a preliminary measure to mitigate the influence of a malicious feedback based on the following theorem.

Theorem 3 Let $f_k = \{z_k, c_k\}$ be a normal feedback to be sent by a node. If the node maliciously changes the feedback reputation value z_k without changing the suggested feedback variance, that is he sends the feedback $f'_k = \{z'_k, c_k\}$ where $z'_k \neq z_k$, then f'_k will have a smaller influence on the reputation value prediction $\langle R \rangle$ than the normal feedback f_k.

Proof: For a malicious feedback f'_k, if it only changes its feedback reputation value z'_k, then the feedback reputation value z'_k will have a bigger deviation from the expected reputation value $\langle R_k \rangle$ than a normal feedback. The bigger deviation $(z'_k - \langle R_k \rangle)$ of a malicious feedback leads to a bigger estimated feedback variance Q_k estimated in Equation (12). Theorem 1 shows that if a feedback has a bigger estimated feedback variance, it will have a smaller influence on the reputation evaluation. Thus, a malicious feedback which only changes its feedback reputation value usually has a smaller influence on the reputation value evaluation than a normal feedback.

Although the EM algorithm can resist part of the malicious feedbacks by creating bigger estimated feedback variance, a malicious node can still manipulate the model by the following model vulnerability. If a malicious feedback sets its suggested feedback variance to be an extremely low value approaching 0, then the EM calibration algorithm will assign a small estimated feedback variance for the feedback according Theorem 4. In such case, no matter how much does the feedback reputation value deviate from the real reputation value, the feedback can still have a high influence on the reputation aggregation based on the proof of Theorem 1.

Theorem 4 Let $f_k = \{z_k, c_k\}$ be a normal feedback to be sent by a node and Q_k and Q'_k denote the estimated feedback variance for f_k and f'_k respectively. If the node maliciously sets the suggested feedback variance to be a lower (resp. bigger) value, that is he sends the feedback $f'_k = \{z_k, c'_k\}$ where $c'_k < c_k$ (resp. $c'_k > c_k$), then we have $Q'_k < Q_k$ (resp. $Q'_k > Q_k$).

Proof: Assuming that there is a feedback f'_k that has a lower suggested feedback variance c'_k than the original one (c_k). In the initialization step of EM algorithm, the estimated feedback variance Q_k of the RLM model is initialized with c'_k. This lower c'_k makes the feedback reputation value z_k account for a larger portion of the predicted reputation value $\langle R_k \rangle$ in Equation (10). Because of the higher dependency between z_k and $\langle R_k \rangle$, the deviation $(z_k - \langle R_k \rangle)$ will be reduced, leading to a smaller estimated feedback variance Q'_k in Equation (12). Similarly, we can prove that if a feedback has a bigger suggested feedback variance, the EM algorithm will give a bigger estimated feedback variance for the feedback.

5.2 Malicious Feedback Detection

In last subsection, we introduced the EM algorithm to give a robust and autonomous parameter calibration. Although the EM algorithm can resist part of the malicious feedbacks by considering their estimated feedback variance, a malicious node can still manipulate the model by setting the suggested feedback variance to be an extremely low value. In this case, the malicious feedback will have a big influence and cause great performance decline to our reputation evaluation.

To make our RLM model robust under such attack, we further introduce the hypothesis test technology to detect the malicious feedbacks. Let H_0 be the hypothesis that the reputation feedback is honest. Recall from section 4 that the Kalman aggregation provides the predicted reputation value $\langle R_k \rangle$ after receiving a feedback $f_k = \{z_k, c_k\}$. In a system without malicious feedbacks, the deviation between $\langle R_k \rangle$ and z_k should follows a zero-mean normal distribution with variance $P_k + Q_k$, where

P_k is yielded by the Kalman aggregation and Q_k is yielded by our EM algorithm.

To detect the malicious feedback, the hypothesis testing simply evaluates whether the deviation between the feedback reputation value and the predicted reputation is normal enough. Given a significance level α, which determines the confidence level of the test, the problem is to find the threshold value t_k such that:

$$P\left(|z_k - \langle R_k \rangle| \geq t_k \,|H_0\,\right) = \alpha \tag{14}$$

Under the hypothesis H_0, $(z_k - \langle R_k \rangle)$ follows a zero-mean normal distribution with variance $P_k + Q_k$, so we can also have that:

$$P\left(|z_k - \langle R_k \rangle| \geq t_k \,|H_0\,\right) = 2 \times \theta\left(t_k \Big/ \sqrt{P_k + Q_k}\right) \tag{15}$$

where $\theta(x) = 1 - \Phi(x)$, with $\Phi(x)$ being the cumulative distribution function (CDF) of a zero-mean unit variance normal distribution. Solving Equations 14 and 15, we can get:

$$t_k = \sqrt{P_k + Q_k}\,\theta^{-1}(\alpha/2) \tag{16}$$

If the deviation between the feedback reputation value and the predicted reputation value exceeds the threshold t_k, then the hypothesis is rejected. Therefore, the feedback is flagged as malicious, and the update of the reputation and the prediction variance is aborted.

