Rating Fraud Detection---Towards Designing a Trustworthy Reputation Systems

Completed Research Paper

Yuanfeng Cai
Zicklin School of Business
City University of New York--Baruch College
New York, NY
Yuanfeng. Cai@baruch. cuny.edu

Dan Zhu
College of Business
Iowa State University
Ames, IA
dzhu@iastate.edu

Abstract

Reputation systems could help consumers avoid transaction risk by providing historical consumers' feedback. But, traditional reputation systems are vulnerable to the rating manipulation. It will undermine the trustworthiness of the reputation systems and users' satisfaction will be lost. To address the issue, this study uses the real-world rating data from two travel website: Tripadvisor.com and Expedia.com and one e-commerce website Amazon.com to empirically exploit the features of fraudulent raters. Based on those features, it proposes the new method for fraudulent rater detection. First, it examines the received rating series of each entity and filter out the entity which is under attack (termed as target entity). Second, the clustering based method is applied to discriminate fraudulent raters. Experimental studies have shown that the proposed method is effective in detecting the fraudulent raters accurately while keeping the majority of the normal users in the systems in various attack environment settings.

Keywords: Reputation systems, rating fraud, time series

Introduction

Over the past decades, the Internet has come to play an important role in many parts of our daily lives. For example, Amazon had 240 million users by May 2014 (Smith, 2015) and 2 billion products are purchased in one year (Smith, 2015). By March 2015, 300 million active users shared information on Twitter (Welch and Popper, 2015). With the advances in information technology, the cyber world has transformed itself into the dominant platform for people to express themselves and connect with others all over the world. Unlike the "real-world", interaction in the cyber world is characterized by anonymity. It can occur among people who do not know each other's real identity. In this way, the Internet has broken geographical limitations and provided a vast collection of information sources.

Despite the convenience resulting from social media for information exchange, interaction with strangers may involve risks. For instance, false rumors spread across the Internet can lead financial investors to make bad decisions. Purchasing products from unreliable sellers may result in heavy losses. Hence, people should be cautious in interacting with strangers so as to take advantage of opportunities while protecting themselves. In the real-world, people seek to deal with entities (e.g. people, items, organizations and etc.) that have positive reputations. Similarly, reputation can be a critical precautionary measure for people to regulate their interactions with strangers in the cyber world. Reputation is a distribution of opinions, estimations, or evaluations about an entity in an interest group (Bromley, 2001). An interest group indicates that people within this group have some relationship or concern with the entity (Bromley, 2001). In the cyber world, without knowing the real identity of the interest group, the opinion about the entity is difficult to collect by direct inquiry. Therefore, reputation systems have been designed to provide people with the reliability of strangers before making contact with them.

Reputation systems can collect, aggregate, and distribute feedbacks about entities' past behavior (Resnick et al. 2000). The aggregated feedbacks are called the reputation score. In most commercial systems, feedback could be contributed by users in the form of numerical ratings. In this paper, we term the user providing ratings as rater. For example, raters in Amazon can rate the entity on a scale from 1-5 stars, with the higher value indicating the greater satisfaction. The rating score is calculated based on the rating from every user, and will be updated with the arrival of the new rating. The calculated score is disseminated to all customers as their decision reference. Previous research has already shown that reputation systems are effective means of decreasing transaction risks, facilitating buyer satisfaction, and generating premiums for e-retailers (Ba and Pavlou 2002; Houser and Wooders 2006).

In spite of their effectiveness, reputation systems are vulnerable to rater manipulation. Raters may inject biased ratings on entities for their own benefit. We term this behavior where the rater provides unfair ratings as rating fraud, and such users are fraudulent raters. There is abundant evidence for the existence of fraudulent raters. In August 2013, Samsung admitted to hiring people to inject negative ratings into products of its competitor HTC (Chang 2013). Even worse, there are rating management organizations that provide professional services for online rating manipulation. For instance, nineteen review management companies were caught and fined $350,000 for injecting fake consumer ratings into various sites including Yelp, Google Local, and Yahoo Local in early 2014(Sved 2014). When the rating score is biased by such organized and profit-driven activities, the trustworthiness of the reputation systems will be undermined. Therefore, it is necessary to develop mechanisms to detect rating fraud.

Several methods have been proposed for rating fraud detection. One research stream relies on examining the deviation of individual rating values from the majority or the past rating values (Fei et al. 2013; Jindal and Liu 2008; Lim et al. 2010; Liu. et al. 2011; Mukherjee et al. 2013). However, misleading results will be generated when the majority or the early raters are fraudulent (i.e Sybil Attack (Irissappane et al. 2012)). Another stream of research first identifies a reliable rater and utilizes his or her ratings to filter out the suspicious raters by noting dissimilarities (Dellarocas 2000; Teacy et al. 2006), while it faces difficulty when fraudulent raters strategically behave like the honest raters (i.e. Camouflage Attack (Irissappane et al. 2012)). In summary, previous methods have primarily concentrated on identifying the fraudulent rater purely by examining the rating values, which are vulnerable in Sybil attack and Camouflage Attack environment.

We propose an improved approach to address the limitations. Different from previous methods, we look into not only the rating values, but also their order of occurrence (i.e. the rating time). Hence, we consider features from both value and temporal dimensions in rating fraud detection. For each entity or each rater, we examine its associated ratings from the rating series perspective to discover the features of the target entity or the fraudulent rater. In this study, we introduce the features of the target entity and the fraudulent rater by providing detailed analysis and empirical evidence based on using two real-world rating datasets: one hotel dataset from Expedia.com and TripAdvisor.com, and another one from Amazon.com. Then, we propose a two-step procedure for fraudulent rater detection. In this newly proposed procedure, we first examine the suspicious entity by applying Ljung-Box test its rating series. Subsequently, we retrieve a list of users who have rated the suspicious entity(s). For users on this short list, we adopt clustering-based methods to identify raters with the similar rating pattern, and then discriminate the cluster with fraudulent raters based on one proposed group indicator called as *GBurst*. Here, we can detect the group of fraudulent raters, as most of the influential rating fraud is collaborative and organized. While a single rater can also perform an independent attack, the collaborative rating fraud, which occurs when a seller or the rating management organization can control multiple users or user IDs to inject unfair ratings strategically on the target entity(s), brings more significant challenges to the accuracy of the reputation systems and is the focus of the present study. Finally, we evaluate the effectiveness of the proposed method based on a real-world cyber competition data. The broad consideration of rating fraud features could facilitate the robustness of the proposed method various attack environment.

In the next section we provide a brief review of research on rating fraud model, followed by an overview of the literature on rating fraud detection. In the third section, we introduce the proposed method for rating fraud detection. The fourth section presents the experimental studies. Finally, the last section concludes the papers and discusses the future research directions.

Literature Review

Reputation systems

Reputation systems could be either user-driven or content-driven. User-driven reputation systems are popularly applied to electronic-commerce websites, such as the eBay and Amazon, where users could perform transactions and provide feedbacks to the sellers. User-driven reputation systems calculate the reputation score for the target entity, e.g. the seller, on the basis of the raters' ratings. User-driven reputation systems adopt a variety of algorithms to calculate the reputation score: it can be calculated as the differences between all positive scores and negative scores (e.g. eBay) (Resnick and Zeckhauser 2002); or as the mean of all ratings (e.g. Amazon) (Schneide et al. 2000), or as the weighted average of all the ratings where the weights include the rating age, the rater helpfulness, and etc. (Liu et al. 2013). In a more complex method, for instance, Beta Reputation Systems (BRS) utilizes the previous positive and negative ratings as the parameters to formulate the beta probability density functions. Given by the previous rating score and the new rating, the system can update the reputation score timely (Jøsang et al. 2007). User-driven reputation systems are advantageous in that they collect the feedbacks from all users willing to share their experiences. The potential buyers can obtain a relative comprehensive view of the target entity. However, as user-driven reputation systems rely on the rater' input, they are vulnerable to rating fraud and the reliability of the systems is at risk. We will introduce rating fraud models in the next Subsection.

