Securing Online Reputation Systems Through Trust Modeling and Temporal Analysis

Yuhong Liu, *Member, IEEE*, Yan (Lindsay) Sun, *Member, IEEE*, Siyuan Liu, and Alex C. Kot, *Fellow, IEEE*

***Abstract*—With the rapid development of reputation systems in various online social networks, manipulations against such systems are evolving quickly. In this paper, we propose scheme TATA, the abbreviation of joint Temporal And Trust Analysis, which protects reputation systems from a new angle: the combination of time domain anomaly detection and Dempster–Shafer theory-based trust computation. Real user attack data collected from a cyber competition is used to construct the testing data set. Compared with two representative reputation schemes and our previous scheme, TATA achieves a significantly better performance in terms of identifying items under attack, detecting malicious users who insert dishonest ratings, and recovering reputation scores.**

***Index Terms*—Information security, social network, information filtering.**

I. Introduction

As MORE people use the Internet for entertainment, building personal relationships, and conducting businesses, the Internet has created vast opportunities for online interactions. However, due to the anonymity of the Internet, it is very difficult for normal users to evaluate a stranger's trustworthiness and quality, which makes online interactions risky. Is a piece of news on Reddit true? Does a product at Amazon.com have high quality as described? Is a video on YouTube really interesting or informative? In most cases, the answers can hardly be predicted before the interactions are committed. The problem is how the online participants protect themselves by judging the quality of strangers or unfamiliar items beforehand.

To address this problem, online reputation systems have been built up. The goal is to create large-scale virtual word-of-mouth networks where individuals share opinions and experiences, in terms of reviews and ratings, on various *items*, including products, services, digital contents and even other people. These opinions and experiences, which are called users' *feedback*, are collected as evidence, and are analyzed, aggregated, and disseminated to general users. The disseminated results are called *reputation score*. Such systems are also referred to as *feedback-based reputation systems*.

Online reputation systems are increasingly influencing people's online purchasing/downloading decisions. For example, according to comScore Inc., products or services with a 5-star rating could earn 20% more than products or services with a 4-star rating could [1]. More and more people refer to Yelp rating system before selecting hotels and restaurants; to Amazon product ratings before purchasing products online; to YouTube video ratings before viewing a video clip; and etc. Furthermore, a recent survey indicates that around 26% of adult Internet users in the U.S. have rated at least one item through online reputation systems [2].

Meanwhile, driven by the huge profits of online markets [3], diverse *manipulations* against online reputation systems are evolving rapidly. Many sophisticated programs are developed to automatically insert feedback. Furthermore, some reputation management companies even control large affiliate networks of real user IDs to provide "rating services" for their customers. For about $750, a company named "VideoViralViews.com" [4] can provide 100 real user ratings to a piece of music on iTunes. For just $9.99, a video on YouTube could receive 30 "I like" ratings or 30 real user comments provided by "IncreaseYouTubeViews.com". Without proper defense schemes, attacks against reputation systems can overly inflate or deflate the item reputation scores, crash users' confidence in online reputation systems, eventually undermine reputation-centric online businesses and lead to economic loss. Securing online reputation systems is urgent.

In this paper, we propose a reputation defense scheme, named TATA, for feedback-based reputation systems. Here, TATA is the abbreviation of joint Temporal And Trust Analysis. It contains two modules: *a time domain anomaly detector* and *a trust model based on the Dempster–Shafer theory*. Specifically, we consider the ratings to a given item as a time sequence, and a time domain anomaly detector is introduced to detect suspicious time intervals where anomaly occurs. A trust analysis is then conducted based on the anomaly detection results. We borrow the concept of user *behavior uncertainty* from the Dempster–Shafer theory to model users' behavior patterns, and evaluate whether a user's rating value to each item is reliable or not.

The performance of TATA, two other representative reputation schemes [5], [6] and our previously proposed scheme TAUCA [7] is evaluated against real user attack data collected through a *cyber competition*. TATA demonstrates significant advantages in terms of identifying items under attack, detecting malicious users who insert dishonest ratings, and recovering reputation scores.

Manuscript received June 14, 2012; revised November 05, 2012; accepted December 24, 2012. Date of publication January 11, 2013; date of current version May 16, 2013. This work was supported by NSF award #0643532. The associate editor coordinating the review of this manuscript and approving it for publication was C.-C. Jay Kuo.

Y. Liu and Y. Sun are with the Department of Electrical and Computer Engineering University of Rhode Island, Kingston, RI 02881 USA (e-mail: yuhong@ele.uri.edu; yansun@ele.uri.edu).

S. Liu and A. C. Kot are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: lius0036@ntu.edu.sg; eackot@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2013.2238929

The rest of the paper is organized as follows. Section II discusses the related work, Section III describes the system and attack models, Section IV introduces the details of TATA, and Section V presents the experiment results, followed by the conclusion in Section VI.

II. Related Work

As diverse manipulations against reputation systems appear and develop rapidly, defense schemes protecting reputation systems are also evolving accordingly. In this section, we roughly divide them into four categories.

In the first category, the defense approaches limit the maximum number of ratings each user could provide within a certain time duration [8]. Such type of approaches actually restrict the rating power of each user ID. This can prevent the attackers from inserting a large amount of dishonest ratings through a few user IDs within a short time.

In the second category, the defense schemes aim to increase the cost of launching an attack. Some reputation systems in practice, such as Amazon, assign higher weights to users who commit real transactions. This method can effectively increase the cost to manipulate competitors' item reputation. However, it has little impact on attacks in which attackers buy their own products for reputation boosting. Some other schemes increase the costs of acquiring multiple user IDs by binding identities with IP addresses [9] or using network coordinates to detect sybil attacks [10]. Such schemes will greatly increase the attack costs, but cannot defeat the attackers with plenty of resources. For example, some reputation boosting companies [4] often acquire a large affiliate network of user IDs.

In the third category, the defense approaches investigate rating statistics. They consider ratings as random variables and assume dishonest ratings have statistical distributions different from normal ratings. Representative schemes are as follows. A Beta-function based approach [5] assumes that the underlying ratings follow Beta distribution and considers the ratings outside q (lower) and $(1-q)$ (upper) quantile of the majority's opinions as dishonest ratings. An entropy based approach [11] identifies the ratings that bring a significant change in the uncertainty of the rating distribution as dishonest ratings. In [12], dishonest rating analysis is conducted based on Bayesian model. Controlled anonymity and cluster filtering are used to eliminate dishonest ratings in [13].

The defense approaches in the fourth category investigate users' rating behaviors. Assuming that users with bad rating history tend to provide dishonest ratings, such approaches determine the weight of a rating based on the reputation of the user who provides this rating. Such reputation is also referred to as trust or reliability. Several representative schemes are as follows. Iteration refinement approach proposed in [6] assigns weights to a user's ratings according to the inverse of this user's rating variance. In [14], a personalized trust structure is introduced so that different users may assign different trust values to the same user. In [15], a user's trust is obtained by accumulating neighbors' beliefs through belief theory [16]. REGRET reputation system, proposed in [17], calculates user reputation based on fuzzy logic. Flow models, such as EigenTrust [18] and Google PageRank [19], compute trust or reputation by transitive iteration through looped or arbitrarily long chains.

Although many schemes have demonstrated very good performance in protecting reputation systems, there are still limitations that are not fully addressed. **First**, time domain, which contains rich information, is not fully exploited. The current approaches address the time factors in two ways. In the first way, all the ratings are treated equally and the time when these ratings are provided is ignored. In the second way, recent ratings are given larger weights when computing the reputation scores. These simple approaches neglect the great potential of investigating time-domain information. **Second**, most defense schemes in the third category follow the "majority rule", which detects dishonest ratings by examining whether some rating values are far away from the majority's opinions. This rule works fine when the attackers insert a small number of dishonest ratings that are very different from normal users' rating values. However, it may (1) generate misleading results when the number of dishonest ratings is large and (2) yield high false alarm rate when normal ratings have a large variance and dishonest ratings are not too far away from the majority's opinions. **Third**, schemes in the fourth category, trust based approaches, are relatively vulnerable to attacks where malicious users conduct good and bad behaviors alternatively. Malicious users could first accumulate high trust values by providing normal ratings to the items that they do not care and then provide dishonest ratings to the items that they want to manipulate. **Fourth**, to evaluate a reputation system, the researchers need data representing malicious attacks. However, it is extremely difficult to obtain attack data from real systems mainly because there is no ground truth indicating whether particular ratings are from attackers or not. The real human users can create multifaceted, coordinated, and sophisticated attacks that are not well understood yet. Thus, the lack of realistic attack data can hurt the performance evaluation.

In this work, we propose a reputation defense scheme, TATA. The objective of the proposed scheme is to (1) detect the malicious users who provide dishonest ratings; (2) recover reputation score of the target item, that receives dishonest ratings; and (3) avoid interference to normal items' reputation scores. Specifically, TATA is a combination of an anomaly detector, which belongs to the third category, and a Dempster–Shafer theory based trust model, which belongs to the fourth category. Different from the "majority rule" based schemes in the third category, the anomaly detector in TATA detects dishonest ratings from a new angle: investigating time domain information. To further reduce false alarms caused by normal ratings with large variance, the Dempster–Shafer theory based trust model is introduced. Different from most trust methods in the fourth category, the proposed trust model, instead of assigning each user an overall trust value, evaluates a user's reliability on different items separately. In this way, the attackers cannot easily avoid detection by accumulating high trust values on the items that they do not care. Furthermore, a cyber competition was held to collect real user attack data for testing data construction, which makes the performance evaluation more realistic and convincing. Last but not lease, TATA is compatible with many defense schemes in the first and second category.