In EM calibration algorithm, a malicious feedback can attack the RLM model by setting its suggested feedback variance to be an extremely low value. Theorem 5 demonstrates that the hypothesis test technology can enhance the model to resist such attacks.

Theorem 5 Let $f_k = \{z_k, c_k\}$ be a normal feedback to be sent by a node and let t_k and t'_k denote the test threshold values for f_k and f'_k respectively. If the node maliciously sets the suggested feedback variance to be a lower value, that is he sends the feedback $f'_k = \{z_k, c'_k\}$ where $c'_k < c_k$, then we can get $t'_k < t_k$.

Proof: Assuming that there is a malicious feedback f'_k that gives a lower suggested feedback variance c'_k than the original one. From the proof of Theorem 4, we can find that the lower c'_k will result in a smaller estimated feedback variance Q'_k in Equation (12), which will further lead to a smaller reputation prediction variance P'_k evaluated in Equation (7) based on Theorem 2. In brief, the lower suggested feedback variance c'_k will create a smaller $P'_k + Q'_k$, leading to a smaller test threshold value t'_k in Equation (16). A smaller threshold means that the malicious feedback reputation value cannot deviate from the normal value too much. Thus it will be more difficult for the malicious feedback to pass the hypothesis feedback test.

Finally as shown in Algorithm 1, every node in a network needs to run the reputation evaluation locally upon receiving an indirect feedback in the RLM model. After receiving a feedback, the algorithm outputs the result for the reputation evaluation and malicious feedback detection. Firstly, it initializes the dynamic parameters in lines 3-5, and uses the EM algorithm to get a preliminary parameter estimation in line 6. To detect malicious feedbacks, the algorithm uses the estimated parameters to evaluate the new reputation value and its prediction variance (line 7), and then calculates the malicious feedback threshold according to the hypothesis test (line 8). If the deviation between z_k and $\langle R_k \rangle$ is beyond the threshold (line 9), the feedback is labeled as malicious(line 10), and the update caused by the feedback is abandoned (line 11). Otherwise, the algorithm runs another EM iteration to get a more accurate parameter estimation, and uses the Kalman aggregation method to give the final reputation evaluation $\langle R_k \rangle$ and P_k.

Algorithm 1 Reputation Evaluation Algorithm for RLM

```
1: INPUTS : f_k = {z_k, c_k}, ⟨R_{k-1}⟩, P_{k-1}, W_{k-1}
2: OUTPUTS: ⟨R_k⟩, P_k, W_k, isMalicious
3: accept the suggested feedback variance as the local
   estimated feedback variance Q_k = c_k
4: assume the reputation state does not change A_k = 1
5: set the state transfer variance according to the expe-
   rience W_k = W_{k-1}
6: run an EM algorithm iteration to estimate Q_k, A_k, W_k
   using equations 9-13
7: use the Kalman aggregation to compute ⟨R_k⟩, P_k
   using equations 3-7
8: compute the malicious feedback threshold t_k using
   equation 16
9: if z_k − ⟨R_k⟩ > t_k then
10:   isMalicious = true
11:   ⟨R_k⟩ = ⟨R_{k-1}⟩, P_k = P_{k-1}, W_k = W_{k-1}
12: else
13:   isMalicious = false
14:   run another EM iteration to update Q_k, A_k, W_k
      using equations 9-13
15:   use the Kalman Aggregation to get the final pre-
      diction ⟨R_k⟩, P_k
16: end if
17: return ⟨R_k⟩, P_k, W_k, isMalicious
```

6 Experiments and Results

In this section, we evaluate our RLM trust model in a simulated reputation environment. We do three sets of experiments to assess the validation, accuracy and robustness of our RLM trust model respectively. In our simulation, the reputation about a node is conducted over $N = 1000$ feedback sessions, which constitute a feedback dataset. Over the feedback sessions, the real reputation value R_i of a node changes randomly with a factor f (next reputation value / current reputation value). We assume a wide range [0.6, 1.4] for the factor f so that the RLM model can be tested in a difficult situation. Moreover, the minimum and maximum values of a node's real reputations are set to be 0.1 and 1 respectively.

At each feedback session, as the node's real reputation R_i changes, a new reputation feedback f_i is created. There are two kinds of reputation feedbacks: normal feedback and malicious feedback. Normal reputation feedbacks are created to reflect the opinion of a normal recommender. In real scenarios, because of the incomplete local knowledge, a recommender usually cannot give an exactly accurate feedback. As illustrated in section 3, we simulate the normal feedback reputation value z_i as the real reputation value R_i added by a deviation that follows a zero-mean Gaussian distribution. The variance of the distribution is set to be $k\sigma$, where k is a scaling factor (e.g. k = 1, 2, 3), and σ is the deviation unit. Since the feedback is a subjective inaccurate rating, we set $\sigma = 0.01$, which means a relatively big deviation noise [19]. Hence, when k = 1 (resp. k = 3), each feedback reputation value will have a different deviation that follows a zero mean normal distribution with variance 0.01 (resp. 0.03).