Content-driven reputation systems could be applied to wiki-based websites, such as the Wikipedia, where people could contribute contents as well as modify others' contributions freely. To prevent spam, the reputation of the contributor is of interest. Different from user-driven reputation systems, content-driven reputation systems do not rely on users' rating to evaluate the contributor's reputation, instead, they derive the reputation score based on contributors' content evolution (Adler and Alfaro, 2007). In general, contributors will gain high reputation if their contents are long-lasting and are not reverted (Adler and Alfaro, 2007). Although content-driven reputation systems become unreliable if the contents are modified maliciously (Chatterjee et al. 2008), such behavior is different from rating fraud as there is no rater or rating activity involved. Herein, this study focuses on user-driven reputation systems and content-driven reputation systems are beyond the scope.

Rating Fraud Model

Intuitively, the goal of rating fraud could be either to increase their own reputation score, or to decrease the competitor's reputation score. This thus leads to two types of rating fraud: Ballot Stuffing, the unfairly high ratings are injected to the target entity, and Bad Mouthing, the unfairly low ratings are injected to the target entity (Dellarocas 2000). To realize the goal, various types of rating fraud models have been discovered, and the typical list of rating fraud model is briefly introduced as below (Irissappane et al. 2012). Each one of them could be applied for Ballot Stuffing and Bad Mouthing, with the only difference in the rating values of the target entity.

- *Consistent Attack*: the fraudulent rater consistently provides the unfairly high (low) rating to the target entity(s) with low(high) quality during a Ballot Stuffing (Bad Mouthing) attack;
- *Camouflage Attack:* the fraudulent rater launches the rating fraud strategically. Besides injecting dishonest ratings, either unfairly high or unfairly low, to the target entity(s), the fraudulent rater also submits fair ratings to certain non-target entity(s) to camouflage himself or herself as the honest rater. Intuitively, camouflage attack makes it more difficult to differentiate fraudulent raters as they behave similarly to honest raters.
- *Whitewashing Attack:* in most of the e-commerce websites, there is no limitation that one person can only register one account. Hereby, a fraudulent rater, who has provided the target entity(s) with either unfairly high or unfairly low ratings, could easily whitewash the history by creating a new account and behave like an honest rater.
- *Sybil Attack:* different from the previous three types, Sybil Attack does not specify the fraudulent rater's behavior. It describes the overall rating fraud environment. When the number of fraudulent raters is larger than that of honest raters, this scenario is termed as Sybil Attack (Douceur 2002). As Sybil Attack does not define the individual fraudulent rater behavior, it could combine with the first three types to constitute three Sybil-based Attack models, which are *Sybil*

Consistent Attack, Sybil Camouflage Attack and *Sybil Whitewashing Attack*. Each one indicates that fraudulent raters are more than honest raters and perform Consistent Attack, Camouflage Attack and Whitewashing Attack respectively.

Rating Fraud Defense Mechanisms

To deal with rating fraud, previous studies have already proposed various defense mechanisms, including preventative and detective mechanisms. Preventative mechanisms can either discourage raters from dishonest or utilize incentive for raters to be honest. For instance, if the account is bound with one unique IP address, it will increase the cost to register many accounts (Douceur 2002). Yu et al. (2006) also proposed the protocol SybilGuard, which increases the difficulty of controlling multiple rater accounts to perform attack. The preventative approaches are designed to limit the occurrence of fraudulent activity, not to capture. And their purposes are hardly to realize if raters spoof their IP address or they can realize more profits by injecting attacks.

The purpose of defensive solution is to detect the existed fraudulent raters. Various methods have already been proposed. Whitby et al. (2004) develop Beta Reputation Systems to deal with rating fraud. If the overall entity rating falls in the rejection areas of the beta distribution of the target user's ratings, this user is considered dishonest according to the majority rule. Jindal and Liu (2008) adopt logistic regressions based on rater and rating features including textual features, the number of feedbacks and rating deviation. Lim et al. (2010) propose scoring method to identify fraudulent raters whose rating deviate from others. Wang et al. (2011) use a graphical method which considers the relationship among raters, ratings and entities. Liu et al. (2014) develops the model iCLUB which utilizes both local and global rating deviation to identify dishonest users. Mukherjee et al. (2013) design author spamicities model to detect fraud using features including extreme rating and rating deviation. Liu et al. (2013) propose a fuzzy logic which combines user's rating time, rating value similarity and rating quantity to against unfair ratings. Although these solutions improve the robustness of reputation systems against rating fraud, they rely on comparing the individual rating with the overall rating trend from other raters to identify the fraudulent rater. Thus, they are vulnerable to Sybil Attack.

Instead of relying on the majority rule, Liu et al. (2011) assumes that the entry of a large amount of malicious ratings would lead to the sudden change of the overall rating of the entity. Their proposed method detects fraudulent rater by locating the rating change period. Although experiments have shown that this method increases the detection accuracy significantly, this method is vulnerable when the malicious users are the early raters. Dellarocas (2000) utilizes the pre-known honest rater's rating to filter out the suspicious raters based on the rating dissimilarities. Another method TRAVOS is proposed by Teacy et al. (2008). It evaluates the trustworthiness of the rater by comparing its rating activity with that from other pre-labeled honest raters. These methods require the prior knowledge of honest rater and assume that raters have a constant behavior, which is vulnerable to Camouflage Attack.

Rating Fraud Detection

In this section, we discuss the proposed method in details. This study addresses the collaborative rating fraud. It can be carried out by multiple users or by one user controlling multiple user IDs. We refer each of these users as a fraudulent rater. Fraudulent raters have their *target entity(s)* in which they inject unfairly high (Ballot Stuffing) or low (Bad mouthing) ratings. They can also provide ratings beyond the target entity (or entities). Due to the collaborative rating's fraudulent nature, multiple malicious users could be associated with each target entity. Thus, our defense mechanism includes two steps: 1) identify the potential target entity(s) based on the entity features; 2) Retrieve the associated users and filter out the fraudulent raters by user features. Thus, we should first examine the important characteristics of target entity and fraudulent raters respectively that are used in our fraud defense methodology. Different from previous methods, we look into features not only from the rating values dimensions, but also from temporal dimensions. We examine the associated rating series for entities and raters to discover the feature of the target entity and that of the fraudulent rater. Next we will introduce each feature, which is identified with the detailed analyses and empirical evidence.

Target Entity Detection

For each entity e, we create a rating series for all its ratings over the time and denote it as r_t^e.

$$r_t^e = \{ r^e{}_{t_1}, r^e{}_{t_2}, \ldots r^e{}_{t_k} \};$$

Where t_k is the order for this rating instead of its accrual time. For example, $r^e{}_{t_1}$ indicates that it is the first rating for the entity e. We use the order (relative time) instead of the accrual time since our concern is the correlation among ratings rather than the change of entity overall rating.

We use the real-world dataset to empirically examine the feature of the rating series of the target entity. Despite the widely existence of the rating fraud, the target entity is seldom explicitly labeled in the current reputation systems. Hence, it is not easy to analyze its characteristic directly. In this study, we examine the feature of the entity using one hotel rating dataset. Specifically, we evaluate the rating series for identical hotel entities in two different websites: Expedia.com and TripAdvisor.com. Both of them are the famous travel websites which provide users' feedbacks towards hotels all over the world. Although neither of the two websites labeled the target entity, there is an intrinsic difference in their rating mechanism. TripAdvisor.com is an open space that anyone could post a rating, but raters at Expedia.com must be the real customers, i.e. who have checked in for at least one night. Thus, ratings at Expedia.com are claimed as "verified rating". Therefore, TripAdvisor has a higher potential than Expedia to be exposed to rating fraud. Accordingly, we can exploit the characteristics of the rating fraud by comparing the rating series of the identical hotel at both Expedia.com and TripAdvisor.com. We have gathered one panel dataset from both Expedia.com and TripAdvisor.com. We collected all the ratings series for 2116 hotels which has at least 10 ratings in both websites. Both websites use the scale from 1-5 stars. For every hotel, we retrieved all its raters' rating value and the associated posting time. The final dataset consist of 2116 hotels with 300,521 ratings from Expedia.com and 438,703 ratings from TripAdvisor.com. Table 1 shows the descriptive statistics of the rating distribution in the dataset. As expected, compared to Expedia, TripAdvisor has a larger number of ratings due to the loose rating submission restriction. Also, for the same hotel, the average rating at TripAdvisor.com is lower than that at Expedia.com. Moreover, we calculate the proportion of each possible rating value for each hotel at two websites. Hotels at Expedia.com have a significantly higher (lower) proportion of high (low) rating values compared to those at TripAdvisor.com. For example, on average, 35.10% of rating for each hotel at Expedia.com is 5-star. It is consistent with the result that the average rating of Expedia.com is higher. Since the average rating at Expedia.com largely reflects users' real opinions, rating fraud may exist at TripAdvisor.com. Thus, hotels at TripAdvisor.com are more likely to be the target entity. We could exploit the characteristics of the target entity by comparing their rating series difference for the same hotel in two websites.

