Dishonest Behaviors in Online Rating Systems: Cyber Competition, Attack Models, and Attack Generator

Yafei Yang†, Qinyuan Feng*, Yan Lindsay Sun† and Yafei Dai*

† University of Rhode Island, Kingston, RI, USA

*CNDS Lab, Peking University, Beijing, China

Abstract

Recently, online rating systems are gaining popularity. Dealing with unfair ratings in such systems has been recognized as an important but challenging problem. Many unfair rating detection approaches have been developed and evaluated against simple attack models. However, The lack of unfair rating data from real human users and realistic attack behavior models has become an obstacle toward developing reliable rating systems. To solve this problem, we design and launch a rating challenge to collect unfair rating data from real human users. In order to broaden the scope of the data collection, we also develop a comprehensive signal-based unfair rating detection system. Based on the analysis of real attack data, we discover important features in unfair ratings, build attack models, and develop an unfair rating generator. The models and generator developed in this paper can be directly used to test current rating aggregation systems, as well as to assist the design of future rating systems.

I. Introduction

With the evolutionary development of E-Commerce, online feedback-based rating systems are gaining popularity. While these systems are increasing influence on today's consumers, ensuring trustworthiness of such systems remains as an important and challenging task [1]–[4]. The major challenge toward building a trustworthy online rating system is to deal with *unfair ratings* from *dishonest raters*. In commercial systems, it has been observed that collaborative dishonest raters provide unfair ratings intentionally to boost or downgrade the rating scores of certain products or reputation of other users [5].

This work is partial supported by NSF(award 0643532), NSFC(award 60673183), and Doctoral Funding of MOE(20060001044). The early version has been presented in the IEEE ICDCS Workshop on Trust and Reputation Management (Tram 2008).

There have been many approaches proposed to deal with unfair ratings [2], [3], [6]–[9]. However, most of them are evaluated against simple attack behaviors models, in which assumptions are made to greatly simplify the behavior of dishonest raters. For example, the arrival of unfair ratings is often assumed to be a Poisson process with a fixed arrival rate, and the unfair rating values are often assumed to follow a Gaussian or uniform distribution. However, most of the assumptions have not been validated by real user data and the simplification has not been well justified. More importantly, these simple models cannot reflect the smart attacks from real human users who can always adjust the attack strategies based on their observation of original rating data and gain knowledge about the rating system. More realistic and possibly complicated models of dishonest raters need to be developed.

The development of dishonest behavior model faces two challenges. The *first* is the lack of unfair rating data from real human users. Although there are plenty of unfair ratings in commercial systems, there is no ground truth telling which ratings are unfair ratings and which are not. *Second*, the attack behaviors are affected by the defense system. In other words, the dishonest raters may behave differently when different rating aggregation algorithms are used. For example, when simple averaging is used as the rating aggregation algorithm, providing the largest or smallest possible rating values is the most effective attack strategy. When majority-rule based detection algorithms, such as [2], are used, dishonest raters may provide rating values not too far away from the majority's opinion. When the signal-based detection algorithm, such as [9], are used, more complicated attack behaviors are expected. Here, the signal-based algorithm is a novel detection method based on signal modeling. In particular, this method treats fair ratings as *noise* and unfair ratings as *signal*. The algorithm models the overall ratings using an autoregressive (AR) signal modeling technique and examines the model errors. The model error is proved to be a good indicator of whether the 'signal' (unfair ratings) is present. This detection method does not rely on the majority rule.

To address the above challenges, we first design and launch a *rating challenge* to collect attack data from real human users. Moreover, we extend the approach in [9] and design a new rating aggregation algorithm, which includes most of the latest and complex defense strategies, such as feedback reputation and signal-based unfair rating detection. By using the new rating aggregation algorithm in the rating challenge, we are able to collect the real attacks against it such that the collected data cover a broad range of smart attack behaviors. Then, we analyze the attack behavior of real human users and evaluate

the performance of both the complex and simple defense schemes against real attacks. Many important results are obtained. Especially, we were able to classify attacks according to the unfair rating values as well as the time when the unfair ratings are provided. Finally, we build novel attack models as well as a comprehensive unfair rating generator. The attack models provide an in-depth understanding on the behavior of dishonest raters, which provides guidelines for future system design. The unfair rating generator is also a great tool for the research community and industry to perform realistic evaluation of rating aggregation systems.