From all the created normal feedbacks, some are selected to be simulated as malicious feedbacks. In the simulation, the malicious feedback probability p_m is a variable (e.g. 10%, 20%, and 30%), so that we can evaluate its influence on the trust prediction. As defined in section 3.2, a malicious feedback can be a random positive/negative feedback or a coordinated positive/negative feedback. In one feedback dataset, all the malicious feedbacks are assumed to be collusive, which means that they are of the same kind.

In our RLM reputation model, besides the feedback reputation value, a feedback also comprises the suggested feedback variance c_i. For an honest recommender, c_i should equal to its estimated prediction variance P_i for the feedback reputation value. An attacker can set c_i to be a value bigger or smaller than P_i. If an attacker sets c_i to be a bigger value (intuitively the attacker suggests that he has less confidence in the feedback, and the feedback reputation value may have a bigger deviation), then the aggregator will assign a bigger estimated feedback variance for the feedback (demonstrated in Theorem 4). Therefore, the feedback will have a smaller influence on the reputation aggregation based on Theorem 1. Meanwhile, it will result in a bigger estimated prediction variance, meaning that the aggregator will be less confident about the reputation aggregation. This is contrary to the intent of a malicious attacker. Hence we assume that an attacker always tries to set the suggested feedback variance as lower as possible than P_i. In this scenario, the malicious feedback can have a bigger influence on the reputation aggregation, and mislead the aggregator to believe in the aggregation with more confidence. Hence in the experiment, the suggested feedback variance of a malicious feedback is set to be a low value 10^{-4}.

6.1 Performance Metrics

To evaluate the accuracy of reputation predictions, we calculate the prediction variance and normalized mean squared error (NMSE) of the predictions given by different trust models. Given N trust predictions, the prediction variance is the their mean square error, which can be defined as:

$$PredictionVariance = \sum\nolimits_{i=1}^{N} (\langle R_i \rangle - R_i)^2/N \tag{17}$$

The NMSE is the mean square error of all the reputation predictions normalized by the variance of the real reputation. It can be calculate as $(\sum_{i=1}^{N} (\langle R_i \rangle - R_i)^2/N)/(\sum_{i=1}^{N} (R_i - (R_i))^2/N)$, hence we can get:

$$NMSE = (\sum\nolimits_{i=1}^{N} (\langle R_i \rangle - R_i)^2)/(\sum\nolimits_{i=1}^{N} (R_i - (R_i))^2) \tag{18}$$

For the comparison of robustness, we use the classical false/true positives/negatives indicators. Specifically, a positive is a malicious reputation feedback which should be rejected by the trust model, and a negative is a normal reputation feedback which should be accepted. The number of positives (resp. negatives) in all the feedbacks is n_p(resp. n_n). A false positive is a normal feedback that has been wrongly labeled as malicious, and a true positive is a malicious feedback that has been correctly detected. The number of false positives (resp. true positives) reported by the trust model is n_{fp}(resp. n_{tp}). The false positive rate (FPR) is the proportion of all the normal feedbacks that have been wrongly detected, thus $FPR = n_{fp}/n_n$. Similarly, the true positive rate (TPR) is the proportion of malicious feedbacks that have been correctly detected, which is $TPR = n_{tp}/n_p$. To detect the malicious feedback, RLM model use the significance level α to decide the confidence (strictness) of the detection. Normally, a higher significance level will increase both the true and false positive rates. According to many experiments in other testing [17,19], a significance level of 5% offers a good compromise between the true and false positive rates. Hence, we also set α as 5% in our experiments.

6.2 Validation of RLM Model

To validate the RLM trust model, we run the model in a clean trust environment with no malicious feedbacks. The RLM model predicts the reputation value of a node, and evaluates the variance of the reputation prediction after each session. Hence, we need to evaluate the fitness of RLM model to represent the reputation value and the reputation prediction variance. First, we set the variance of the feedback deviation to be 1σ, and the malicious feedback probability $p_m = 0$. Fig.4 shows a typical result given by RLM trust model over sessions. The red line denotes the real reputation value of a node at each session, the stars represent the noised reputation feedbacks, and the blue line denotes the reputation value predicted by RLM model. To have a full test about the