III. Models and Assumptions

In this section, we discuss the system model, attack model and basic assumptions used in this paper.

System Model: We model the feedback-based reputation systems as the system in which *users* provide ratings to *items*. This model can describe many practical systems. For example, buyers provide ratings to products on Amazon.com, and readers rate social news on Reddit.com. The items in above systems are products and social news, respectively. We consider that each user will provide rating to one item at most once, and the rating values are integer values ranging from 1 to 5. In practice, reputation systems often allow users to provide reviews as well. These reviews can also be untruthful. In this paper, we focus on the detection of dishonest ratings. The analysis of untruthful reviews is beyond the scope of this paper, whereas the dishonest rating detection and untruthful review detection complement each other.

Attack Model: An *attacker* can control one or multiple user IDs and each of these user IDs is referred to as a *malicious user*. Malicious users provide ratings to manipulate the reputation score of items. The item whose reputation score is manipulated by malicious users is called a *target item*. The ratings provided by malicious users to target items are considered as *dishonest ratings*. An *attack profile* describes the behavior of all malicious users controlled by the attacker.

Assumptions: In this work, we assume that items have intrinsic quality, which does not change rapidly. The rating values to a given item depend on the users' personal preference as well as the item quality. In some applications, such as ratings for movies or books, the item quality judgement is very subjective and users' personal preference plays a more important role, whereas in some other applications, such as Amazon product ratings, the item quality plays a more important role. In this work, we focus on the product-rating type applications, where the rating distribution of an item is relatively stable. Therefore, if *rapid* changes in the rating distribution occur, it is possible that anomaly happens.

Furthermore, we notice that due to personal preference, normal users sometimes may also provide "biased ratings" that are far away from the real quality of the items. Meanwhile, to avoid being detected by reputation defense schemes, malicious users may imitate normal users' behaviors by providing honest ratings to the items that they do not care. We call these ratings as "spare ratings". We assume that most of the ratings from normal users can reflect the real quality of the items, whereas malicious users who have limited rating resources would mainly focus on rating target items and can provide "spare ratings" to few other items. This is also observed in the attack data in the cyber competition.[1] The trust module of the proposed scheme TATA is built up based on this assumption and may not well differentiate normal users from malicious users in the two following scenarios.

- Scenario I, attackers provide a large number of "spare ratings". This type of malicious users is extremely difficult to detect but usually cause small damage to the reputation systems. In other words, if malicious users must provide a large number of "spare ratings" to avoid detection when a defense scheme is used, this defense scheme successfully increases the cost of conducting attacks.
- Scenario II, a normal user with very few ratings has provided a "biased rating", and this "biased rating" leads to a change in the item's rating statistics. This case is very rare since a single rating value usually cannot cause an obvious change in the rating statistics.

[1]Available: http://www.ele.uri.edu/nest/cant.html

IV. Joint Temporal and Trust Analysis (TATA)

A. Overview

The proposed TATA scheme contains two components: (a) a time domain anomaly detector and (b) a trust model based on the Dempster–Shafer theory.

In TATA, we propose to detect anomaly from a new angle: analyzing time domain information. Specifically, we organize the ratings to a given item as a sequence in the descending order according to the time when they are provided. This sequence, denoted by y, actually reflects the rating trend to the given item. As mentioned in Section III, in practice, many items have intrinsic and stable quality, which should be reflected in the distribution of normal ratings. If there are rapid changes in the rating values, such changes can serve as indicators of anomaly. Therefore, we propose a **change detector** in TATA as the anomaly detector, which takes the rating sequences as inputs and detects changes occurring in the rating sequences. The proposed change detector will detect not only sudden rapid changes but also small changes accumulated over time. In this way, even if malicious users insert dishonest ratings with small shifts to gradually mislead items' reputation scores, such type of changes will still be accumulated and finally be detected by the proposed change detector. If the change detector is triggered by an item, the time intervals in which the changes occur are called *change intervals*.

However, the change intervals may still contain normal ratings. Therefore, we introduce the **trust analysis** module.

- Instead of assigning a user with an overall trust value, the proposed trust model evaluates each user's reliability on different items separately. It can reduce the damage from the malicious users who aim to accumulate high trust values by providing "spare ratings" to uninterested items.
- Furthermore, based on the Dempster–Shafer theory, the proposed trust model introduces user *behavior uncertainty*. In this way, a user could yield high trust values only if the user's behavior yields a sufficient amount of good observations.

Finally, the users with low trust values will be identified as malicious users and their ratings to the detected target items will be removed. The remaining ratings are used to calculate the item reputation.

Fig. 1 demonstrates the structure of TATA. The design details will be discussed in the rest of this section.

B. Temporal Analysis—Change Detector

Many change detectors have been developed for different application scenarios [20], [21]. In online reputation systems, since normal ratings do not necessarily follow a specific distribution and attackers may insert dishonest ratings with small bias, we need to choose a change detector that is insensitive to

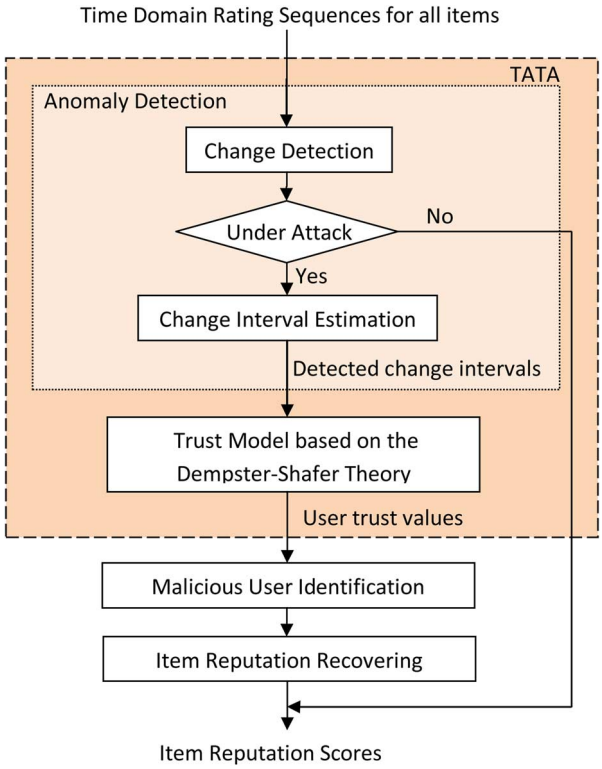


Fig. 1. System architecture.

the probability distribution of data and is able to reliably detect small shifts. Therefore, we choose the CUSUM detector [21], which fulfills these requirements, as the base to build our change detector.

1) Basic CUSUM: We first introduce the basic CUSUM detector, which determines whether a parameter θ in a probability density function (PDF) has changed. That is, to determine between two hypothesis: $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$. Let p_{θ_0} and p_{θ_1} denote the PDF before and after the change, respectively. Let y_k denote the k^{th} sample of the data sequence (i.e. rating sequence). The basic CUSUM decision function is

$$g_k = max\left(g_{k-1} + \ln\frac{p_{\theta_1}(y_k)}{p_{\theta_0}(y_k)}, 0\right), \quad (1)$$

$$t_a = min\{k : g_k \geq \overline{h}\}, \quad (2)$$

where $\overline{h}$ is the change detection threshold. Here, t_a is called *stopping time*, the time when the detector identifies a change and raises an alarm. Each time when $g_k \leq 0$ or $g_k \geq \overline{h}$, CUSUM detector restarts by setting $g_k = 0$ and a new round of detection begins.

When p_{θ_0} is a Gaussian process with mean μ_0, p_{θ_1} is a Gaussian process with mean μ_1, and both have variance σ^2, (1) detects the mean change and becomes

$$g_k = max\left(g_{k-1} + \left(y_k - \mu_0 - \frac{\mu_1 - \mu_0}{2}\right), 0\right). \quad (3)$$

We use Gaussian distribution here for two reasons. First, there is no common acknowledgement about the distribution of the rating sequence for one item. Second, according to [22], even if the distributions are not Gaussian, the above detector is still sensitive to the mean change.

2) Revised CUSUM: The basic CUSUM detector could sensitively detect changes occurring in the time domain. However, it cannot be directly applied to our problem for two reasons. **First**, basic CUSUM does not provide a way to estimate the starting time of a change. In the basic CUSUM, once the detection function (g_k) exceeds the threshold $(\overline{h})$, an alarm is issued and the CUSUM detector restarts. The stopping time (t_a) denotes the time when the change has been detected but not the actual change starting time. We need to trace back on the change detection function g_k to estimate the change starting time. **Second**, the change may last for some time. Restarting the detector will make it impossible to trace the change ending time.

Therefore, we develop a revised CUSUM detector to estimate the change starting time and ending time. The revised CUSUM can detect multiple change intervals. For a specific change interval, let t_s denote the *starting time* of the change and t_e denote the *ending time* of the change.