Table 1. Summary Statistics of Rating Distribution at Two Websites (N=2116)

	Expedia	TripAdvisor	Difference
Average Number of Rating	142.02	207.33	-65.30(6.01)*
Minimum Number of Rating	10	10	N/A
Maximum Number of Rating	999	1140	N/A
Mean of Rating	3.89	3.64	0.25(0.01)*
Standard Deviation of Rating	0.56	0.68	N/A
1 Star	4.70%	10.49%	-5.79%(0.37%)*
2 Star	7.59%	8.92%	-1.33%(0.23%)*
3 Star	16.53%	18.86%	-2.33%(0.33%)
4 Star	36.08%	31.73%	4.35%(0.40%)*
5 Star	35.10%	29.99%	5.11%(0.61%)*

*p<0.001 Notes: The sample sizes are 300,521 (ratings from Expedia.com) and 438,703 (ratings from TripAdvisor.com).

Raters have various backgrounds. If there is no fraud, ratings from different raters should be independent with each other (Hu et. al 2012; Xie et al. 2012). Accordingly, the rating values for a non-target entity should be mutually independent and identically distributed with respect to time. Herein, $\{r_t^e - \mu\}$ is white noise, where r_t is the rating at the timestamp t and μ is the mean of all ratings. In the collaborative rating fraud, fraudulent raters are organized so that their ratings are not independent with each other. Hence, for target entity, its $\{r_t^e - \mu\}$ is not able to be modeled as a white noise with the existence of the fraudulent raters. Although in certain cases, a self-selection process may exist so that ratings from normal users may not be random (Li and Hitt 2008). Hu et al. (2010) analyze datasets from Amazon and Barnes & Noble.

The result shows that regardless of self-selection, rating fraud must exist in the entity if there is dependency in its ratings.

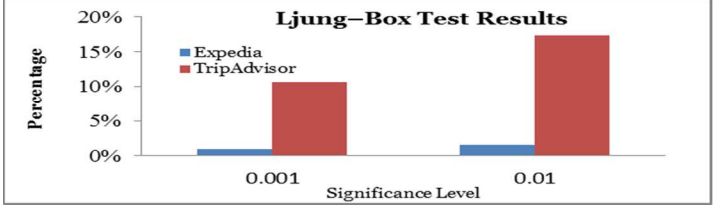

Figure 1. Ljung–Box Test Results at Two Websites

To detect the existence of dependency in the rating series, we adopt Ljung–Box test (Ljung and Box, 1978). The null hypothesis of Ljung–Box test is that the data series are independently distributed. For each hotel, we created two r_t^e for its rating series at Expedia.com and TripAdvisor.com respectively. To differentiate the two time series, we denote them as r_t^{eExpe} and r_t^{eTrip} respectively, where $eExpe \in [1,2116]$ and $eTrip \in [1,2116]$. If the p-value of the test for r_t^{eExpe} or r_t^{eTrip} is below the significance level, entity $eExpe$ or $eTrip$ will be labeled as the target entity. Figure 1 displays the Ljung–Box Test result for all r_t^{eExpe} and r_t^{eTrip}. The x-axis is the significance level and the y-axis is the percentage of hotels which are labeled as target entities. Figure 1 shows that there are significantly more hotels labeled as target entities at TripAdvisor.com, e.g. 11% at the 0.001 significance level and 17% at the 0.01 significance level, while only 0.9% at the 0.001 significance level and 1.5% at the 0.01 significance level at Expedia.com. The result is consistent with the expectation that more entities at TripAdvisor are under attack, which indicates that Ljung-Box Test could be a good indicator for target entity detection.

Fraudulent Rater Rating Series

Similar to the issue in target entity detection, no real-world reputation systems have labeled the dishonest rater. And since Expedia.com does not track the historical ratings for each rater, the hotel dataset used in previous subsection could not be applied. Thus, we need to find another way to empirically examine the feature of the fraudulent rater. To save the attack cost, fraudulent raters are usually not the real buyers. That is to say, if most of the submitted ratings are from a real purchase, the rater is highly likely to be an honest one. On the contrary, if none of the ratings is from "verified purchase", the rater may be malicious. Currently, several e-commerce websites, such as Amazon.com, label the rating to indicate whether it is from a "verified purchase". Herein, we collect one user-entity rating dataset from Amazon.com to analyze the feature of fraudulent rater. We have gathered all the historical rating activity for 1523 raters who have ever rated the product of digit camera. For each rating activity, we record the rating value, the rating date, the rated entity ID, and a binary indicator of whether it is from "verified purchase". For each rater, we calculate the ratio of "verified purchase" rating. We admit that the rating not from "verified purchase" is not necessary fraudulent, since it may lack the transaction record. However, the larger ratio of the "verified purchase" rating can indicate a higher probability of honest rater. Otherwise, it would be too costly to perform rating manipulation. Here, we label raters as honest if their ratio of "verified purchase" rating is greater than 75%, and those as fraudulent if none of their ratings is from the verified purchase. Finally, the dataset contains 73,522 ratings from 359 honest raters and 52,209 ratings from 305 fraudulent raters. Table 2 lists the summary statistics of the dataset. We could note that users classified as fraudulent raters indeed have a significant different rating behavior that those labeled as honest raters. Honest raters have the significantly higher average rating and a higher proportion of larger rating values.

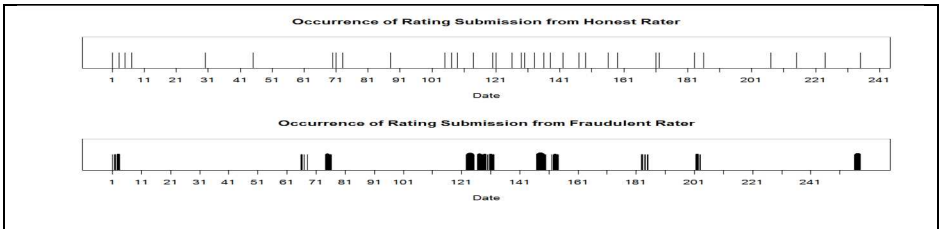

Figure 2. Rating Time Series from Different Type of Raters

Fraudulent raters have unique temporal features in their own rating series. As the deadline is usually given, profit driven raters usually accomplish their tasks within a short period intensively, i.e. right before the deadline (Parker 2011). Hereby, the rating submission may burst within a very short time interval, and the overall time interval between every two consecutive ratings for a malicious user should be very small. However, time interval between ratings for honest users usually is larger since it is affected by their time interval of online shopping and the time needs for making rating decisions. This phenomenon has been observed in network intrusion and mobile Apps ranking fraud (Soldo 2011; Sorrel 2009; Zhu et al. 2013).Figure 2 illustrates an example of the rating time series from one honest rater and one fraudulent rater in the dataset. We could find that compared to the honest rater, the fraudulent rater's time series contains more conglomeration. For fraudulent rater, most ratings are injected in quick succession and its time interval has more variability. Thus, the fraudulent rater is expected to have larger variability of time interval as well as the smaller average time interval. Accordingly, we term such a feature of user's rating series as the *degree of burst (*$Burst_u$*)* and define its measurement as the ratio of the standard deviation of time interval (σ) to the mean of minimum rating interval (MRI). The value of $Burst_u$ will be higher for a rating series with a higher degree of burst. The calculation formula is shown (1).

$$Burst_u = \frac{\sigma_u}{MRI_u} \quad (1)$$

Where

$$\sigma_u = \sqrt{\frac{\sum_{n=1}^{n_u-1}((t_{n+1}^u - t_n^u) - \mu_u)^2}{n_u - 1}} \quad (1.a)$$

$$\mu_u = \frac{\sum_{n=1}^{n_u-1}(t_{n+1}^u - t_n^u)}{n_u - 1} \quad (1.b)$$

$$MRI_u = \frac{\sum_{2}^{n_u-1} \min[(t_n^u - t_{n-1}^u), (t_{n+1}^u - t_n^u)]}{(n_u - 2)} \quad (1.c)$$

where n_u is the total number of entities that the user u has rated and t_n^u is the actual day of the nth rating the user u has given for all possible entities. Here, we use MRI_u other than μ_u to represent the average time interval since the later one is not a good indicator of burst. For any rating time series, as long as the total number of rating and the difference between the first and the last rating timestamp are the same, $\mu_u$will be always the same value regardless of occurrence time of other ratings. Based on (1), we calculate the values of $Burst_u$ for each user in the Amazon dataset and Table 2 shows the results. The users with a higher proportion of verified purchases have a significantly lower degree of burst than those with no records of verified purchases. Hence, $Burst_u$ could be a potential indicator to detect the fraudulent rater.