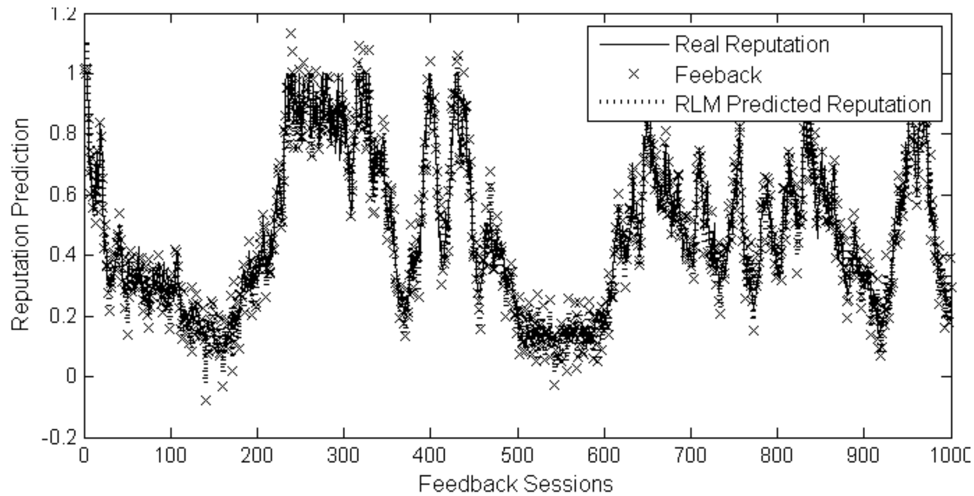


Fig. 4. Sketch map for the real reputation of a node, the reputation feedback and the reputation predicted by RLM model over sessions

model performance, the real reputation value evolves randomly with a big change factor over the sessions. A smooth reputation change will be much easier for the trust models, hence, it is not tested in our experiment. We can find that although the feedbacks are not exactly accurate, the RLM model can still give a good reputation prediction, which is so close to the real reputation that their two curves are indistinguishable at most of the sessions. Fig.5 plots the prediction error between the real reputation and RLM predicted reputation. Most of the prediction errors are less than 0.06, which demonstrates that the RLM trust model can capture the real reputation effectively.

The RLM model also gives an estimation of the reputation prediction variance P, which can be called the RLM estimated prediction variance. To test the fitness of the estimated prediction variance, we compute the real prediction variance between the predicted reputation value and the real reputation value. Fig.6 shows that the RLM trust model has a high efficiency to estimate the prediction variance. The curves of the RLM estimated prediction variance and the real prediction variance are close except at the initial 200 sessions. This is because the RLM model is initialized with some constant parameters, so it needs some time to stabilize.

6.3 Accuracy of RLM Model

For the accuracy test, we compare our RLM model with two other typical general trust models: summation model [1] and Bayesian model [8]. The summation model is widely used in commercial services like eBay, and it can be used in a specific environment like the Engentrust in P2P networks. Since our RLM trust model is a general model without considering the underlying architecture, we implement a pure summation model for comparison. Based on the Beta distribution, the Bayesian model computes the reputation by two parameters: α and β, indicating the number of positive and negative results.

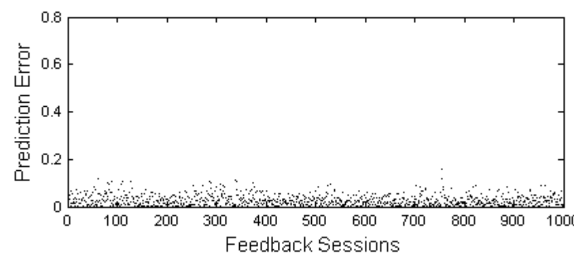


Fig. 5. Prediction errors given by RLM model

We do two experiments for the accuracy test in a clean trust environment. In the first experiment, the variance of the feedback deviation is 1σ, and the three trust models (Summation, Bayesian and RLM) are tested with the same feedback input. Fig.7 plots the cumulative distribution function of the prediction errors given by these three models. We can see that the majority errors given by RLM model are less than 0.1, while the errors given by the summation and Bayesian models spread to 0.2. Hence, we can get the conclusion that the RLM model has the best prediction accuracy, and the Bayesian model is slightly better than the summation model.

In the second experiment, the variance of the feedback deviation is set to be 1σ, 2σ and 3σ respectively. We compute the prediction variance between the real reputation value and the reputation value predicted by each trust model. Since the Bayesian model is more accurate than the summation model, Fig.8 only compares the result of Bayesian and RLM trust models. We can see that, under all the cases, the prediction variance given by RLM model is smaller than the Bayesian model. In particular, RLM model achieves a considerably higher improvement ratio (of about 50%) for prediction accuracy when the variance of the feedback deviation is small (1σ), as compared to when the variance of the feedback deviation is big (3σ). This is because the RLM model calibrates the parameters with the maximum likelihood estimation, which is hugely influenced by the feedback deviation. Hence, as the feedback deviation increases,