The rest of the paper is organized as follows. Related work and the rating challenge are discussed in Section II. The algorithms used in the challenge is described in Section III. The attack data analysis is presented in Section IV, followed by the conclusion in Section V.

II. **Related Work and Rating Challenge**

A. Related Work

As discussed in Section I, the evaluation of rating algorithms mostly rely on simple attack models. For example, [10] [11] [12] only considered the probability of lying; [13] only considered the percentage of unfair raters and the strategies of either bad-mouthing and ballot-stuffing; [9] [14] considered the fraction of unfair rating and the unfair rating following a simple distribution; [15] considered some types of collusion. However, none of these models are built from real world data, and the parameters in their probability models are seldom changed. As we discussed later, these are critical in the evaluation of the rating algorithms. In addition, [4] built a statistical model for honest entities in decentralized systems, which could help to identify the suspicious entities. Relatively, our work is focusing on the dishonest behavior modeling.

Trust establishment is another key element in some rating aggregation systems. There is a rich literature on trust establishment for electronics commerce, peer-to-peer networks, distributed computing, ad hoc and sensor networks, and social networking systems [16]–[21]. For rating aggregation problem, simple trust models are used to calculated trust in raters in [7], [22], [23]. However, their effectiveness is restricted due to the limitation of the underlying detection algorithms.

Cyber Challenges are effective ways to collect real user data. For example, there is the Netflix Challenge [24] whose purpose is to build a better recommendation system based on user ratings. This purpose is

very different from the focus of this work and the data collected in the Netflix challenge is not suitable for studying the unfair rating problem. So we launched a **Rating Challenges** [25], which is described in Section II-B.

B. Rating Challenge

As pointed in [9], the unfair ratings can be classified into two categories:

- *Individual unfair ratings*: an individual rater provides unfairly high or low ratings without collaborating with other raters. This type of ratings may result from raters' personality/habit (i.e. dispositional trust [26]), irresponsibleness, and randomness.
- *Collaborative unfair ratings*: a group of raters provide unfairly high or low ratings to boost or downgrade the overall ratings of an object. This type of rating may due to the strategic manipulation from the owner of the object [5].

Compared with collaborative unfair ratings, individual unfair ratings are much less harmful. Therefore, *the focus of this work is to deal with collaborative unfair ratings.*

In order to investigate dishonest rating behaviors of real human users, we launched a **Rating Challenges** [25], in which participants insert collaborative unfair ratings into a regular rating data set. The participants who mislead the aggregated rating scores the most can win a cash prize. In this challenge,

- We collected real online rating data for 9 flat panel TVs with similar features. The data are from a well-known online-shopping website.
- The participants download the rating dataset and control 50 biased raters to insert unfair ratings. In particular, the participants decide when the 50 raters rate, which products they rate for, and the rating values.
- The participants' goal is to boost the ratings of two products and reduce the ratings of another two products.
- The successfulness of the participants' attack is determined by the **overall manipulation power**, i.e MP value. For each product, we calculate $\Delta_i = |R_{ag}(t_i) - R^o_{ag}(t_i)|$ during every 30 day period, where $R_{ag}(t_i)$ is the aggregated rating value with unfair ratings, and $R^o_{ag}(t_i)$ is the aggregated rating value without unfair ratings. The overall MP value is calculated as $\sum_k (\Delta^k_{max_1} + \Delta^k_{max_2})$, where $\Delta^k_{max_1}$ and $\Delta^k_{max_2}$ are the largest and 2nd largest among $\{\Delta_i\}'s$ for product k.

- The participants that can generate the largest MP value win the competition.