Table 2. Degree of Burst of Different Types of Raters

	Fraudulent Rater	Honest Rater	Difference
Mean of Rating	3.90	4.12	-0.23(0.04)*
Standard Deviation of Rating	1.04	1.04	N/A
Average Number of Rated Item	169.55	204.22	N/A
1 Star	7.27%	6.60%	0.67%(0.72%)
2 Star	8.19%	5.49%	2.71%(0.53%)*
3 Star	14.72%	11.32%	3.40%(0.73%)*
4 Star	27.04%	22.08%	4.95%(1.1%)*
5 Star	42.78%	54.51%	-11.72%*(1.7%)*
Burst	103.08	22.71	80.37(26.4)*

*p<0.001. Notes: The sample sizes are 359 honest raters and 305 fraudulent rater

Besides the degree of burst, fraudulent raters have another feature in collaborative rating fraud scenario. Since fraudulent raters have their own specified objectives, they behave differently from the genuine users. Genuine users provide ratings for the entity based on their preference. Individual difference makes it difficult for a group of users to rate the exactly same entities. However, as the activities of fraudulent raters are usually controlled and panned, particularly those hired by the organizations, Entities rated by colluded fraudulent raters are different from those rated by normal users. For fraudulent raters, there are two types of entities to rate: target entity, and non-target ones (if under camouflage attack). For target entities, the colluded fraudulent raters rate the exactly same ones since they are predetermined and are usually not interested by genuine users. For non-target entities s, fraudulent raters select entities randomly in order to camouflage themselves. Despite the randomness, the non-target entities selected by

colluded fraudulent rater IDs would still have a high similarity since several rater IDs may be controlled by the same user. The user with several fraudulent rater IDs may just pick up the same set of entities for convenience. Therefore, fraudulent raters are expected to have a similar rating pattern. This feature can be utilized to cluster fraudulent raters.

Fraudulent Rater Detection Algorithm

Based on the features of the target entity and fraudulent rater, our algorithm is designed to consist of two phases. First, we select the suspicious entity based on the entity's rating series and retrieve its associated raters. Second, we examine the rating series of every rater selected in the first step and discriminate a set of fraudulent raters through clustering with the consideration of the group degree of burst.

Phase 1. Select a set of P target entities and their corresponding Q users.
Input:
$R = r^{ue}$: (n*m) user-entity rating matrix, where $u \in [1, n], e \in [1, m]$;
$T = t^{ue}$: (n*m) user-entity rating time (day) matrix, where $u \in [1, n], e \in [1, m]$;
Procedures:
Step 1. For each entity e, construct its non-void rating vector $\boldsymbol{r}^e = \{r^e{}_1, r^e{}_2, \dots r^e{}_k\}$and its corresponding rating time vector $t^e = \{t^e{}_1, t^e{}_2, \dots t^e{}_k\}$;
Step 2. Sort $\boldsymbol{r}^e$based on the values in $\boldsymbol{t}^e$ in ascending order such that $\boldsymbol{t}^e = \{t^e{}_{t_1}, \dots t^e{}_{t_k}\}$, $where t^e{}_1 \leq \cdots \leq t^e{}_k$. Thus, $\boldsymbol{r}^e = \{r^e{}_{t_1}, \dots r^e{}_{t_k}\}$ where t_k is the order for the rating $r^e{}_{t_k}$ in the vector $\boldsymbol{r}^e$;
Step 3. Apply Ljung–Box test to $\boldsymbol{r}^e$. If the p-value is below the significance level, $Flag_e$=1.
Step 4. Repeat Step 1~3 for all entities. Find the P entities with $Flag_e$=1;
Step 5. Find Q users who have ever rated at least one of the P entities.
Output:
A binary vector $\boldsymbol{Flag} = \{Flag_1, \dots Flag_{\mathrm{e}} \dots Flag_{\mathrm{m}}\}$ where $e \in [1, m]$;
$\bar{T} = t^{ue}$: (q*m) user-entity rating time (day) matrix consisting of the selected users in Step 5, where $u \in [1, q]$;
$\bar{R} = r^{ue}$: (q*m) user-entity rating matrix consisting of the selected users in Step 5, where $u \in [1, q]$;

Phase 1 is based on the feature of target entity. In Step 1 we retrieve rating value and rating time for each entity from the user-entity rating matrix. As the input is the user-entity rating matrix, the original rating vector is ordered by the user ID. Herein, in Step 2, we sort ratings in ascending order of time to construct the rating time series of each entity. In Step 3, the entity with the significant auto-correlated rating series is identified as the suspicious entity. A significance level need to be selected in this step. The higher significance level is used, the higher likely of committing type 1 error, i.e. higher false positive rate in target entity detection. In contrast, if the significance level used is too low, a large number of target entities may not be detected. In addition, as the null hypothesis of Ljung-Box test is no autocorrelation up to lag h, we need to make sure that none of the h p-values are above than the significance level. In this study, as suggested by Tsay (2010), we choose the number of lag $h = \ln(n)$, where n is the length of the rating series. If $\ln(n)$ is not integer, we will round it up. Step 4 and 5 find all the suspicious entities and filter out their associated users. For users in this shortlist, we construct a user-entity rating matrix for all the entities. If one entity is not rated by one user, the cell will be empty. Similarly, a user-entity rating time matrix is also constructed between users in the shortlist and all the entities.

Phase 2. Cluster a set of fraudulent raters based on their rating pattern.
Input:
$\bar{T} = (t^{ue})$ and $\bar{R} = (r^{ue})$: the output from the phase 1;
Procedures:
Step 1. Construct an (q*q) user-user distance matrix $\boldsymbol{D} = (d_{ui})$, based on the user-entity matrix $\bar{R}$.
Step 2. Cluster users into J groups using $\boldsymbol{D}$.
Step 3. For each cluster C, C=1,…, J, calculate its $GBURST_C$. See below for the detailed calculation.
Step 4. Select the cluster with the highest $GBURST_C$.
Output:
The group of users selected in Step 4. They are identified as fraudulent raters and are suggested to be removed from the user list.

Step 1 and 2 use the feature that fraudulent raters have a similar rating patterns towards the entity. In Step 1, based on the user-entity rating matrix, we calculate the distance between every two users by using "1-Tanimoto coefficient". Tanimoto coefficient is a general form of Jacarrd coefficient, which has shown a clear advantage over other similarity measures in the case of extremely asymmetric distributed or sparse data vectors, such as in the rating dataset (Mild and Reutterer 2003). For user u and user i, $d_{ui} = 1 - \frac{\widehat{r^u} * \widehat{r^i}}{(||\widehat{r^u}||^2 + ||\widehat{r^i}||^2 - \widehat{r^u} * \widehat{r^i})}$, where $\widehat{r^u}$ and $\widehat{r^i}$ denoting the rating vector for their common rated entities respectively. Then users are clustered using $\boldsymbol{D}$ in Step 2 using hierarchical clustering method. The distance between clusters is calculated by Ward's method. Step 3 calculates the degree of burst in its users' ratings for each cluster. Here, instead of using $BURST_u$ directly, we use the degree of burst in the entire group C as $GBURST_C$. To calculate $GBURST_C$ we first combine all the rating vectors from users in the cluster C and sort the rating values in the ascending order of their corresponding actual rating time. The rationale behind merging all the users' rating in one cluster is that the clusters we aim to find are those perform collaborative rating fraud. Thus, one person can control multiple user accounts. The real person may inject ratings in a short period of time using multiple accounts and stop for a while before perform the next attack. Hence, it is $GBURST_C$, not $BURST_u$ of each fraudulent account, which could display a high degree of burst. We use $GBURST_C$ as it represents more accurately the feature of the degree of burst. Adapted from formula (1), the formula of $GBURST_C$ is shown in (2) as below.