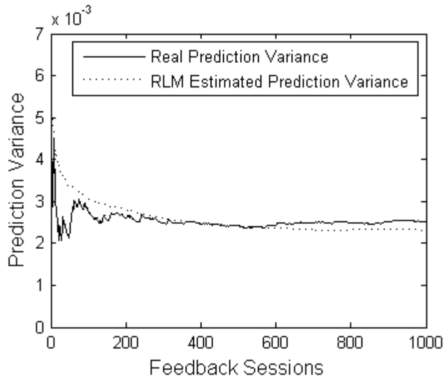


Fig. 6. Real and estimated prediction variance

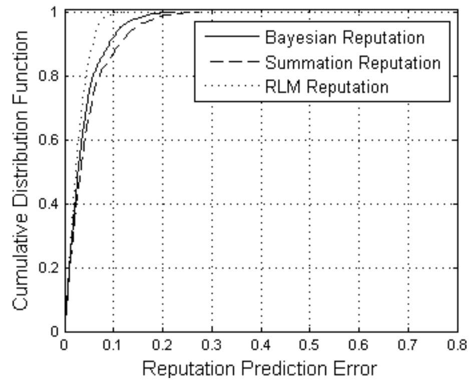


Fig. 7. CDF of reputation prediction errors

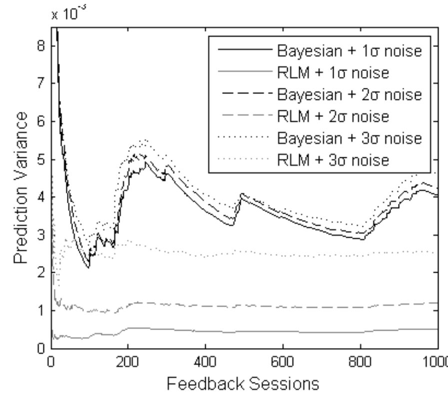


Fig. 8. NMSE of the trust models over sessions

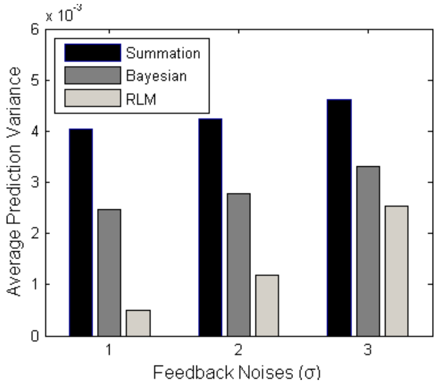


Fig. 9. Prediction variance with different reputation feedback noises

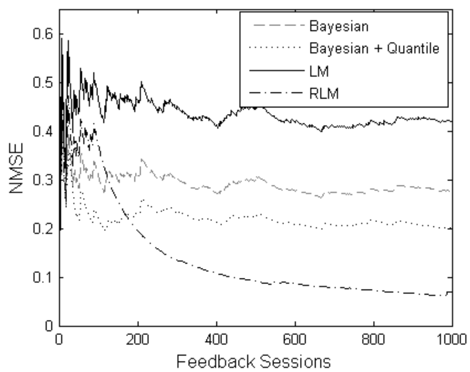


Fig. 10. Average prediction variance with different reputation feedback noises

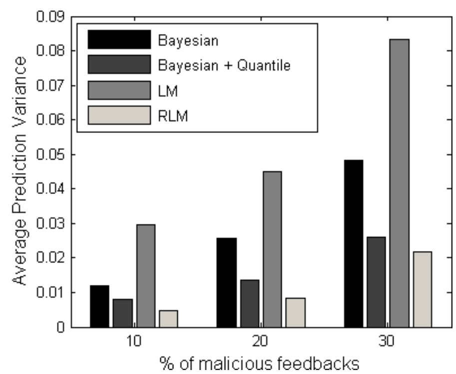


Fig. 11. NMSE of the trust models with malicious feedbacks

the accuracy benefits of RLM model will be reduced. In Fig.8, the result comes from only one trial (each method running on one feedback dataset). In Fig.9, we compute the average prediction variance of the three methods running over five trials. It confirms the result that, compared with the summation and Bayesian models, the RLM trust model can give a more accurate reputation prediction, especially when the feedback deviation is small.

6.4 Robustness of RLM Model

In last two subsections, we examined the validation and accuracy of the trust model in a clean trust environment. Next, we evaluate the robustness of RLM trust model under the attack of malicious feedbacks. To resist the malicious feedback, Whitby et. al [9] introduced the quantile filtering method based on the Bayesian reputation system. They filtered out a feedback if it is outside the q quantile and $(1-q)$ quantile of the Beta distribution for the reputation. The quantile filtering is an intuitive solution without guarantee of any quantitative confidence about the filtering. In contrast, based on RLM model, our hypothesis test method can filter out a feedback with a specific confidence level α through the statistical theory. For the comparison, the Bayesian trust model with quantile malicious feedback filtering is called the Bayesian + Quantile model, and we set the q as 0.01 which is a good choice as reported in [9]. Beside the Bayesian + Quantile model, we also test the robustness of the RLM trust model without the hypothesis test technology, which is called LM trust model.