In practice, when merchants intend to make more profits through unfair ratings, they have many options. They may boost their own products or downgrade the rivals' products for a relatively short duration (e.g. a few months) or during the lifetime of the product. Recall that each user ID can only rate once. Given a fixed number of user IDs, the attacker either promote the product a tiny bit for a very long time or promote it to a meaningful score during several months. The MP score encourages the participants to conduct the attack during two months. In the rating challenge, however, we observed that the participants' behavior is very diverse. The attack duration ranges from a few days to the entire rating time.

III. Reliable Rating Aggregation System in the Rating Challenge

Although it is desirable to collect attack data targeting many rating aggregation systems, we must choose one rating aggregation system for the rating challenge. During our investigation, we observe that the attacks against majority-rule-based unfair rating detection systems are *straightforward*.

- As pointed in [9], when there are a sufficient number of dishonest raters, the unfair ratings can become the majority and totally disable the majority-rule based methods. In other words, when the majority of ratings are unfair in certain time intervals, the majority-rule-based methods would totally *fail*.
- When there is no sufficient dishonest raters, the best attack strategy is to provide higher or lower ratings that are not too far away from the majority.

Therefore, if the rating challenge adopts one of the majority-rule-based methods as the underlying unfair rating detection algorithm, we can predict the participants' attack methods very well. There is no incentive for the participants to create more complicated attacks.

On the other hand, the investigation on the signal-based unfair rating detection methods is still in its early stage. Signal-based methods can handle straightforward attacks but the effective attacks against them are unknown. If the signal-based methods are used in the rating challenge, the participants can be encouraged to exploit new and complicated attacks. This will make the rating challenge more meaningful and useful.

However, the current signal-based methods are still not mature. The method reported in [9] is designed for a specific type of attack but cannot handle a wide range of attacks. This motivates us to design

an advanced signal-based reliable rating aggregation system, and use it in the rating challenge. This advanced system is described in the following subsections. Due to space limitation, the specific design considerations and derivations are omitted. The readers who are only interested in attack modeling can go to Section IV-A first to read the attack data analysis and then go back to Section III for details in the underlying unfair rating detection algorithms in the rating challenge.

A. Rating Aggregation Overview

The rating aggregation process contains four steps.

First, raw ratings are analyzed. Four analysis methods, arrival rate detection, model change detection, histogram detection and mean change detection, are applied independently.

Second, the outcomes of four detectors are combined to detect the time intervals in which unfair ratings are highly likely. Additionally, the suspicious rating detection module can mark some specific ratings as suspicious.

Third, we design a trust manager by simplifying the generic framework of trust establishment proposed in [27]. The trust manager determines how much individual raters can be trusted.

Fourth, the highly suspicious ratings are removed from the raw ratings by a rating filter. Then, the ratings are combined using trust information by the rating aggregation algorithm.

B. Mean Change Detector

The mean change detector contains three parts.

1) Mean Change Hypothesis Test: For one product, let $t(n)$ denote the time when a particular rating is given, $x(n)$ denote the value of the rating, and $u(n)$ denote the IDs of the rater. That is, at time $t(j)$, rater $u(j)$ submits a rating for the product with rating value $x(j)$, where $j = 1, 2, \cdots, N$ and N is the total number of ratings for this product.

We first study the mean change detection problem inside a window. Assume that the window contains $2W$ ratings. Let X_1 denote the first half ratings and X_2 denote the second half ratings in the window. We model X_1 as an i.i.d Gaussian random process with mean A_1 and variance σ^2, and X_2 as an i.i.d Gaussian random process with mean A_2 and variance σ^2. Then, to detect the mean change is to solve