$$GBURST_C = \frac{\sigma_C}{MRI_C} \qquad C \in [1, J] \qquad (2)$$

Where $$\sigma_C = \sqrt{\frac{\sum_{n=1}^{n_C-1}((t_{n+1}^C - t_n^C) - \mu_C)^2}{n_C - 1}} \qquad (2.a)$$

$$\mu_C = \frac{\sum_{n=1}^{n_C-1}(t_{n+1}^C - t_n^C)}{n_C - 1} \qquad (2.b)$$

$$MRI_C = \frac{\sum_{2}^{n_C-1} \min[(t_n^C - t_{n-1}^C), (t_{n+1}^C - t_n^C)]}{(n_C - 2)} \qquad (2.c)$$

where n_C is the total number of entities that all the users in cluster C has rated, and t_n^C is the actual day of the n^{th} rating after all the users' rating in cluster C are combined and sorted in ascending of time. Finally, clusters with the highest calculated $GBURST_C$ in Step 3 are selected. The users in those clusters are labeled as fraudulent raters.

Experimental Results

In this section, we evaluate the performance of our proposed method and present the experimental results. To evaluate the fraudulent rater detection performance accurately, the testing dataset is expected to have pre-labeled fraudulent rater and honest rater, which is difficulty to get from today's e-commerce sites. Previous studies usually conduct simulation to obtain the pre-labeled raters. But the simulation could not fully represent the real-world environment, so that the accuracy of performance evaluation may be affected accordingly.

In this study, we use a cyber-competition data to solve this dilemma, which includes both normal rating data and attack rating data (Liu et al. 2011). In both normal and attack data, each piece of rating contains the entity ID, the user ID, the rating time and the rating value. The normal data are collected from a famous e-commerce site with rating values in the numeral scale 1 to 5. The normal dataset contains 5688 user-entity rating records collected over 150 consecutive days from 300 normal users for 300 products (entities), denoted as (u_1,.....u_{300}) and (e_1,.....e_{300}) respectively. The 300 normal users (u_1,.....u_{300}) are considered as honest users in this study. The attack data are obtained from an attack competition designed for this e-ecommerce site. The goal of the competition is to encourage participants to downgrade the rating of the target entity (e_1 in this competition). One participant can control at least 20 user IDs to launch an attack, and all his/her submitted ratings are recorded as in one attack file. Each user ID in one attack file is considered as one fraudulent rater. There are a total of 13028 attack files in the attack dataset.

As all the fraudulent raters in one attack file are controlled by one participant, it is collaborative rating fraud so that our proposed method can be applied.

For every attack file, we measure its attack effectiveness using the *rating shift*, denoted as $\Delta = \bar{r}^{e} - \bar{r}^{e\prime}$. The $\bar{r}^{e}$ and $\bar{r}^{e\prime}$ represent the average rating for the target entity before and after the attack, respectively. Since the goal of this competition is to downgrade the target entity, a larger value of Δ indicates a stronger attack and a smaller value indicates that the participant's behavior is quite similar to the normal user. For all the attack files, we categorize them based on their own Δ values into three attack level groups: *weak attack group* with Δ value between [0.1, 0.25), *moderate attack group* with Δ value between [0.25, 0.4), and *strong attack group* with Δ value between [0.4, 0.55). There are 1543, 4254 and 7231 files in each group respectively. It is consistent with the purpose of the competition that the majority of participants are strong attackers. The detection performance will be evaluated in each group respectively.

Although the attack task for all participants is the same, the original attack dataset shows two types of fraud models: 8169 files with consistent attack, which only inject unfair ratings to the target entity, and 4859 files with camouflage attack, which not only rate the target entity unfairly but also camouflage themselves by rating non-target entity. There are 1384, 3476 and 3306 files with consistent attack and 159, 845, 3855 files with camouflage attack in weak, moderate and strong attack group respectively. Though no testing attack file explicitly shows the Whitewashing attack model, the original testing dataset incorporate this type of attack detection. There are two phases in Whitewashing attack: in the earlier phase, an old user account attacks the target entity via either consistent attack or camouflage attack; in the later phase, a new account behaves as the normal rater. Thus, we only need to filter out the old user account, as the new account is honest. The situation in each phase is contained in the original testing dataset. Hence, as long as the method could successfully detect user accounts for consistent attack and camouflage attack, the whitewashing attack can be defended by removing the old account in its earlier phase. For Sybil-based Attack models, despite its different fraudulent rater sizes, each single attack file is not enough for Sybil Attack, as the attacker size is not over 50%. However, Sybil Attack environment could be constituted by combining multiple attack files so as to evaluate the proposed method.

The effectiveness of the proposed method is tested in two cases. First, we evaluate the performance using the original attack data, which contains Non-Sybil Based Attacks. Second, we combine multiple attack files to test the robustness of the proposed method in Sybil-Base Attacks. To demonstrate the incremental accomplishment of our method, we compare our proposed approach against one well-known benchmark, the iCLUB (Liu et al. 2014). The radius value used for DBSCAN clustering is 0.3, which has shown to outperform other values and is also consistent with that suggested by Liu et al. (2014).

Single Attack File Detection Accuracy (Non-Sybil Based Attack)

We first detect the performance of the proposed method by adding each single attack file in the original testing dataset to the normal data, which creates 13,028 merged test files. The user IDs from the attack file, i.e. those start from $u_{301,}$ u_{302}....., are labeled as the fraudulent raters while user IDs (u_1,.....u_{300}) are honest raters. One user-entity rating matrix and one user-entity rating time matrix are generated from the ratings in each test file. Following the method proposed, suspicious entities are first identified and then a group of users are retrieved as the predicted fraudulent raters. The effectiveness of the method is evaluated by comparing the predicted fraudulent raters with the pre-labeled ones.

We first assess the accuracy of target entities detection, which is the precondition of the following fraudulent rater detection. We adopt *recall*, as defined in (3), to measure that, among all target entities, what is the percentage of the accurately detected target entity. Meanwhile, we use *precision*, as defined in (4), to calculate the percentage of the identified target entities that are accurately under attack. The overall detection performance is measures by *F1*, as shown in (5). To apply Ljung-Box test, we need to set the significance level. If the significance level is large (e.g., 0.05), there are more false positives, meaning that there are more non-target entities misclassified as target ones. When the significance level is small (e.g., 0.001), there are more false negatives (i.e. more undiscovered target entities). Since raters associated with entities selected in this step will be the input of the second step, we are more concerned with false negatives and use 0.01 as the significance level in the testing. For each testing file, we calculate

a pair of (recall, precision). Then, in every attack level group under each attack model, we calculate the average recall and precision, as shown in Table 3. Obviously, the proposed method can accurately detect the target-entity regardless of the attack model. Even in the weak attack group, where participants mostly pretend to be the normal users, the proposed method can still detect most target entity. Additionally, we could notice that stronger groups under camouflage attack have more non-target entities mistakenly identified as target entities. It is since their participants also inject ratings to the non-target entity, which make those entities distinguishable from others. However, misidentifying them as target ones may not hurt the fraudulent rater detection, as their ratings may come from the camouflaged fraudulent raters.

$$Recall = \frac{True\ positive}{True\ positive + False\ negative}, \tag{3}$$

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}, \tag{4}$$

$$F1 = \frac{2 * True\ positive}{2 * True\ positive + False\ negative + False\ positive} \tag{5}$$

Table 3. Performance of Target Entity Detection

		Weak	Medium	Strong
Consistent Attack	Recall	0.99	1	1
	Precision	0.97	1	1
	F1	0.98	1	1
Camouflage Attack	Recall	0.97	0.99	1
	Precision	0.97	0.96	0.95
	F1	0.97	0.97	0.97