Firstly, the feedback dataset is created with random positive/negative feedbacks, and the malicious feedback probability p_m is set to 20%. We run the pure Bayesian model, the Bayesian + Quantile model, the LM model and the RLM trust model on the same feedback dataset. For LM and RLM models, the suggested feedback variance of the malicious feedback is set to be a low value 10^{-4}, so that all malicious feedbacks can have a big threat to the models. Fig.10 plots the normalized mean squared errors given by the four models. Within the initial 100 feedback sessions, the performances of all the four models are not stable. Then, the RLM trust model gradually reaches the smallest NMSE, meaning that RLM model has the best prediction performance under the attack. Fig.10 also shows that the RLM model without the hypothesis test technology is highly vulnerable to the malicious feedback with low suggested feedback variance. Unsurprisingly, the Bayesian model with quantile filtering has a better performance than the pure Bayesian

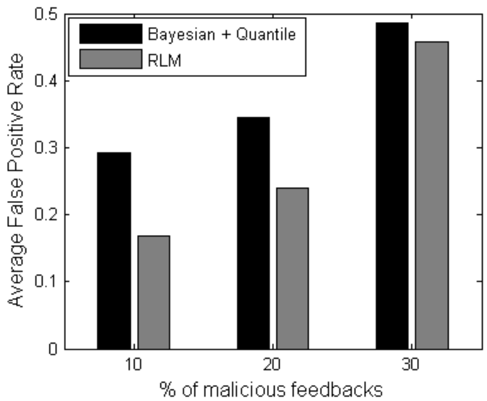


Fig. 12. Average prediction variance with malicious feedbacks

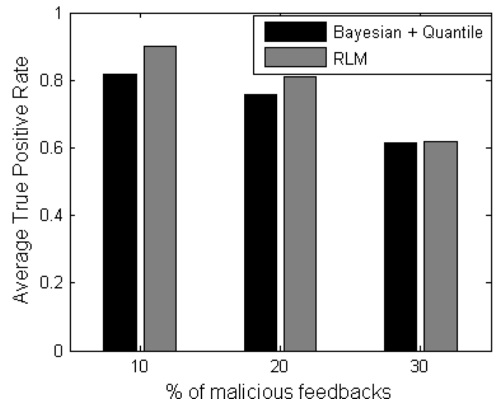


Fig. 13. Average FPR of the different detection methods

trust model.

Next, we set the malicious feedback probability p_m as 10%, 20% and 30% respectively. With each p_m value, we create five feedback datasets, so that we can get the representative average result for each case. Fig.11 plots the average prediction variance given by the four trust models. It confirms the result that the RLM model has the best prediction performance under the attack. Compared with Bayesian + Quantile model, the prediction variance given by RLM model is much smaller (26% on average). In addition, we can observe that, when the probability p_m gets close to 30%, all the four models have a huge performance decline.

Both the Bayesian + Quantile trust model and our RLM model try to detect malicious feedbacks. Therefor based on last experiment, we evaluate the detection efficiency of the different models by comparing their false/true positive rate (FPR/TPR). Fig.12 and Fig.13 show that, with all the different malicious feedback probabilities, RLM model has better detection performance than the Bayesian + Quantile model. In particular, when the malicious feedback probability p_m is low (10%), RLM model has a significantly lower false positive rate (0.12 on average) and a higher true positive rate (0.09 on average) than the Bayesian + Quantile model. When the probability p_m is high (30%), the performance advantage of RLM model decreases, with a lower false positive rate (0.03 on average) and an almost same true positive rate. This demonstrates that RLM model has higher detection accuracy than Bayesian + Quantile model. However, as the malicious feedbacks probability increases, RLM's accuracy advantage will decrease.

In the last three experiments, we use the random positive/nagative feedback model to simulate malicious feedbacks. Next, the coordinated positive/nagative feedbacks are simulated, which try to give adulating or defaming reputation values coherent to each other. We compare the detection performance of RLM model under attacks of random and coordinated malicious feedbacks. The two kinds of malicious feedbacks are created with probabilities 10%, 20%, and 30%, and their suggested feedback variance is set to a low value 10^{-4}. Fig. 14 and Fig. 15 show that with all the different malicious feedback probabilities, the coordinated malicious feedback has more severe influence on RLM model than the random malicious feedback. Under the attack of coordinated feedbacks, RLM model has a higher FPR and lower TPR, which illustrates that the coordinated feedback makes it more difficult to detect a malicious feedback. We can also find that as the malicious feedback probability grows, the performance gap of RLM model under the two attacks increases. In particular, coordinated feedbacks result in a considerably larger performance gap (of about 20%) for both FPR and TPR when the malicious feedback probability is high (30%), as compared to when the probability is low (10%). This shows that coordinated feedbacks are more efficient to hide malicious behaviors of the attacker.