We define *off-time*, denoted by t_b, as the time when g_k falls below threshold $\overline{h}$. That is,

$$t_b = min\{k : g_k < \overline{h}\ \&\ k \geq t_a\} \quad (4)$$

Based on the detection curve (g_k), we can obtain t_a and t_b, as the time when g_k goes above and falls below the threshold $\overline{h}$, respectively. The next task is to estimate t_s and t_e, which define a specific change interval.

We first determine the time interval where t_s and t_e lie in. As discussed above, a change starts earlier than the time when the change detector is triggered. Therefore, we define t_1 as the *counting-start-time*. If there is no previously detected change, we set $t_1 = 0$. If there is a previously detected change, whose off-time is t_b^{pre}, we set $t_1 = t_b^{pre}$. The change starting time t_s lies between t_1 and t_a. Furthermore, the change ending time t_e lies between t_s and t_b. Based on these, we derive t_s and t_e as follows.

- *Starting time of the change interval* (t_s)
  Assume that the data sequence follows distribution p_{θ_0} before t_s, and p_{θ_1} after t_s. Let c denote the index of data samples. Then maximum likelihood estimator (MLE) of t_s is derived as

$$\hat{t_s} = \arg\max_{t_1 \leq c \leq t_a} \ln\left[\prod_{i=t_1}^{c-1} p_{\theta_0}(y_i)\prod_{i=c}^{t_a} p_{\theta_1}(y_i)\right]$$

$$= \arg\max_{t_1 \leq c \leq t_a}\left[\ln\prod_{i=t_1}^{t_a} p_{\theta_0}(y_i) + \ln\frac{\prod_{i=c}^{t_a} p_{\theta_1}(y_i)}{\prod_{i=c}^{t_a} p_{\theta_0}(y_i)}\right] \quad (5)$$

$$= \arg\min_{t_1 \leq c \leq t_a}\sum_{i=t_1}^{c-1}\ln\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)} \quad (6)$$

  Notice that (6) is derived from (5) using the fact that $\ln\prod_{i=t_1}^{t_a} p_{\theta_0}(y_i)$ is a fixed term. The estimated starting time $\hat{t}_s$ is between t_1 and t_a.
- *Ending time of the change interval* (t_e)
  Next, we estimate the ending time t_e using the MLE estimator. Here, we assume that the data sequence follows distribution p_{θ_0} between t_1 and t_s, follows distribution p_{θ_1}

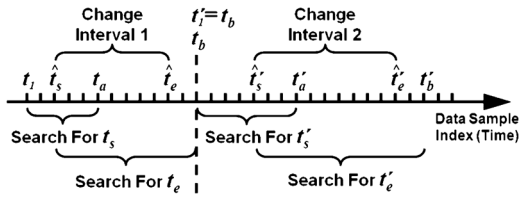


Fig. 2. Illustration of the revised CUSUM.

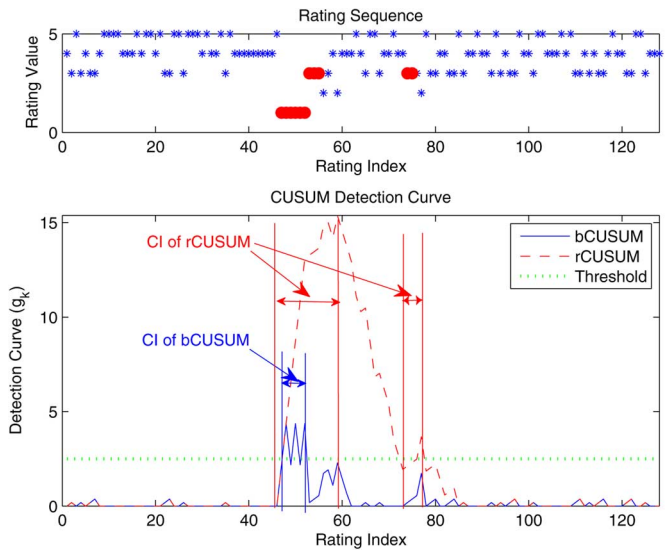


Fig. 3. Demonstration of CUSUM detection curve. (CI: change interval; rCUSUM: revised CUSUM; bCUSUM: basic CUSUM).

between t_s and t_e, and follows distribution p_{θ_0} between t_e and t_b. Let d denote the index of data samples.

$$\hat{t_e} = \arg\max_{\hat{t_s} \le d \le t_b} \ln\left[\prod_{i=t_1}^{\hat{t_s}-1} p_{\theta_0}(y_i) \prod_{i=\hat{t_s}}^{d} p_{\theta_1}(y_i) \prod_{i=d+1}^{t_b} p_{\theta_0}(y_i)\right] \quad (7)$$

$$= \arg\max_{\hat{t_s} \le d \le t_b} \left[\ln \prod_{i=\hat{t_s}}^{t_b} p_{\theta_1}(y_i) + \ln \frac{\prod_{i=d+1}^{t_b} p_{\theta_0}(y_i)}{\prod_{i=d+1}^{t_b} p_{\theta_1}(y_i)}\right] \quad (8)$$

$$= \arg\max_{\hat{t_s} \le h \le t_b} \sum_{i=d+1}^{t_b} \ln \frac{p_{\theta_0}(y_i)}{p_{\theta_1}(y_i)} \quad (9)$$

We use the fact that $\ln \prod_{i=t_1}^{\hat{t_s}-1} p_{\theta_0}(y_i)$ is a fixed term to derive (8) from (7), and the fact that $\ln \prod_{i=\hat{t_s}}^{t_b} p_{\theta_1}(y_i)$ is a fixed term to derive (9) from (8). Fig. 2 illustrates the relationship between t_1, t_a, t_b, t_s and t_e.

Fig. 3 illustrates the performance of the basic CUSUM and the revised CUSUM. The x axis is the index of ratings. The upper plot shows the original rating sequence ordered according to the time when the ratings are provided. The y axis is the rating value ranging from 1 to 5. The normal ratings are marked as stars, whereas the dishonest ratings are marked as circles. The lower plot shows the detection curves (g_k) of the basic CUSUM and the revised CUSUM, as well as the detection threshold ($\overline{h}$). The *change intervals* are defined by $\hat{t_s}$ and $\hat{t_e}$. Two observations are made.

- Once the detection curve goes above the threshold, the revised CUSUM will trace back for estimation of the change starting time. Compared to the basic CUSUM, which counts changes starting at the time when the detection curve exceeds the threshold, the revised CUSUM yields a more accurate change starting time.
- Due to restarting, the basic CUSUM misses the real change ending time. Furthermore, each time when the basic CUSUM restarts, the accumulation of changes also restarts, leading to the ignorance of some small changes. Compared to the basic CUSUM, the revised CUSUM is more sensitive to small changes.

Procedure 1 summarizes the revised CUSUM detector. Line 6–8 are determining whether an item is under attack. If yes, (2), (4), (6) and (9) are used to determine the change intervals, as presented in Line 9–23 in Procedure 1.

Procedure 1 Change detection procedure

1: $Under\_Attack = []$ //the set containing items under attack
2: **for** each item I_i **do**
3: collect all ratings for I_i and order them according to the time that they are provided
4: $Alarm = 0$ //a flag, which is 1 when g_k exceeds threshold
5: Compute $g_i = \{g_i(1), g_i(2), \cdots g_i(N)\}$
6: //Under attack or not
7: **if** $(max(g_i) > \overline{h})$ **then**
8: add I_i to $Under\_attack[]$
9: //Get change starting time and ending time
10: **for** each rating k **do**
11: **if** $(Alarm == 0)$ **then**
12: **if** $(g_i(k) > \overline{h})$ **then**
13: Estimate change starting time t_s
14: $Alarm = 1$
15: **end if**
16: **else**
17: **if** $(g_i(k) < \overline{h})$ **then**
18: Estimate change ending time t_e
19: $Alarm = 0$
20: **end if**
21: **end if**
22: **end for**
23: **end if**
24: **end for**

As a summary, the revised CUSUM detector is used to (a) detect whether changes occur in the ratings of an item and (b) estimate the time intervals (i.e. change intervals) in which attacks are suspected.

C. Trust Model Based on the Dempster–Shafer Theory

We define users who provide ratings during the detected change intervals as *suspicious users*. Not all suspicious users are malicious users because normal users may occasionally provide "biased ratings" due to personal reasons or even human errors. Therefore, we propose to further differentiate normal users from malicious users by trust analysis.

In most trust models, users' trust values are determined only by their good and bad behaviors. However, it is not sufficient. Consider two trust calculation scenarios. First, user A has conducted 5 good behaviors and 5 bad behaviors. Second, user B is

a new coming user and has no behavior history. In several trust models [5], [12], both of their trust values will be calculated as 0.5, although we are more confident in user A's trust value. To differentiate these two cases, the concept of behavior uncertainty is introduced by the Dempster–Shafer theory, to represent the degree of the ignorance of behavior history. In this work, we adopt the behavior uncertainty by proposing a trust model based on the Dempster–Shafer theory.