Next, we evaluate the fraudulent rater detection performance of the proposed method and the benchmark. Table 4 summarizes the overall fraudulent rater detection precision, recall and F1 of each attack group in Consistent Attack and Camouflage Attack model respectively. The results confirm the effectiveness of the proposed method in detecting the fraudulent raters. We can observe that for the proposed method, the group with higher attack level has better performance in both precision and recall perspectives. The precision and the recall values in strong attack group are above 99% under Consistent Attack, while they are above 90% under Camouflage Attack, which is more difficult for detection. When the attack level is low, e.g. the weak attack group, the fraudulent raters behave similarly to the normal users, so that it is more difficult to differentiate them. Thus, the recall values in both attack models are relatively lower in the weak attack group, which has least impact on the accuracy of the systems. When compared to the benchmark, on one hand, we can see that the proposed method outperforms the iCLUB with its higher F1 value in every attack group and attack model. Moreover, its advantage increases under Camouflage Attack or under stronger attack. On the other hand, we can find that under Camouflage Attack, the recall of the iCLUB is higher than that of the proposed method in the weak attack group. The precision of the iCLUB in the weak attack group under Camouflage Attack, however, is below 0.5 (e.g. 0.31) and is lower than of the proposed method. It indicates that more honest than malicious ones would be removed from the systems, which will hurt the reliability of the systems. The results show effectiveness of the proposed method in discriminating the fraudulent raters while leaving the majority of the normal users in the systems.

Table 4. The Overall Performance for Single File Testing

Attack Model		Method	Weak	Moderate	Strong
Consistent Attack	Precision	Proposed	0.66	0.86	0.99
		iCLUB	0.65	0.86	0.99
	Recall	Proposed	0.49	0.85	1
		iCLUB	0.46	0.80	0.82
	F1	Proposed	0.57	0.85	0.99
		iCLUB	0.54	0.83	0.90
Camouflage Attack	Precision	Proposed	0.64	0.88	0.97
		iCLUB	0.31	0.71	0.63
	Recall	Proposed	0.33	0.75	0.92
		iCLUB	0.68	0.75	0.75
	F1	Proposed	0.44	0.81	0.95
		iCLUB	0.43	0.68	0.73

A more detailed analysis on the precision and recall in each attack level group for the two methods is presented in Figure 3 and Figure 4. The precision/the recall result of every testing file, is categorized into five intervals including [0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1.0] respectively. For every attack level group, we calculate its own frequency in each interval. Then we plot the frequency distribution of the precision and the recall in each group with Consistent Attack (Figure 3) and Camouflage Attack (Figure 4) respectively. In both figures, the x-axis represents the precision/recall interval and the y-axis is the relative frequency value in every attack group. Obviously, for the proposed method, almost every testing file in the strong attack group has very high precision and recall values. For the iCLUB, however, there are still a number of files in the strong attack group has very low recall under Camouflage Attack Environment. For the weak attack group using both methods, it has around 20% testing files with the recall values below 20% under two attack models, which means only 20% of the fraudulent raters are successfully identified. However, the ratings from the undiscovered fraudulent raters may not affect the reliability of the item reputation largely due to their near to normal behavior. Meanwhile, we observe that the weak attack group using the proposed method has a relatively better performance in precision, with around 2% below 0.2 in both attack models. It indicates that the raters removed are less likely to be the honest users so that the rating quality can retain. Hereby, even though the detection performance in the weak attack group is not high, the reputation systems using the proposed method may still maintain its accuracy. However, it is not the case for the iCLUB method, which has majority low precision values in the weak attack group, particularity under Camouflage Attack.