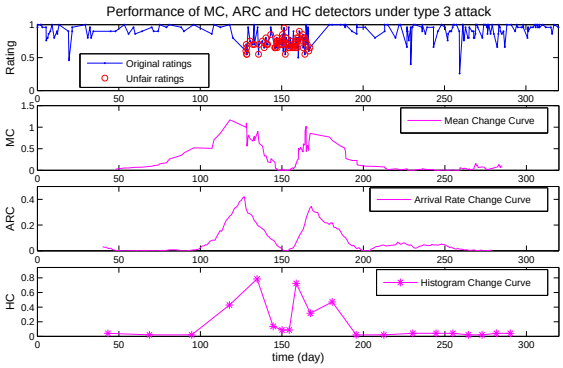

Fig. 1: Illustration of MC, ARC and HC detection (attack duration: 40 days, bias: 0.2, variance: 0.5 $\times$ variance of honest ratings, arrival rate: 3 $\times$ arrival rate of the honest ratings.)

the hypothesis testing problem

$$\mathcal{H}_0 : A_1 = A_2$$

$$\mathcal{H}_1 : A_1 \neq A_2.$$

It has been shown in [28] that the Generalized Likelihood Ratio Test (GLRT) is

$$\text{Decide } \mathcal{H}_1 \text{ (i.e. there is a mean change), if}$$
$$2 \ln L_G(x) = \frac{W(\hat{A}_1 - \hat{A}_2)^2}{2\sigma^2} > \gamma \tag{1}$$

where $\hat{A}_1$ is the average of X_1 and $\hat{A}_2$ is the average of X_2, and γ is a threshold.

2) Mean Change Indicator Curve: Second, the detector constructs the mean change indicator curve using a sliding window with window size W. Based on (1), the mean change indicator curve is constructed as $MC(k)$ versus $t(k)$, where $MC(k)$ is the value of $W(\hat{A}_1 - \hat{A}_2)^2$ calculated for the window containing ratings $\{x(k-W), \cdots, x(k+W-1)\}$. In other words, the test in (1) is performed to see whether there is a mean change at the center of the window.

The example of mean change indictor curve is shown in Figure 1. The top plot shows the rating data $x(n)$ vs. $t(n)$. The blue dots represent the rating values for a flat panel TV (the first data set) in the rating challenge [25], the red $\circ$ represent the unfair ratings added by simulation. On the MC curves (the 2nd plot), the U-shape (two peaks) clearly shows the beginning and end of the attack.

3) MC Suspiciousness: Based on the peak values on the mean change indicator curve, we detect the time interval in which abnormal mean change occurs. This interval is called *mean change (MC) suspicious interval*.

When there are only two peaks, the MC suspicious interval is just between the two peaks. When there are more than 2 peaks, it is not straightforward to determine which time interval is suspicious. We use trust information to solve this problem. In particular, we divide all ratings into several segments, separated by the peaks on the mean change indicator curve. Assume there are M segments. In each segment, the mean value of ratings are calculated as B_j for $j = 1, 2, \cdots, M$. And B_{avg} is the mean value of the overall ratings. A segment j is marked as ***MC suspicious*** if either of the following conditions is satisfied:

1) $|B_j - B_{avg}| > threshold_1$. That is, there is a very large mean change.
2) $|B_j - B_{avg}| > threshold_2$ and T_j/T_{avg} is smaller than a threshold, where T_j is the average trust value of the raters in the j^{th} segment, T_{avg} is the average trust value of the raters in all segments. Here, $threshold_2 < threshold_1$. This condition says that there is a moderate mean change and the raters in the segment is less trustworthy.

C. Arrival Rate Change Detector

1) Arrival Rate Change Hypothesis Test: For one product, let $y(n)$ denote the number of ratings received on day n. We first study the arrival rate detection problem inside a window. Assume that the window covers $2D$ days, starting from day k. We want to detect whether there is an arrival rate change at day k', for $k < k' < k + 2D - 1$.