1) The Dempster–Shafer Theory Framework: Before discussing the proposed trust model, we first introduce the Dempster–Shafer theory. The Dempster–Shafer theory [23] is a framework for combining evidence from different sources to achieve a degree of belief. It has introduced the concept of "uncertainty" by allowing the representation of ignorance.

We use a binary case to show the basic definitions in the Dempster–Shafer theory. Let the frame of discernment $\Theta = \{a, b\}$ be two possible events under consideration (e.g., $a = $ good behavior, $b = $ bad behavior). The power set $2^{\Theta} = \{\{a\}, \{b\}, \{a, b\}, \emptyset\}$ is the set of propositions regarding the event that has actually happened. Here, $\{a, b\}$ represents ignorance, meaning that it is uncertain regarding the event that has actually happened.

Definition 1: Let Θ be a frame of discernment. A function m: $2^{\Theta} \rightarrow [0, 1]$ is defined as a Basic Belief Assignment (BBA) when it satisfies the following two properties:

1) $m\{\emptyset\} = 0$;
2) $\sum_{A \in 2^{\Theta}} m(A) = 1$.

Then for the binary case, we have $m(\{a\}) + m(\{b\}) + m(\{a, b\}) = 1$.

Definition 2: The belief (Bel) for a set A is defined as the sum of all the assignments of the subsets of A:

$$Bel(A) = \sum_{B \subseteq A} m(B).$$

For the binary case, we have the following equations:

$$\begin{aligned} Bel\left(\{a\}\right) &= m\left(\{a\}\right), \\ Bel\left(\{b\}\right) &= m\left(\{b\}\right), \\ Bel\left(\{a, b\}\right) &= m\left(\{a\}\right) + m\left(\{b\}\right) + m\left(\{a, b\}\right). \end{aligned}$$

$Bel(A)$ can be considered as the least probability that one event belonging to A will happen. For instance, for a given BBA, if $m(\{a\}) = 0.9$, $m(\{b\}) = 0$, $m(\{a, b\}) = 0.1$, we will have $Bel(\{a\}) = 0.9$, $Bel(\{b\}) = 0$, $Bel(\{a, b\}) = 1$, meaning that the subject will perform good behavior with at least 0.9 probability, perform bad behavior with at least 0 probability, and perform either good behavior or bad behavior with 1 probability.

2) Mapping Between Evidence Space and Belief Space: In subjective logic [24], a mapping method has been defined between the observed evidence space and the belief space. Suppose two events $\{a, b\}$ are under consideration, where $a = $ good behavior and $b = $ bad behavior, and a subject is observed to perform good behaviors for r times, and perform bad behaviors for s times. Then a mapping between the observed evidence space and the belief space is defined as:

$$\begin{cases} B_g = \frac{r}{r+s+2}, \\ B_b = \frac{s}{r+s+2}, \\ u = \frac{2}{r+s+2}, \end{cases}$$

where B_g is the belief that the proposition that the subject will perform good behavior is true, B_b is the belief that the proposition that the subject will perform bad behavior is true, and u is the uncertainty.

According to belief definition in the Dempster–Shafer theory, we will have:

$$\begin{cases} Bel\left(\{a\}\right) = \frac{r}{r+s+2}, \\ Bel\left(\{b\}\right) = \frac{s}{r+s+2}, \\ u = \frac{2}{r+s+2}, \end{cases}$$

3) Trust Model Using the Dempster–Shafer Theory: Before describing the proposed trust model, we first introduce some important concepts used in our model.

- **Behavior Value**: We define a user's *behavior value* on a single item as a binary value to indicate whether his/her rating behavior is good or bad. When user u_j provides a rating to item I_i, if his/her rating falls into the change intervals detected by the change detector described in Section IV-B, the behavior value of u_j for item I_i, denoted by $Beh_{u_j}(i)$, is set to 0. Otherwise, $Beh_{u_j}(i)$ is set to 1. In this work, we assume that each user will rate one item no more than once, therefore, each user has only one behavior value on each item that he/she has rated.
- **Combined Behavior Value**: To evaluate users' behaviors on multiple items, we introduce the *combined behavior value*, which is computed based on the discussion in Section IV-C4. For example, if user u_j has rated $r + s$ items, where the behavior values for r items are 1 and for s items are 0, u_j's combined behavior value on these $r + s$ items is calculated as $Beh_{u_j}^{com} = r/(r + s + 2)$.
- **Behavior Uncertainty**: Similarly, we define a user's *behavior uncertainty* using the Dempster–Shafer theory. In the above example, the behavior uncertainty of user u_j who provides r normal ratings and s suspicious ratings is calculated as $Beh_{u_j}^{uncer} = 2/(r + s + 2)$. Based on this calculation, the more ratings user u_j has provided, the less uncertain his/her behavior will be.

Using these basic concepts, we build the trust model as follows. Suppose that a user u_j has rated M items (i.e. $I_1, I_2, \ldots, I_i, \ldots, I_M$) in total. We would like to calculate the trust value of user u_j on item I_i, which indicates how much we could trust the rating provided by user u_j to item I_i. Let us consider this issue from two perspectives.

From the first perspective, not considering user u_j's rating to item I_i, purely based on his/her ratings to other items, how much could we trust him/her in providing an honest rating? To answer this question, we need to first calculate the combined behavior value of user u_j on the $M - 1$ items (i.e. $I_1, \ldots I_{i-1}, I_{i+1}, \ldots I_M$), which is denoted by $Beh_{u_j}^{com}(i)$. Assume that among these $M - 1$ items, r is the number of items where u_j has behavior value 1, and s is the number of items where u_j has behavior value 0. We have

$$Beh_{u_j}^{com}(i) = \frac{r}{r + s + 2} \tag{10}$$

Then, we calculate the behavior uncertainty as:

$$Beh_{u_j}^{uncer}(i) = \frac{2}{r + s + 2}. \tag{11}$$

Next, how much can we trust on user u_j based on his/her ratings to the $M-1$ items? This trust value is denoted by $T_{u_j}^{com}(i)$ and calculated by

$$T_{u_j}^{com}(i) = Beh_{u_j}^{com}(i) * \left(1 - Beh_{u_j}^{uncer}(i)\right). \quad (12)$$

From the second perspective, when consider only user u_j's rating to item I_i, how much could we trust him/her? Similarly, the trust value $T_{u_j}^{I_i}$ is calculated as

$$T_{u_j}^{I_i} = Beh_{u_j}(i) * Beh_{u_j}^{uncer}(i). \quad (13)$$

Recall that $Beh_{u_j}(i)$ is the behavior value of user u_j on item I_i. Equation (13) indicates that the more uncertain user u_j's behavior on other items, the larger weight we will assign to his/her behavior on item I_i.

The total trust of user u_j on item I_i is computed as

$$T_{u_j}(i) = T_{u_j}^{com}(i) + T_{u_j}^{I_i}. \quad (14)$$

As a summary, after the first module: anomaly detection, we propose a trust model based on the Dempster–Shafer theory, which has the following features.

- Instead of assigning an overall trust value for each user, we evaluate users' trust values on each items that they have rated. The advantages can be viewed from two aspects. First, normal users with a few "biased ratings" will only have lower trust values on the items to which they provide the "biased ratings", and these lower trust values will not directly affect their normal ratings on other items. Second, although malicious users may keep high trust values on the items to which they provide "spare ratings", their trust values on the target items can be very low.
- When calculating a user's trust value on a specific item, we consider user behaviors from two perspectives: the behavior on this item and the behaviors on the rest of items. As a consequence, it is harder for malicious users to gain high trust on the target items through "spare ratings". Let us compare the proposed model with the well-known beta-function based trust model in [5]. Assume that u_j conducts 1 dishonest rating to the target item and 5 honest ratings (i.e. spare ratings) to other items. Using the beta trust model [5], the overall trust of this user is $(5+1)/(5+1+2) = 0.75$. Note that most trust models only calculate an overall trust value. Using the proposed model, the u_i's trust on the target item is $0*(2/(5+2))+(5/(5+2))*(5/(5+2)) = 0.51$. Obviously, if the malicious user wants to gain high trust, he/she needs to insert much more "spare ratings" when the proposed trust model is used.
- The introduction of behavior uncertainty makes it possible to further differentiate users' trust values by the observation number of their behavior history. To obtain high trust values, users have to conduct a sufficient number of good behaviors.

D. Malicious User Identification and Rating Aggregation

Finally, we examine the trust values of each user. We detect the users with low trust values on items as malicious users. Instead of removing all the ratings provided by the malicious users, we only remove their ratings that yield low trust values. Specifically, for user u_j, if $T_{u_j}(i) < T_h$, user u_j's rating to item I_i is removed and u_j is marked as malicious user. Here, T_h is called the trust threshold, which could be adjusted according to different application scenarios. In our testing data, most normal users provide more than 10 normal ratings, while malicious users provide less normal ratings. Based on the proposed trust model, if a user has provided 1 suspicious rating to item I_i, and 10 normal ratings to other items, his/her trust value on item I_i is calculated as $0*(2/12)+(10/12)*(10/12) = 0.694$. Therefore, we choose the T_h as 0.69 in the experiments below, so that a malicious user has to provide at least 10 other normal ratings to cover his/her dishonest rating to the target item. After the rating removal, our method is compatible with any existing reputation schemes that calculate the item quality reputation. Without loss of generality, we use simple averaging in this paper for computing item quality reputation.