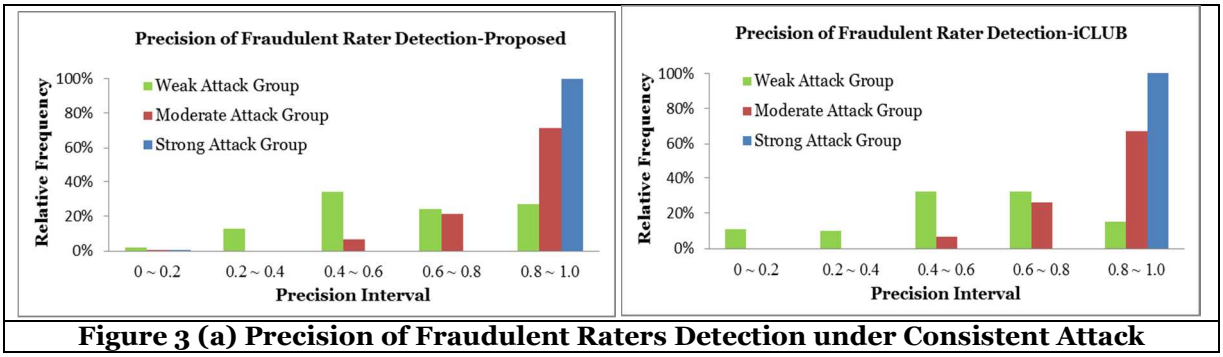

Figure 3 (a) Precision of Fraudulent Raters Detection under Consistent Attack

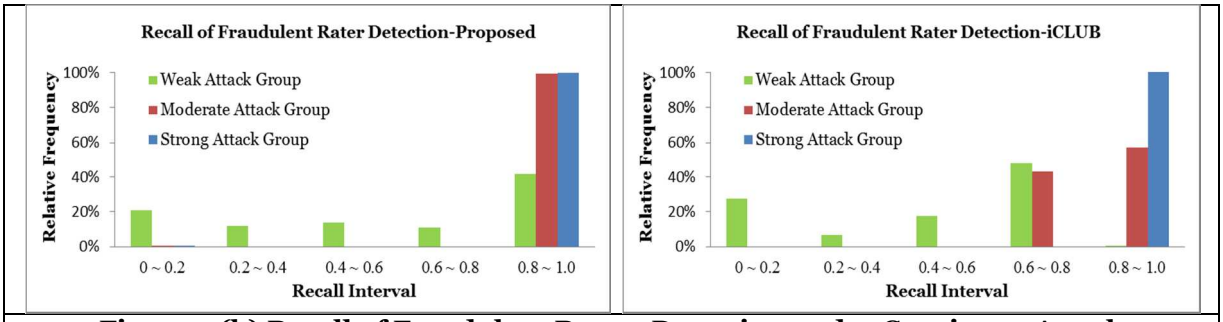

Figure 3 (b) Recall of Fraudulent Raters Detection under Consistent Attack

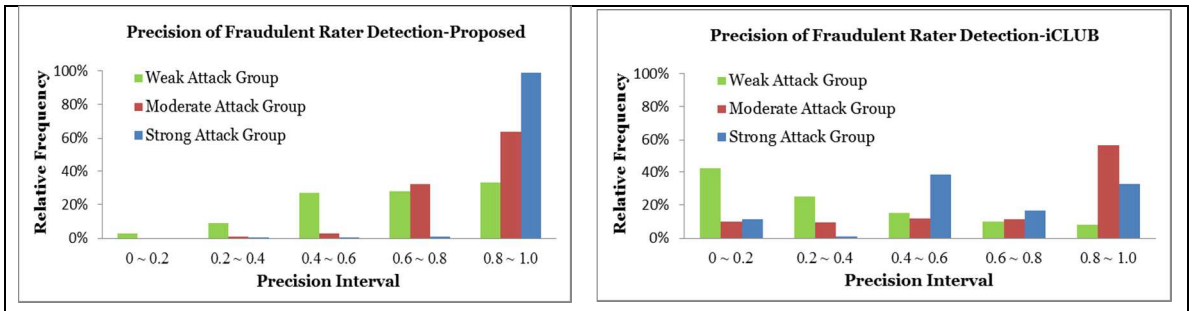

Figure 4 (a) Precision of Fraudulent Raters Detection under Camouflage Attack

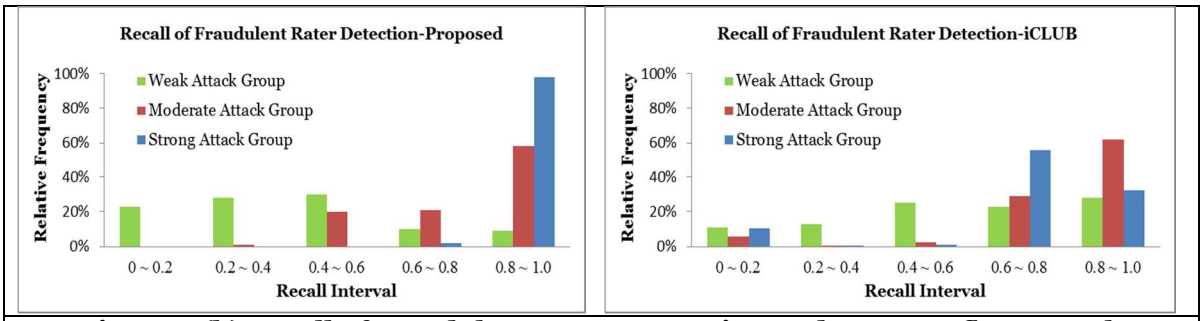

Figure 4 (b) Recall of Fraudulent Raters Detection under Camouflage Attack

Based on the above observations, we further assess how the rating of the target entity has been affected after removing all the predicted fraudulent raters by recalculating its rating shift for all 13028 testing files. As introduced earlier, we calculate the difference between the original rating before attack and the current rating after attack detection. The average rating shift for the strong attack group, moderate attack group and weak attack group is 0.0005, 0.0108, and 0.0084 respectively. Compared to the original rating shift interval for each group, the impact from the fraudulent raters is trivial. Figure 5 displays the frequency distribution for the rating shift after the detection of every testing file. The x-axis represents three rating shift levels and the y-axis denotes the relative frequency of the rating shift for each group. Clearly, zero is the majority rating shift value in each group after applying the proposed method. For both the moderate and the weak attack group, most of their rating shifts have been significantly eliminated. And in the strong attack group, more than 97% of testing files have returned to the normal rating.

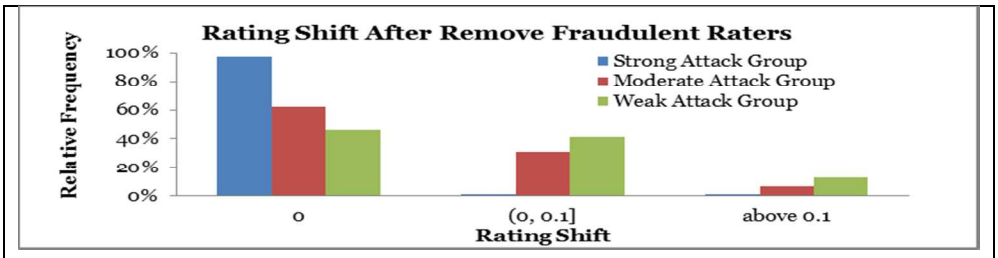

Figure 5. Rating Shift after Detection in Various Attack Levels

Sybil-Based Attacks Detection Testing

Although the original testing dataset has presented a variety of fraudulent rater's behaviors, the size of the fraudulent rater is limited. Various sizes of the fraudulent rater, particularly those above 50%, can be used to examine the robustness of the proposed method under Sybil-Based Attacks. We need to create new attack files to represent various fraudulent rater sizes. Fraudulent rater size is the percentage of the raters who are fraudulent in the attack file. For example, 25% means there are 0.25*p user IDs in the attack file, where p is the total number of raters in the reputation systems. Four different fraudulent rater sizes are used, i.e. 25%, 40%, 60% and 85%, where the latter two sizes are Sybil-Based Attacks. We construct attack files for every fraudulent rater size. When a fraudulent rater size is selected, the corresponding number of the fraudulent raters is fixed. For example, suppose the number of the raters is M. In each attack group under each attack model of the original attack data (i.e. weak, moderate and strong attack group under Consistent or Camouflage Attack), we randomly select a certain amount of attack files which have a total of m fraudulent raters. We adjust the user ID in different attack files to make sure no duplicates. Then the selected attack files are merged together in each attack group respectively. After applying the proposed method, the precision and the recall of the fraudulent rater detection is recorded in all three attack groups. For each attack size, we repeat the experiments 1000 times and make sure that there is no same combination of the files.

Table 5 shows the average performance results for the both methods from three attack level (i.e. weak, moderate, and strong) and four attack models (i.e. Consistent Attack, Camouflage Attack, Sybil-Consistent

Attack and Sybil-Camouflage Attack). We could find that the proposed method outperforms the iCLUB in every attack environment. In addition, the impact of the fraudulent rater size on the performance for the two methods is different. For the proposed method, the detection performance is generally improving with the increasing of the attack size, regardless of the attack group or attack model. For the strong attack group in all attack models, the performance always maintains a very high-level accuracy even if the size is small. For the moderate attack group, when the attack size is above 40%, its F1 measure values are above 95% in all four models. For the weak attack group, its detection performances improve from Non-Sybil Based Attacks to Sybil-Based Attacks. . This is since when the number of the fraudulent raters increases, more users share the similar behaviors so that they have a higher chance to be detected.

For the iCLUB, however, the impact of the fraudulent rater on detection performance is different in four attack models. Under Camouflage and Sybil-Camouflage Attack, the performance is generally decreasing with the increasing of the attack size, in particular for the weak attack group. Also, under Sybil-Consistent Attack, the detection performance decreases significantly at the higher fraudulent rater size (i.e. 85%) in the weak attack group. Only under the Consistent Attack, the iCLUB maintains the decent performance in all attack groups and improves with the increasing fraudulent rater size. Therefore, the performance of the iCLUB is affected by the larger size of the fraudulent rater and the weaker attack level. When the attack size increases, there are more fraudulent raters than the honest users in the system. For the iCLUB, when the honest rater relies on the global knowledge to evaluate the reputation, it will be inaccurate when the majority is attacker. In addition, in the weak attack group, the dishonest rater behaves similar to the normal ones. For the iCLUB, when the honest rater use its own rating experience to detect the dishonest rater based on the similarity, the decision will be biased when the dishonest rater camouflage themselves or inject weak attack. Herein, the performance of iCLUB is vulnerable to Camouflage-Based Attacks (i.e. Camouflage and Sybil Camouflage Attack) and is not stable under Sybil Attack.

Table 5. Fraudulent Rater Detection Performance under Different Rater Sizes

Attack Group Level	Measure		Fraudulent Rater Size							
			25%	40%	60%	85%	25%	40%	60%	85%
		Attack Model	**Consistent**		**Sybil- Consistent**		**Camouflage**		**Sybil- Camouflage**	
Weak Attack Group	Precision	Proposed	0.76	0.80	0.84	0.92	0.83	0.91	0.97	0.98
		iCLUB	0.49	0.59	0.65	0.20	0.18	0.09	0.00	0.00
	Recall	Proposed	0.93	0.96	0.97	0.98	0.85	0.93	0.93	0.98
		iCLUB	0.74	0.91	0.96	0.20	0.22	0.69	0.00	0.00
	F1	Proposed	0.84	0.87	0.90	0.95	0.84	0.92	0.95	0.98
		iCLUB	0.59	0.72	0.78	0.20	0.20	0.17	0.00	0.00
Moderate Attack Group	Precision	Proposed	0.95	0.96	0.97	1.00	0.92	0.95	0.96	0.98
		iCLUB	0.87	0.96	0.98	0.98	0.71	0.41	0.04	0.00
	Recall	Proposed	0.94	0.97	0.97	1.00	0.93	0.95	0.96	0.98
		iCLUB	0.92	0.94	0.97	0.98	0.75	0.76	0.69	0.00
	F1	Proposed	0.94	0.96	0.97	1.00	0.92	0.95	0.96	0.98
		iCLUB	0.89	0.95	0.97	0.98	0.73	0.53	0.08	0.00
Strong Attack Group	Precision	Proposed	0.97	0.98	0.98	1.00	0.95	0.96	0.99	1.00
		iCLUB	0.98	0.99	0.98	1.00	0.82	0.90	0.70	0.00
	Recall	Proposed	0.99	1.00	1.00	1.00	0.99	0.99	1.00	1.00
		iCLUB	0.98	0.94	0.97	0.98	0.88	0.94	0.62	0.00
	F1	Proposed	0.98	0.99	0.99	1.00	0.97	0.97	0.99	1.00
		iCLUB	0.98	0.97	0.97	0.99	0.85	0.92	0.66	0.00

Conclusions

Due to the anonymity in the Internet, it may be risky to interact with unfamiliar items or strange sellers. Reputation systems have been shown to be effective for customers to judge the quality of the object and reduce the interaction-specific risk. However, reputation systems are vulnerable to rating fraud, which will mislead the customers and further affect their motivation in participating into the future interaction. In this paper, we address the rating fraud issue and design the fraudulent rater detection method to improve the reliability of reputation systems. By discovering certain temporal characteristics of both the

target entity rating series and the fraudulent rater rating series, we could discriminate the target entity and cluster the corresponding dishonest raters. Several experiments are conducted to validate the performance of the proposed method by using the real-world cyber competition data. Our proposed method has shown its performance advantage over the benchmark. Moreover, our methods have shown its robustness in various attack environments including the Sybil Attack, Consistent Attack and Camouflage Attack. The method proposed in this study could facilitate the organizations relying on the reputation systems for their better customer retention. It could also help reduce the financial risk associated with the e-commerce transactions.