Let $Y_1 = [y(k), y(k+1), \cdots, y(k'-1)]$ and $Y_2 = [y(k'), y(k'+1), \cdots, y(k+2D-1)]$. It is assumed that $y(n)$ follow Poisson distribution. Then, the joint distribution of Y_1 and Y_2 is

$$p[Y_1, Y_2; \lambda_1, \lambda_2] = \prod_{j=k}^{k'-1} \frac{e^{-\lambda_1} \lambda_1^{y(j)}}{y(j)!} \prod_{j=k'}^{k+2D-1} \frac{e^{-\lambda_2} \lambda_2^{y(j)}}{y(j)!}, \tag{2}$$

where λ_1 is the arrival rate per day from day k to day $k'-1$, and λ_2 is the arrival rate per day from day k' to day $k + 2D - 1$. To detect the arrival rate change is to solve the hypothesis testing problem

$$\mathcal{H}_0 : \lambda_1 = \lambda_2$$

$$\mathcal{H}_1 : \lambda_2 \neq \lambda_1$$

It is easy to show that

$$p[Y_1, Y_2; \lambda_1, \lambda_2] = \frac{e^{-a\lambda_1} \lambda_1^{a\bar{Y}_1}}{\prod_{j=k}^{k'-1} y(j)!} \cdot \frac{e^{-b\lambda_2} \lambda_2^{b\bar{Y}_2}}{\prod_{j=k'}^{k+2D-1} y(j)!}. \tag{3}$$

where

$$\bar{Y_1} = \frac{1}{a}\sum_{j=k}^{k'-1} y(j), \quad \bar{Y_2} = \frac{1}{b}\sum_{j=k'}^{k+2D-1} y(j),$$
$$a = k' - k, \qquad b = k - k' + 2D.$$

A GLRT decides $\mathcal{H}_1$ if

$$\frac{p[Y_1, Y_2; \hat{\lambda_1}, \hat{\lambda_2}]}{p[Y_1, Y_2; \hat{\lambda}, \hat{\lambda}]} > \gamma, \tag{4}$$

where $\hat{\lambda_1} = \bar{Y_1}$, $\hat{\lambda_2} = \bar{Y_2}$, and $\hat{\lambda} = \frac{1}{2D}(\sum_{j=k}^{k+2D-1} y(j)) = \bar{Y}$. Taking logarithm at both sides of (4), we derive

Decide $\mathcal{H}_1$(i.e. there is an arrival rate change) if

$$\frac{a}{2D}\bar{Y_1}\ln\bar{Y_1} + \frac{b}{2D}\bar{Y_2}\ln\bar{Y_2} - \bar{Y}\ln\bar{Y} \geq \frac{1}{2D}\ln\gamma. \tag{5}$$

2) Arrival Rate Change Curve: Based on (5), the Arrival Rate Change (ARC) curve is constructed as $ARC(k')$ vs $t(k')$. Here, the k' value is chosen as the center of the sliding window, i.e. $k' = k + D$. When $D < k' < N - D + 1$, $ARC(k')$ is just the left-hand side of equation (5) with $a = b = D$. When $k' \leq D$ or $k' \geq N - D + 1$, $ARC(k')$ can be calculated using a smaller window size, similar as the approach used in Section III-B2.

3) ARC Suspiciousness: Based on the peaks on the ARC curve, we divide all ratings into several segments. If the arrival rate in one segment is higher than the arrival rate in the previous segment and the difference between the arrival rates is larger than a threshold, this segment is marked as *ARC suspicious*.

4) H-ARC and L-ARC: For some practical rating data, the arrival rate of unfair ratings is not very high or the poisson arrival assumption may not hold. For those cases, we design H-ARC, which detects the arrival rate change in high value ratings, and L-ARC, which detects the arrival rate change in low value ratings.

Let $y_h(n)$ denote the number of ratings that are higher than $threshold_a$ received on day n, and $y_l(n)$ denote the number of ratings that are lower than $threshold_b$ received on day n. The $threshold_a$ and $threshold_b$ are determined based on the mean of all ratings.

- H-ARC detector: replace $y(n)$ in the ARC detector by $y_h(n)$
- L-ARC detector: replace $y(n)$ in the ARC detector by $y_l(n)$.