V. Experiment Results and Discussion

A. Experiment Setup

1) Cyber Competition: For any online reputation systems, it is very difficult to evaluate their attack-resistance properties in practical settings due to the lack of realistic attack data. Even if one can obtain data from e-commerce websites, there is usually no ground truth about which ratings are dishonest. To understand human users' attack behaviors and evaluate the performance of reputation systems against nonsimulated attacks, we designed and launched a Cyber Competition[1] running from 05/12/2008 to 05/29/2008. Before the competition, we collected real online rating data for 300 products from a famous e-commerce website[2] in China. Such data contained 300 users' ratings to these products from day 1 to day 150. We considered this data as normal rating data and built up a virtual reputation system based on this data.

The web-based cyber competition ran for 18 days on our server. Each player who participated in the competition downloaded the normal rating data from the server, analyzed the normal rating data using his/her own knowledge and methods, and submitted his/her attack methods, also known as attack profiles, to the server. The server computed the effectiveness of the attack profiles and evaluated the performance of each player. The best performing players won cash awards. Some details are as follows.

- A player could control at most 30 malicious user IDs to provide less than 100 ratings in total. The player's goal was to downgrade the reputation score of a product (i.e. item) I_1.
- A player could make many submissions. In each submission, also called an attack profile, the player used a specific number of malicious users (denoted by N_a) and a specific number of ratings (denoted by N_r). Here, N_a ranged from 1 to 30 and N_r ranged from 1 to 100. Each malicious user could rate any of those 300 products, and could not rate one product more than once. An attack profile described the rating behavior (i.e. which products to rate, when to rate, and the rating values) of all malicious users.

[2]Available: www.douban.com

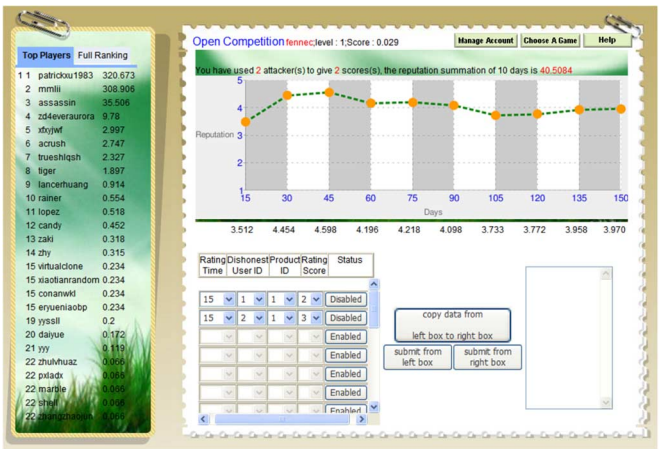


Fig. 4. Screenshot of the CANT cyber competition.

- Once an attack profile was submitted, the competition server simulated a reputation system in which malicious users inserted ratings, as described by the attack profile, into the normal ratings. This simulated reputation system had a simple defense method for removing ratings that were far away from the majority's opinions. The players did not know about this specific defense method. The competition server calculated the reputation score of product I_1 every 15 days. In other words, 10 reputation scores were calculated on day 15, day 30, ... and day 150. The average of these 10 reputation scores, denoted as Rep_{att}, was the overall reputation of I_1.
- The effectiveness of each attack profile was measured by the bias introduced by the attack ratings in the reputation of I_1. The bias was computed as $|REP_{org} - REP_{att}|$, where REP_{org} and REP_{att} were the overall reputation scores of item I_1 calculated before and after inserting the ratings from malicious users, respectively. The attack profile leading to larger bias was more effective.
- The competition rule encouraged the players to adjust the *attack resources* that they used, including the number of malicious user IDs and the number of ratings from the malicious user IDs. Each player could submit many attack profiles. The attack profiles using the same attack resources were compared. If one attack profile was stronger (i.e. more effective) than any other profiles using the same attack resource, this attack profile was marked as the *winning profile*. The performance of each player was measured by the number of winning profiles among all of his/her submissions. The detailed description of the evaluation criteria can be found at http://www.ele.uri.edu/nest/cant.html [25] and a screenshot is shown in Fig. 4.
- The competition had attracted 630 registered players from 70 universities in China and the United States. We collected 826,980 valid submissions.

2) Test Data Preparation: Since we have collected a large amount of attack data, it is important to classify them and construct representative data set for testing purpose. In this paper, we construct two data sets for testing.

First, we group attack profiles according to the number of malicious user IDs used in each profile (i.e. N_a), and then select 6 groups with $N_a = 5$, 10, 15, 20, 25, and 30, respectively. Since there are 300 normal users in the system, in these 6 groups of data, the malicious users have taken up 1.6%, 3.2%, 4.8%, 6.2%, 7.7% and 9% of the total user number. They are selected to represent attacks with very small, small, medium, large and very large number of malicious users. Attacks with a larger number of malicious users usually have stronger attack power and may cause larger attack impact. We want to test whether TATA could yield a consistent defense performance under different attack scenarios. Let $A^I_{set\_N_a}$ denote the set of attack profiles using N_a malicious users. Thus, our first testing data set, denoted by A^I_{set}, contains $A^I_{set\_5}$, $A^I_{set\_10}$, $A^I_{set\_15}$, $A^I_{set\_20}$, $A^I_{set\_25}$, and $A^I_{set\_30}$. There are totally 103,054 attack profiles in A^I_{set}.

Second, we construct another data set that contains only **strong attacks**. To address the attack strength when there are defense mechanisms, we define *attack power* (AP) as $|REP_{org} - REP_{att}|$, where REP_{org} and REP_{att} are the reputation scores calculated before and after inserting the ratings from malicious users, respectively. The reputation is calculated by reputation system employed in the cyber competition.

We construct attack data subset $A^{II}_{set\_N_a}$ as follows. Among all attacks using the same number of malicious users (i.e. N_a), we first pick the attack with the largest AP value, and then pick other attacks whose AP values are greater than 80 percent of this largest AP. All of the picked attacks are put into $A^{II}_{set\_N_a}$. We construct $A^{II}_{set\_N_a}$ for $N_a = 5$, 10, 15, 20, 25, and 30. These six data sets are collectively referred to as A^{II}_{set}, which contains 18,175 attack profiles.

We would like to point out that the reputation scheme used in this cyber competition contains a simple combination of dishonest rating detection and rater trust evaluation. Briefly speaking, it uses the philosophy that ratings far away from the majority's opinions are dishonest ratings. A user's trust is evaluated according to the distances between his/her rating values and the item reputation scores calculated by the reputation system. The details can be found in http://www.ele.uri.edu/nest/cant.html. Many of the reputation defense schemes in category 3 and category 4, as discussed in Section II, use the similar philosophy. The players did not know the specific reputation system used in the cyber competition. We have observed that the players tried diverse attacks in the competition. 826,980 valid attack profiles have been collected in total. We believe that A^I_{set} and A^{II}_{set} can well represent the attack behaviors against the majority of reputation systems from real human users.

B. Performance Testing of TATA

In this subsection and the next subsection, we conduct performance testing of TATA. In Section V-B, the Receiver Operating Characteristic (ROC) curve of malicious users and target items are presented. Furthermore, the performance of TATA in terms of recovering the reputation score of the target item is also demonstrated. In Section V-C, TATA is compared with two representative reputation schemes and our previously proposed scheme TAUCA. Testing data set A^I_{set} is used in Section V-B, and A^{II}_{set} is used in Section V-C.

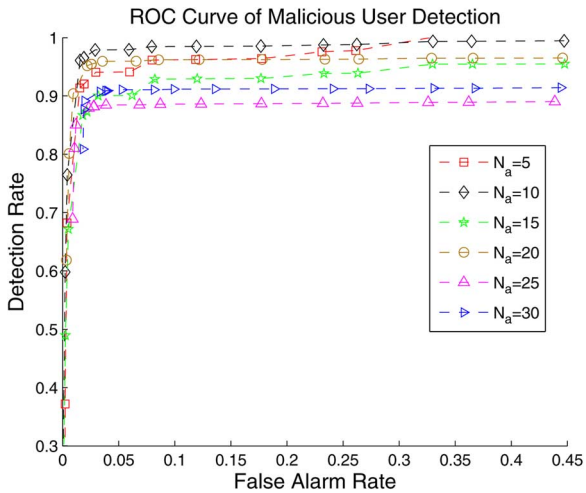


Fig. 5. Performance of malicious user detection for different malicious user number in TATA.

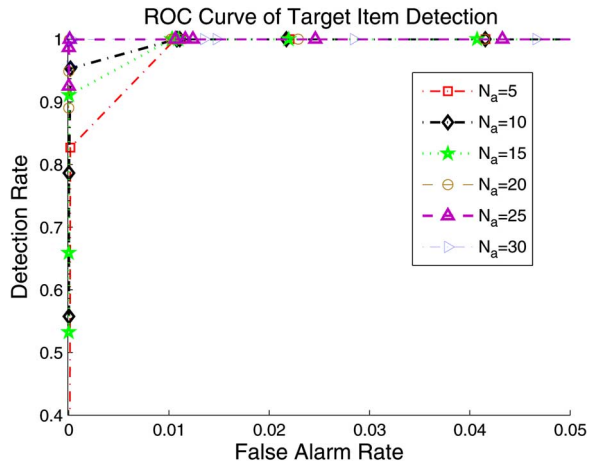


Fig. 6. Performance of the target item detection in TATA.