7 CONCLUSION

Reputation based trust systems can play a vital role in service selection and promoting service providers to improve their service quality. In this paper, we introduced the Robust Linear Markov (RLM) model for trust representation and aggregation. To get a more comprehensive and accurate reputation evaluation, we defined the reputation by two attributes: reputation value and reputation prediction variance. The assessment of the reputation prediction variance can help to achieve a better local decision making and a more intelligent third-party reputation aggregation. For feedback aggregation, we introduced the novel Kalman aggregation method, which supplies a support to defend malicious feedbacks through the parameter: estimated feedback variance. To defend against the adulating/defaming and random/coordinated malicious feedbacks, we first introduced the Expectation Maximization algorithm, which can autonomously tune the parameters to mitigate the influence of a malicious feedback. Then, we further enhanced the model by the hypothesis test method to resist malicious feedbacks precisely with a specific confidence level. We also demonstrated the robustness of our RLM

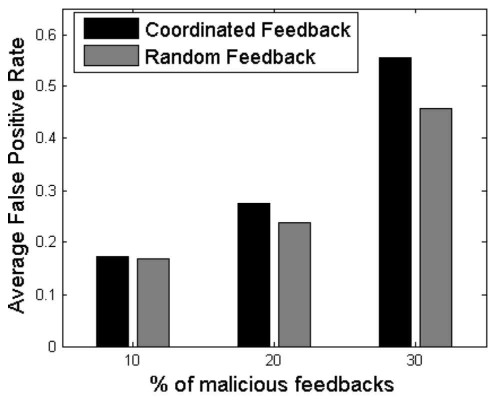


Fig. 14. Average FPR of RLM model under different attacks

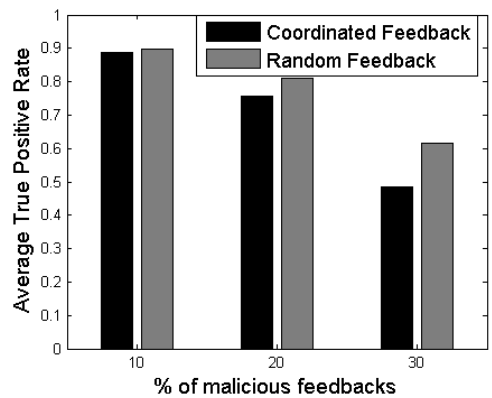


Fig. 15. Average TPR of RLM model under different attacks

model through theoretical analysis. Simulation results show that the RLM model can efficiently capture the reputation value and its prediction variance. Compared with two other typical trust models (i.e., the summation and Bayesian model), our RLM model can give a more accurate reputation prediction with the minimum prediction error. Under the attack of malicious feedbacks, the RLM model can give the best reputation prediction with the smallest NMSE and prediction variance. Moreover, it has higher malicious detection accuracy (lower false positive rate and higher true positive rate) than the Bayesian + Quantile method.

In future, we will investigate how to apply our general RLM model in a specific environment such as service-oriented, P2P and social network environments. In addition, base on inference theory [18,19], we will introduce more robust inference technologies to our model to resist the malicious feedback attack.

ACKNOWLEDGMENTS

The authors would like to thank Bin Dai, Yiming Zhang and Chee Shin Yeo for their comments. The first and third authors are partially supported by Aid program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province. The second author acknowledges the partial support by grants from NSF CyberTrust, NSF NetSE, IBM SUR, a grant from Intel Research Council.