Our study focused on the collaborative rating fraud since it has more significant impact on the systems. Thus, our proposed method may be vulnerable to the singleton rating fraud, which means the fraudulent rater does not have collaborators but just inject the unfair rating independently. For singleton fraudulent raters, they can be detected by examining the behavior deviation from the general distributions. The development of a more comprehensive detection framework by strategically combining our method with other research streams is also planned for future analysis.

References

Adler, B. and de Alfaro, L. 2007. "A content-driven reputation system for the Wikipedia". *In Proceedings of the 16th international conference on World Wide Web (WWW)*. ACM Press, New York, NY, USA, pp: 261–270.

Azari, R., 2003. *Current Security Management & Ethical Issues of Information Technology*. Hershey, PA: Idea Group Publishing.

Ba, S., & Pavlou, P. 2002. "Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior". *MIS Quarterly*, (26**:**3), pp: 243-268.

Bromley D. B. 2001. "Relationships between personal and corporate reputation". *European Journal of Marketing,* (35:3), pp: 316-334.

Chang, J., 2013. *Samsung Fined For Paying People to Criticize HTC's Products*. Available at *http://abcnews.go.com/Technology/samsung-fined-paying-people-criticize-htcs-products/story?id=20671547*

Chatterjee, K., de Alfaro, L., Pye, I., 2008. "Robust Content-Driven Reputation". *Proceedings of the 1st ACM workshop on Workshop on AIS,* New York, NY, USA. pp: 33-42.

Dellarocas, C. 2000. "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior". *Proceedings of the 2nd ACM Conference on Electronic commerce,* pp:150-157.

Douceur, J. 2002. "The sybil attack". *IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, London, UK. pp:251-260

Fei, L., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. 2013."Exploiting Burstiness in Reviews for Review Spammer Detection" .*Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013)*, Boston, USA.

Goel, V., 2014. "User Growth for Twitter Starts to Slow, and Stock Dips". Available at *http://www.nytimes.com/2014/02/06/technology/twitters-share-price-falls-after-it-reports-4th-quarter-loss.html?_r=1*

Houser, D., and Wooders, J., 2006. "Reputation in auctions: Theory, and evidence from eBay". *Journal of Economics and Management Strategy,* (15:2), pp. 353-369.

Hu, N., Liu, L., and Sambamurthy, V. 2010. "Fraud detection in online consumer reviews". *Decision Support Systems,* (50:3), pp:614-626.

Hu, N., Bose, I., Koh, N, and Liu, L 2012. "Manipulation of online reviews: An analysis of ratings, readability, and sentiments"., *Decision Support Systems*, (52). pp 674-684.

Irissappane, A. A., Jiang S., and Zhang, J. 2012. "Towards a Comprehensive Testbed to Evaluate the Robustness of Reputation Systems against Unfair Rating Attacks*". UMAP Workshops, volume 872 of CEUR Workshop Proceedings*.

Jindal, N., and Liu, B. 2008. "Opinion Spam and Analysis". *Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM-2008)*, Stanford, California,

Josang, A., and Ismail, R. 2002. "The beta reputation system," *Proceedings of the 15th Bled Electronic Commerce conference*.

Li, X., and Hitt, L. M. 2008. "Self selection and information role of online product reviews". *Information Systems Research*, (19:4), pp:456-474.

Lim, E., Nguyen, V., Jindal, N., Liu, B., and Lauw, H. 2010. "Detecting Product Review Spammers using Rating Behaviors". *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, Canada.

Liu, Y., Sun, Y. L., and Yu, T. 2011. "Defending Multiple-User-Multiple-Target Attacks in Online Reputation Systems". *IEEE Third International Conference on Social Computing*. pp:425-434.

Liu, S., Yu, H., Miao, C., and Kot, A. C. 2013. "A fuzzy logic based reputation model against unfair ratings". *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. pp:821-828

Liu, S., Zhang, J., Miao, C., Theng, Y., and Kot, A., 2014. "Iclub: an integrated clustering based approach to improve the robustness of reputation systems", *Computational Intelligence*, (30:2) pp: 316-341.

Ljung, G.M. and Box, G.E.P., 1978, "On a measure of lack of fit in time series models", *Biometrika* (65), pp: 297-303.

Mild, A., and Reutterer, T., 2003. "An Improved Collaborative Filtering Approach for Predicting CrossCategory Purchases Based on Binary Market Basket Data", *Journal of Retailing and Consumer Services*, (10), pp: 123-133.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., & Ghosh, R. 2013, "Spotting Opinion Spammers using Behavioral Footprints",*Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, USA.

Parker, S. 2011. "3 Tips for Spotting Fake Product Reviews – From Someone Who. Wrote Them". Available at *http://www.moneytalksnews.com/2011/07/25/3-tips-for-spotting-fake-product-reviews-%E2%80%93-from-someone-who-wrote-them/*

Resnick, P., Kuwabara, K., Zeckhauser, R., and Friedman E. 2000. "Reputation systems". *Communications of the ACM,* (43:12), pp:45-48.

Resnick, P., & Zeckhauser, R., 2002,"Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System", *The Economics of the Internet and E-commerce*, (11:2), pp: 23-25.

Smith, C. (2015). 33 Amazing Amazon Statistics. Available at *http://expandedramblings.com/index.php/amazon-statistics/#.Uzs4A_ldWSo*

Soldo, F., 2011. "Blacklisting Recommendation System: Using Spatio-Temporal Patterns to Predict Future Attacks", *IEEE Journal on Selected Areas in Communications,* (29:7), pp.1423-1437.

Sorrel,C.,2009, "Apple expels 1,000 apps after store scam", Available at *http://edition.cnn.com/2009/TECH/12/09/wired.apple.apps/index.html*

Sved, D., 2014, "Nineteen companies found guilty of writing fake consumer reviews", Available at *http://www.heralddeparis.com/nineteen-companies-found-guilty-of-writing-fake-consumer-reviews/232920*

Teacy, W., Patel, J., Jennings, N. and Luck, M. 2006. "Travos: Trust and reputation in the context of inaccurate information sources". *Autonomous Agents and Multi-Agent Systems*, (12:2): pp:183-198.

Tsay, R. S. 2010. *Analysis of Financial Time Series* (3rd ed.). New Jersey: Wiley & Sons

Wang, G., Xie, S., Liu, B., and Yu, P. S. 2011. "Identify Online Store Review Spammers via Social Review Graph". *ACM Transactions on Intelligent Systems and Technology*.

Welch, C and Popper, B. (2015) "Twitter reaches 300 million active users, but the stock crashes after earnings leak early". http://www.theverge.com/2015/4/28/8509855/twitter-earnings-q1-2015-leak-selerity

Whitby, A., Josang, A., and Indulska, J. 2004. "Filtering out unfair ratings in bayesian reputation systems". *Proceedings of the 7th Int. Workshop on Trust in Agent Societies.*

Xie, S., Wang, G. Lin, S. Yu, P. S, 2012. "Review Spam Detection via Temporal Pattern Discovery". *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp:823-831.

Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A., 2006, "SybilGuard: defending against Sybil attacks via social networks", *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, New York, USA. pp: 267-278.

Zhu, H., Xiong, H., Ge, Y., and Chen, E. 2013. "Ranking fraud detection for mobile apps: a holistic view". *Proceedings of the 22nd ACM international conference on information & knowledge management,* pp:619-628.