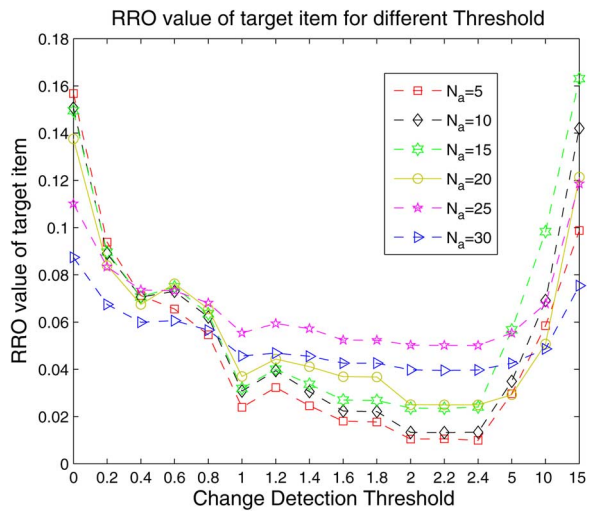


Fig. 7. Recovered reputation offset of the target item for different threshold values in TATA.

Identification of Malicious Users: In the first experiment, we demonstrate the detection rate and false alarm rate of TATA in terms of detecting malicious users. Fig. 5 shows the malicious user detection ROC curves for different N_a values. TATA has demonstrated a consistent good performance. When the false alarm rate is around 5%, TATA yields high detection rate (i.e. $>$88%) for all attacks with different number of malicious users. Note that, the ROC curves for the attacks with the number of malicious users less than 25 are slightly better than that for the attacks with a larger number of malicious users. The reason is that, when attackers have more attack resources (i.e. more malicious IDs), more complex attack strategies can be applied. For example, some malicious users may directly provide dishonest ratings to the target item, while some others may accumulate high trust values to assist the former malicious users and at the same time, keep themselves hidden.

Identification of the Target Item: We then evaluate the performance of TATA in terms of determining the items under attack (i.e. target items). In TATA, we identify the items on which the malicious users have low trust values as target items. In the cyber competition, the ground truth is one item (i.e. I_1) under attack and 299 items not under attack.

Assume that TATA is tested against N attack profiles. Let n_1 denote the number of profiles, for which TATA accurately detects I_1 as the target item. Then, the *detection rate* is defined as $DR = n_1/N$. Let m_1 denote the total number of items that are not under attack but detected as target items. Then, the *false alarm rate* is calculated as $FA = m_1/(299 * N)$.

Fig. 6 shows the ROC curves for detecting target items for different N_a values. We observe that TATA can accurately detect target items. For example, with very small false alarm rates (i.e. $= 0.01$), the detection rates are always above 99% for all attacks with different N_a values.

Recovered Reputation Offset of the Target Item: The detection rate of malicious users cannot fully describe the performance of TATA. Obviously, the amount of damage caused by different malicious users can be very different. We care more about whether the undetected malicious users can cause large damage to the final reputation scores. Therefore, we define the *Recovered Reputation Offset* (RRO) as $|Y - Z|$, where Y is the average value of all the normal ratings to I_1 and Z is the average value of the remaining ratings after TATA has removed dishonest ratings from the detected malicious users. A small RRO value represents that the recovered reputation score is close to the original reputation, which is desirable. Fig. 7 shows the RRO values of the target item (i.e. I_1) against attacks with different N_a when the change detector threshold ($\overline{h}$) is changing.

From Fig. 7, we can make three observations. First, the performance is not sensitive to attack scenario changes. For all the attacks with different N_a, TATA yields similar RRO values for the target item. Second, TATA is not sensitive to the change detection threshold (i.e. $\overline{h}$) selection. For all different attacks, the RRO values of the target item do not change too much when $\overline{h}$ falls between $1 \sim 2.4$. Third, when $\overline{h}$ from 0 to a high value (i.e. 15), the RRO value of the target item initially drops and then starts to grow, forming a roughly concave curve. Based on this curve, the optimal threshold which yields the minimum RRO value could be determined. As shown in Fig. 7, the optimal threshold value for our data falls in between $2 \sim 2.4$. When the optimal threshold is chosen, the maximum RRO value of the target item for all different attacks is 0.05. This means that the attacker can only reduce the reputation score of item I_1 from its true reputation (i.e. Y) to $Y - 0.05$. When simple averaging

method is used, the reputation score of I_1 is 3.98. The attack, where 30 malicious users provide ratings to item I_1 with value 1, will cause the largest damage to the reputation of I_1, which misleads the reputation score from 3.98 to 3.39. This means that TATA reduces the bias in the final reputation score by a factor of $(3.98 - 3.39)/0.05 = 11.8$. As a summary, TATA is a very effective method to protect feedback-based reputation systems.

C. Performance Comparison

In this section, TATA is compared with the Iteration Refinement (IR) model [6] and the Beta function model [5], two existing and representative defense schemes for reputation systems. Furthermore, our previous scheme TAUCA is also compared to TATA. The detailed comparison results are shown below. As mentioned earlier, the data set A_{set}^{II} is used in this part of experiments.

1) Iteration Refinement Model: The IR scheme [6] estimates an item's reputation score as the weighted average of all ratings to this item. The weight assigned to a rating is determined as $w = V^{-\beta}$, where V is the rating variance of the user who provides this rating, and β, ranging from 0 to 1, is the key parameter. When β increases, more ratings from users with large rating variance will be ignored. The main purpose of the IR scheme is reputation recovery.

Fig. 8 shows the RRO values when scheme IR is applied with different β values. The x axis represents the value of β, and the y axis represents the RRO values of the target item. It is seen that for all different attacks, IR can yield very low RRO, when β is properly chosen. For example, when $\beta = 0.8$, the RRO is 0.02.

2) Beta Function Model: The beta scheme assumes that the underlying ratings follow Beta distribution and considers the ratings outside q (lower) and $(1 - q)$ (upper) quantile of the majority's opinions as dishonest ratings. Here, q can be viewed as a sensitivity parameter roughly describing the percentage of feedback being classified as dishonest. The main purpose of the Beta scheme is both malicious rating detection and reputation recovery.

Fig. 9 shows the RRO values when the beta-function based defense method [5] is applied with different q values. The x axis represents the value of q, and the y axis represents the RRO values of the target item. Based on the results in Fig. 9, the best q value is chosen as 0.2, which yields the minimum RRO for different N_a values.

3) TAUCA: In our previous work [7], we proposed a scheme TAUCA, which combines the time domain change detector with user correlation analysis. The major difference between TATA and the previous scheme TAUCA is that we replace the user correlation module by the trust analysis module. The key parameter in TAUCA is also the threshold of the change detector (i.e. $\overline{h_c}$). During our previous study, the optimal threshold value is 2 for this data set. We compare the performance of the previous scheme TAUCA with TATA when the threshold values for both schemes are optimized. Similar to the Beta function scheme, the design goal of TAUCA is malicious rating detection and reputation recovery.

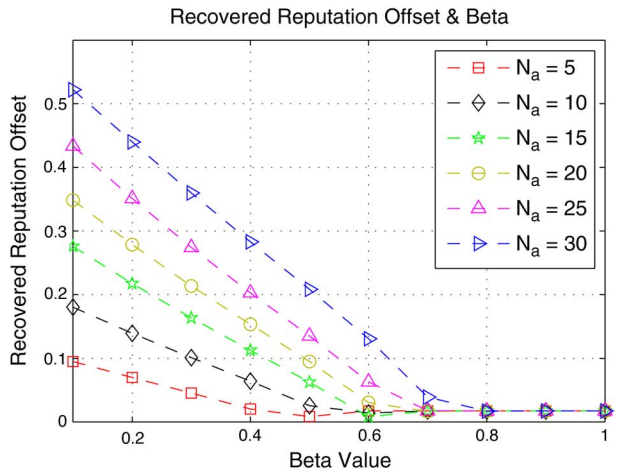


Fig. 8. Performance of IR algorithm for different N_a and β.

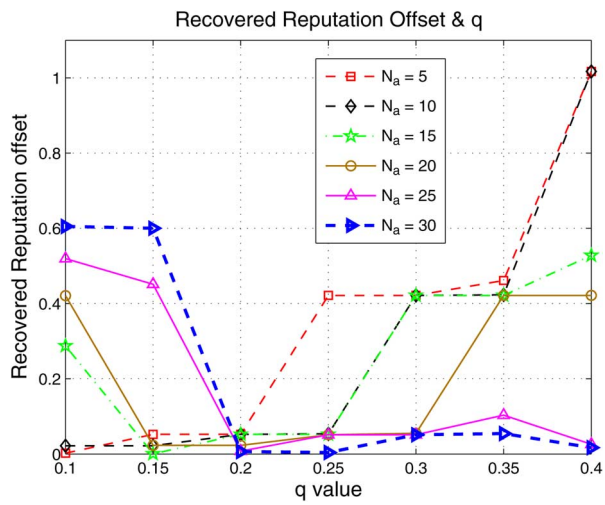


Fig. 9. Performance of beta model for different N_a and q.