REFERENCES

[1] A. Jϕsang, R. Ismail, and C Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, 43(2):618-644, 2007.
[2] J. Golbeck. Weaving a Web of Trust. Science, 321(5896): 1640 - 1641, 2008.
[3] K. Hoffman, D. Zage and C. Nita-Rotaru, A Survey of Attack and Defense Techniques for Reputation Systems, ACM Computing Surveys. 14(4), 2009.
[4] D. Stern, R. Herbrich and T. Graepel, Matchbox: Large Scale Online Bayesian Recommendations, 18th international conference on World Wide Web (WWW), 2009.
[5] L. Xiong and L. Liu, PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities, IEEE Trans. Knowledge and Data Eng, 16(7): 843-857, 2004.
[6] R. Zhou and K. Hwang, PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing, IEEE Trans. on Parallel and Distributed Systems, 18(5): 460-473, 2006.
[7] S. Song, K. Hwang, and Y.K. Kwok, Risk-Resilient Heuristics and Genetic Algorithms for Security-Assured Grid Job Scheduling, IEEE Trans. on Computers, 55(6):703-719, 2006.
[8] Y. Zhang and Y. Fang, A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks?IEEE Trans. on Parallel and Distributed Systems, 18(8): 1134 - 1145, 2007.
[9] A. Whitby, A. Jϕsang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. Int. Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS), 2004.
[10] Y. Wang and M. P. Singh, Trust representation and aggregation in distributed agent systems, Int. Conference on Artificial Intelligence (AAAI), Boston, 2006.
[11] B. Yu and M. P. Singh, Detecting Deception in Reputation Management, Int. Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS), 2003.
[12] U. Kuter and J. Golbeck, SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models, Int. Conference on Artificial Intelligence (AAAI), 2007.
[13] J. Golbeck and J. Hendler, Inferring Trust Relationships in Web-based Social Networks, ACM Transaction on Internet Technology, 6(4): 497-529, 2006.
[14] M. Raya, P. Papadimitratos, V.D. Gligor, J.P. Hubaux, On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks, In Proceedings of IEEE Infocom, 2008.
[15] S. Kamvar, M. Schlosser, and H. Garcia-Molina, The Eigentrust Algorithm for Reputation Management in P2P Networks, 12th international conference on World Wide Web (WWW), May 2003.
[16] M. Srivatsa, L. Xiong and L. Liu, TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks, 14th international conference on World Wide Web (WWW), May 2005.
[17] J.M. Morris, The Kalman filter: A robust estimator for some classes of linear quadratic problems, IEEE Transactions on Information Theory, 22(5): 526-534, 1976.
[18] C. Atkeson, A. Moore and S. Schaal. Locally weighted learning. AI Review, 11:11-73, April 1997.
[19] P.S. Maybeck, Stochastic models, estimation, and control. Mathematics in Science and Engineering, Volume 141, Academic Press, 1979
[20] A. Dempster, N. Laird and D.Rubin, Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society. Series B 39(1): 1-38, 1977.
[21] W.Y. Chen, D. Zhang and E.Y. Chang, Combinational Collaborative Filtering for Personalized Community Recommendation, ACM International Conference on Knowledge Discovery and Data Mining, 2009.
[22] J. Ting, A. D'Souza and S. Schaal, Bayesian regression with input noise for high dimensional data, ACM Proceedings of the 23rd International Conference on Machine Learning, 2006.
[23] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. Experimental Economics, 9(2): 79-101, 2006.

[24] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, K. Nahrstedt, A Trust Management Framework for Service-Oriented Environments, 18th international conference on World Wide Web (WWW), 2009.
[25] R. Andersen, C. Borgs, J. Chayes, U. Feige, etc. Trust-Based Recommendation Systems: an Axiomatic Approach, 17th international conference on World Wide Web (WWW), 2008.
[26] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, A near-optimal social network defense against sybil attacks? in Proceedings of the 2008 IEEE Symposium on Security and Privacy, 2008.
[27] Z. Liang, W. Shi, Analysis of Ratings on Trust Inference in Open Environments, Elsevier Performance Evaluation, 65(2): 99-128, 2008.
[28] X. Wang, W. Ou, J. Su, A reputation inference model based on linear hidden markov process, ISECS International Colloquium on Computing, Communication, Control, and Management, 2009.

Xiaofeng Wang received the B.S. degree, M.S. degree and Ph.D. degrees from the National University of Defense Technology (NUDT) in 2004, 2006 and 2010, respectively, all in school of computer. Between Nov. 2007 to Nov. 2008, he was a visiting student at Cloudbus lab, the university of Melbourne, with the China Scholarship Council's Fellowship. Since march 2010, he has been a research assistant in school of computer, NUDT. His current research interests are in distributed computing, trust management and network security.

Ling Liu is a full Professor in the School of Computer Science, College of Computing, at Georgia Institute of Technology. There she directs the research programs in Distributed Data Intensive Systems Lab (DiSL), examining various aspects of data intensive systems with the focus on performance, availability, security, privacy, and energy efficiency. Dr. Liu has published over 250 International journal and conference articles in the areas of databases, data engineering, and distributed computing systems. She is a recipient of the best paper award of ICDCS 2003, WWW 2004, the 2005 Pat Goldberg Memorial Best Paper Award, and the best data engineering paper award of Int. conf. on Software Engineering and Data Engineering 2008. Dr. Liu is currently on the editorial board of several international journals, including Distributed and Parallel Databases (DAPD, Springer), International Journal of Web Services Research, and Wireless Network (WINET, Springer). Dr. Liu's current research is primarily sponsored by NSF, IBM, and Intel.

Jinshu Su received his B.S degree of mathematics from Nankai University, 1985, and his M.S, and Ph.D degrees from National University of Defense Technology (NUDT) in 1988 and 2000 respectively, both in Computer Science. Currently, he is a full professor in School of Computer, and serves as head of the Institute of network and information security, NUDT. He has lead several national key projects of CHINA, including one national 973 projects, several national 863 projects and NSFC Key projects. Pro. Su is a member of ACM and IEEE, a senior member of CCF(China Computer Federation). He has published more than 70 papers in international journals and conferences, including ICDCS 06, Infocom 08, Mobihoc 08, CCGrid 09 and ICPP 10 etc..