4) Performance Comparison Among Different Schemes: In this section, we compare the performance of different schemes: scheme Beta, scheme IR, scheme TAUCA and the proposed scheme TATA. Note that, for the beta scheme and the proposed scheme TATA, there are two optional factors: (1) the forgetting factor which assigns larger weight to more recent ratings and (2) the discounting factor, which is used to control the weight of ratings according to users' reputation. In our comparison, we do not consider these two factors for both the beta function based scheme and the proposed scheme. The forgetting factor is not considered since we assume that (1) each user can provide no more than one rating to each item and (2) items have intrinsic quality which does not change rapidly. Meanwhile, the discounting factor is not considered since it (1) may mislead item reputation when a large portion of normal users are identified as malicious users and (2) is vulnerable against the attackers who accumulate high reputation by providing honest ratings to items that they do not care. Furthermore, both the beta scheme and the proposed scheme can use similar forgetting or discounting mechanism, and we would expect that either the forgetting mechanism or the discounting mechanism has similar impact on both schemes and will not change the comparison results.

TABLE I
MALICIOUS USER DETECTION

N_a	5	10	15	20	25	30
Beta_DR	0.3660	0.6128	0.5452	0.6680	0.7238	0.5345
TAUCA_DR	0.8552	0.9456	0.9040	0.9229	0.9738	0.9848
TATA_DR	0.9914	0.9969	0.9925	0.9442	0.8729	0.9195
Beta_FA	0.5698	0.5700	0.5700	0.5700	0.5700	0.5700
TAUCA_FA	0.0221	0.0304	0.0337	0.0347	0.0377	0.0440
TATA_FA	0.0169	0.0183	0.0241	0.0376	0.0488	0.0585

(DR: detection rate; FA: false alarm rate.)

TABLE II
PERFORMANCE OF IR FOR MALICIOUS USER DETECTION

N_a	Num-Reg-Weight	Num-Low-Weight
5	3.1	118.8
10	2.7	124.3
15	4.7	126.5
20	4.3	132.6
25	3.8	138.1
30	4.4	142.6

(Num-Low-Weight: the number of low weighted users. Num-Reg-Weight: the number of other users whose opinions really count.)

In this comparison, the key parameters of these schemes are set so that all these schemes could achieve the minimum RRO value of the target item (i.e. $q = 0.2$, $\beta = 0.8$, $\overline{h_c} = 2$ and $\overline{h} = 2.2$). It is important to point out that this comparison favors the IR and the beta function based schemes because their performance is sensitive to their parameter settings. Choosing the parameters for these two schemes is much harder than choosing the parameter for TATA.

Malicious User Detection: To compare the performance of different schemes, we first demonstrate the malicious user detection results. In Table I, the detection rates and false alarm rates for scheme Beta, scheme TAUCA and scheme TATA are demonstrated. We can make several observations. First, scheme Beta performs the worst with low detection rates and high false alarm rates. For example, when there are 20 malicious users, the detection rate of scheme Beta is only 0.67, while the false alarm rate is 0.57, meaning that to detect 13 out of 20 malicious users, 57% of 300 normal users will be misidentified as malicious users. Second, TATA outperforms TAUCA when malicious user number is not large (i.e. $N_a = 5$, 10, 15, 20). And when N_a increases, TATA will have a slightly worse performance than TAUCA. The reason is that TAUCA focuses more on the correlation among users whereas TATA focuses more on individual user's past behavior patterns. When the number of malicious users is small, the collusion among them is not very strong. As a consequence, the scheme investigating individual user behavior patterns may be more suitable. When malicious user number is large, more complicated collusion schemes can be involved, and some malicious users could accumulate high trust values and keep themselves hidden. In this case, due to the behavior similarity among malicious users, the scheme investigating user correlation can be more suitable.

We do not have scheme IR in the Table I, since it does not explicitly detect malicious users. However, it does assign different weights to users according to their rating behavior patterns. In the experiments, we discover that in scheme IR, a large amount of users have very small weights. As a consequence, the calculation of the final reputation score only counts the ratings from very few users. We call a user as a *low weighted user* if his/her weight is lower than 1% of the highest weight. Table II shows the number of low weighted users and the number of other users whose opinions are actually counted. For attacks with different malicious user number, opinions from less than 5 users dominate the reputation score, leaving opinions from more than 120 other users uncounted. Scheme IR may raise concerns since it only counts the opinions of very few users.

RRO Values of the Target Item: In Fig. 10, we compare the RRO of the target item for all different schemes. In Fig. 10, the x axis represents the malicious user number, and y axis represents the RRO values of the target item. We can make three observations. First, the performance of scheme Beta is not stable. For attacks with different number of malicious users, it will achieve very different RRO values for the target item. Second, scheme IR and scheme TAUCA have relatively stable RRO values of the target item for all different attack scenarios. And scheme IR slightly outperforms scheme TAUCA. Third, the proposed scheme TATA has achieved the smallest RRO values of the target item in most of the cases. Especially for attacks with malicious user number as 5, 15 and 20, the RRO values are very close to 0, indicating that the recovered reputation score is very close to the item's real reputation. However, when the malicious user number is 30, the RRO value of TATA is much higher than that of other schemes. This bias is caused by the undetected malicious users. Generally speaking, the proposed scheme TATA has outperformed other schemes in defending most attacks. When the malicious user number goes larger, the performance drops slightly.

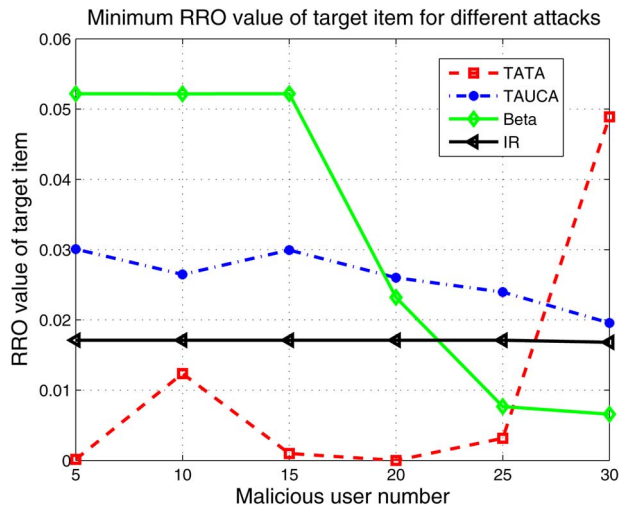


Fig. 10. Recovered reputation offset of the target item comparison for different reputation defense schemes.

RRO Values of All Items: Besides the RRO values of the target item, we need to also consider the RRO values of all items in the reputation system. Although normal items are not attacked by malicious users, some of their normal ratings may be mistakenly determined as dishonest ratings by the reputation defense scheme, leading to reputation distortion. A smaller RRO value indicates a smaller reputation distortion, which is desirable. In Fig. 11, we demonstrate the RRO values of all items when different defense schemes are applied. In Fig. 11, the x axis represents the number of malicious users and the y

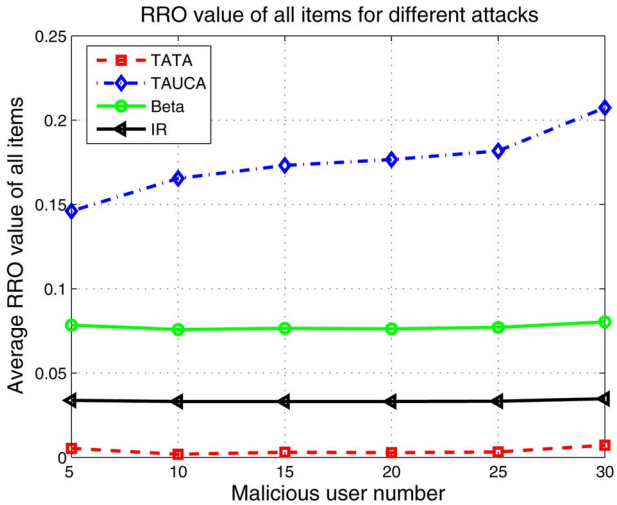


Fig. 11. Recovered reputation offset of all items comparison for different reputation defense schemes.

TABLE III
SUMMARY OF PERFORMANCE COMPARISON

		Sensitive to key parameter?	Detect malicious users?	RRO of target item	RRO of all items
Scheme Beta		Yes	Poor	Moderate	Moderate
Scheme IR		Yes	No	Low	Low
Scheme TAUCA	Small or moderate N_a	No	Good	Moderate	High
	Very large N_a	No	Best	Low	High
Scheme TATA	Small or moderate N_a	No	Best	Very Low	Very Low
	Very large N_a	No	Good	Moderate	Very Low

axis represents the average RRO values of all items. We can observe that (1) scheme Beta, scheme IR and scheme TATA have all yielded stable RRO values of all items, and among them, scheme TATA achieves the smallest RRO values of all items; (2) scheme TAUCA has the largest RRO values of all items. The reason is that in TAUCA, once a user is detected as a malicious user, all of his/her ratings will be removed. For some normal users who are mistakenly determined as malicious users, all of their normal ratings are removed. Therefore, the reputation score of some normal items are affected. This also explains the motivation that in scheme TATA, we evaluate users' trust values on each item and only remove a user's rating if he/she has a low trust value on this rating.

Finally, we summarize the performance comparison results in Table III as follows.

VI. CONCLUSION

In this paper, a comprehensive anomaly detection scheme, TATA, is designed and evaluated for protecting feedback-based online reputation systems. To analyze the time-domain information, a revised-CUSUM detector is developed to detect change intervals. To reduce false alarms, a trust model based on the Dempster–Shafer theory is proposed. Compared with the IR and the Beta model methods, TATA achieves similar RRO values, which represent items' reputation distortion, but much higher detection rate in malicious user detection. For different attacks, the detection rate of TATA is $0.87 \sim 0.99$, whereas IR fails to detect malicious users and Beta model achieves $0.37 \sim 0.72$ detection rate. Compared with our previous scheme TAUCA, which investigates user correlation, TATA achieves a much smaller and more stable RRO values of all items, indicating a small interference on normal items. Furthermore, this study reveals some important insights. When the number of malicious users is not very large, examining individual user's behavior (such as through a well designed trust model in this paper) is a very effective defense approach. When the number of malicious users is very large, investigating user behavior similarity (such as in the TAUCA scheme) becomes a promising method. In the future, one possibility is to jointly consider trust evaluation and user correlation.

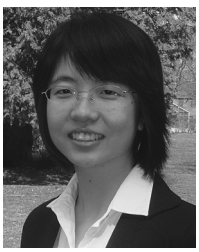
ACKNOWLEDGMENT

The authors would like to thank Q. Feng for designing and administrating the Cyber Competition.

REFERENCES

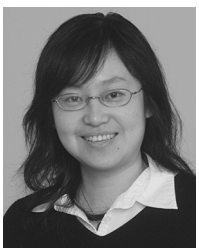
[1] Press Release: Online Consumer-Generated Reviews Have Significant Impact on Offline Purchase Behavior, Nov. 2007 [Online]. Available: http://www.comscore.com/press/release.asp?press=1928
[2] R. Lee and H. Paul, Use of Online Rating Systems Oct. 20, 2004 [Online]. Available: http://www.pewinternet.org/Reports/2004/Use-of-Online-Rating-Systems.aspx
[3] ComScore, Final Pre-Christmas Push Propels U.S. Online Holiday Season Spending Through December 26 to Record $30.8 Billion Dec. 29, 2010 [Online]. Available: http://ir.comscore.com/releasedetail.cfm?ReleaseID=539354
[4] Buy iTunes Ratings and Comments—Increase iTunes Sales and Downloads [Online]. Available: http://www.youtube.com/watch?v=-TWV4XaxCo
[5] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in Bayesian reputation systems," *Icfain J. Manage. Res.*, vol. 4, no. 2, pp. 48–64, Feb. 2005.
[6] P. Laureti, L. Moret, Y.-C. Zhang, and Y.-K. Yu, "Information filtering via iterative refinement," *Europhys. Lett.*, vol. 75, no. 6, pp. 1006–1012, 2006.
[7] Y. Liu and Y. Sun, "Anomaly detection in feedback-based reputation systems through temporal and correlation analysis," in *Proc. 2nd IEEE Int. Conf. Social Computing*, Aug. 2010, pp. 65–72.
[8] Y. Yang, Q. Feng, Y. Sun, and Y. Dai, "Reputation trap: A powerful attack on reputation system of file sharing p2p environment," in *Proc. 4th Int. Conf. Security and Privacy in Communication Networks*, Istanbul, Turkey, Sep. 2008.
[9] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, "A calculus for access control in distributed systems," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 4, pp. 706–734, 1993.
[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," in *Proc. 2006 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2006, pp. 267–278.
[11] J. Weng, C. Miao, and A. Goh, "An entropy-based approach to protecting rating systems from unfair testimonies," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 9, pp. 2502–2511, Sep. 2006.
[12] A. Jøsang and W. Quattrociocchi, "Advanced features in bayesian reputation systems," *TrustBus*, pp. 105–114, 2009.
[13] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proc. 2nd ACM Conf. Electronic Commerce*, 2000, pp. 150–157.
[14] J. Zhang and R. Cohen, "A personalized approach to address unfair ratings in multiagent reputation systems," in *Proc. Fifth Int. Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS) Workshop on Trust in Agent Societies*, 2006, pp. 89–98.

[15] B. Yu and M. Singh, "An evidential model of distributed reputation management," in *Proc. Joint Int. Conf. Autonomous Agents and Multiagent Systems*, 2002, pp. 294–301.
[16] A. Jøsang, "Trust based decision making for electronic transactions," in *Proc. 4th Nordic Workshop on Secure IT Systems*, 1999, p. 99-005.
[17] J. Sabater and C. Sierra, "Social regret, a reputation model based on social relations," *SIGecom Exchanges*, vol. 3, no. 1, pp. 44–56, 2002.
[18] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proc. 12th Int. Conf. World Wide Web*, May 2003, pp. 640–651.
[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proc. 7th Int. Conf. World Wide Web (WWW)*, 1998 [Online]. Available: http://dbpubs.stanford.edu:8090/pub/1998-8
[20] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. Princeton, NJ, USA: Van Nostrand, 1931.
[21] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, Jun. 1954.
[22] T. K. Philips, Monitoring Active Portfolios: The CUSUM Approach.
[23] G. Shafter, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton Univ. Press, 1976.
[24] A. Jøsang, "A logic for uncertain probabilities," *Int. J. Uncertainty, Fuzziness, Knowledge-Based Syst.*, vol. 9, no. 3, pp. 279–311, 2001.
[25] Y. Liu and Y. L. Sun, "Detecting cheating behaviors in cyber competitions by constructing competition social network, poster track," in *IEEE Intl. Workshop Information Forensics and Security (WIFS'11)*, Brazil, Nov. 29–Dec. 2 2011.



Yuhong Liu (S'10–M'12) is currently a postdoctoral research associate at the University of Rhode Island. She received the B.S. degree in information engineering in 2004 and the M.S. degree in signal processing in 2007, both from Beijing University of Posts and Telecommunications, and the Ph.D. degree from the University of Rhode Island (URI) in 2012.

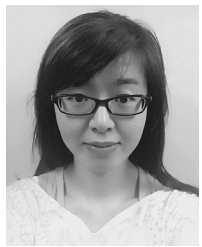
Her primary research interests include trustworthy computing, cyber physical system, and security issues in social network. Her work on detecting dishonest ratings/feedbacks and malicious users in online rating systems received the best paper award at the IEEE International Conference on Social Computing (SocialCom'10, acceptance ratio $= 13\%$).



Yan (Lindsay) Sun (S'00–M'04) received the B.S. degree with the highest honor from Peking University in 1998, and the Ph.D. degree from the University of Maryland in 2004.

She joined the University of Rhode Island in 2004, where she is currently an associate professor in the Department of Electrical, Computer and Biomedical Engineering. Her research interests include trustworthy social computing, trust management in cyber-physical systems, and information assurance.

Dr. Sun is an elected member of the Information Forensics and Security Technical Committee (IFS-TC), in the IEEE Signal Processing Society. She is the recipient of an NSF CAREER Award.



Siyuan Liu received the B.Sc. and M.Sc. degrees in computer science from Peking University of China, both in 2005. She is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Her research interests include software engineering, trust management, multiagent systems, and artificial intelligence.

Alex C. Kot (S'85–M'89–SM'98–F'06) has been with the Nanyang Technological University, Singapore since 1991. He headed the Division of Information Engineering at the School of Electrical and Electronic Engineering for eight years and served as Associate Chair/Research and Vice Dean Research for the School of Electrical and Electronic Engineering. He is currently Professor and Associate Dean for College of Engineering and Director of Rapid-Rich Object SEarch (ROSE) Laboratory. He has published extensively in the areas of signal processing for communication, biometrics, data-hiding, image forensics, information security, and image object retrieval and recognition.

Dr. Kot served as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SIGNAL PROCESSING LETTERS, *IEEE Signal Processing Magazine*, the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BREIFS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He also served as Guest Editor for the Special Issues for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and JASP. He is currently Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE TRANSACTIONS ON IMAGE PROCESSING. He is also Editor for the *EURASIP Journal of Advanced Signal Processing*. He was with the IEEE SPS Image and Video Multidimensional Signal Processing TC and he is now in the IEEE CAS Visual Signal Processing and Communication and IEEE SPS Information Forensic and Security technical committees. He has served the IEEE SP Society in various capacities such as the General Cochair for the 2004 IEEE International Conference on Image Processing (ICIP) and Chair of the worldwide SPS Chapter Chairs and the Distinguished Lecturer program. He was a member in the IEEE Fellow Evaluation Committee and is the Vice-President for the IEEE Signal Processing Society. He received the Best Teacher of the Year Award and is a coauthor for several Best Paper Awards including ICPR, IEEE WIFS, and IWDW. He was the IEEE Distinguished Lecturer in 2005 and 2006 and is a Fellow of IES, and a Fellow of Academy of Engineering, Singapore.