D. Histogram Change Detector

Unfair ratings can change histogram of the rating data. We design a histogram change detector based on clustering technique. There are two steps.

- 1. Within a time window k with the center at t_k, constructed two clusters from the rating values using the simple linkage method. The Matlab function $clusterdata()$ is used in the implementation.
- 2. The Histogram Change (HC) curve, $HC(k)$ versus t_k, is calculated as

$$HC(k) = \min\left(\frac{n_1}{n_2}, \frac{n2}{n1}\right), \tag{6}$$

  where n_1 and n_2 denote the number of ratings in the first and the second cluster, respectively.

E. Signal Model Change Detector

The signal model change detector is just the detector used in [9].

- Model-error-based detection: the ratings in a time window are fit into an autoregressive (AR) signal model. The model error is examined. When the model error is high, $x(n)$ is close to a white noise, i.e. honest ratings. When the model error is small, a 'signal' is present in $x(n)$ and the probability that there are collaborative raters is high.

The *model error (ME) curve* is constructed with the vertical axis as the model error, and horizontal axis as the center time of the windows. The windows are constructed either by making them contain the same number of ratings or have the same time duration. The covariance method [29] is used to calculate the AR model coefficients and errors. The time interval when the model error drops below a certain threshold is marked as the model error (ME) suspicious interval.

F. Integration of Multiple Detectors

We have presented four different detectors. These detectors will look for sudden changes. When there are two sudden changes, the curves will show the U-shape. When there are multiple sudden changes, the curves will have multiple peaks. The outputs of these detectors are intervals marked as MC suspicious, ARC suspicious, HC suspicious, and ME suspicious.

No single detector can handle all types of attacks effectively. In this case, we compare these detectors quantitatively based on their Receiver Operating Characteristics (ROC) curves [28] (in signal detection theory, a receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot of true

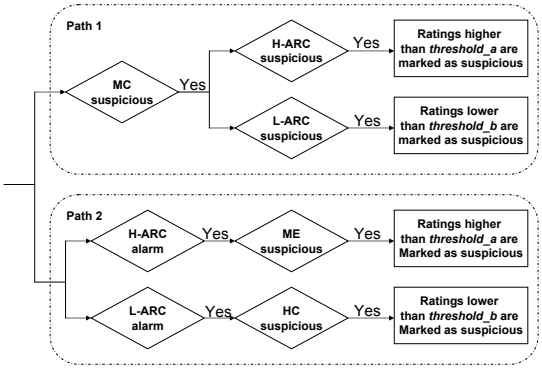

Fig. 2: Join Detection of Suspicious Ratings

positives vs. false positives. ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution). Due to space limitation, we cannot show the ROC curves of each detector in this paper. Based on the ROC analysis and experiments, we develop an empirical method to combine the proposed detectors, as illustrated in Figure 2.

There are two detection paths. Path 1 is used to detect strong attacks. The strong attack means the unfair ratings have large bias. If the MC indicator curve has a U-shape, and H-ARC or L-ARC indicator curve also has a U-shape, the corresponding high or low ratings inside the U-shape will be marked as suspicious. If for some reason, H-ARC (or L-ARC) indicator curve does not have such a U-shape, H-ARC (or L-ARC) alarm is issued. The alarm will be followed by the ME or HC detector. This is path 2. Path 2 detects suspicious intervals. Since there may be multiple attacks against one product, the ratings must go through both paths. Path 1 and Path 2 are in parallel.

G. Trust in Raters and Rating Aggregation

It is noted that we cannot perfectly differentiate unfair ratings and fair ratings in the suspicious intervals. Therefore, some fair ratings will be marked as suspicious. As a consequence, one cannot simply filter out all suspicious ratings. In this work, this suspicious rating information is used to calculate trust in raters, based on the beta-function trust model [30]. The calculation is described in Procedure 1.

Similar as in [9], we adopt the weighted average trust model to combine rating values from different raters. In particularly, let R denote the set of raters whose ratings are the inputs to the aggregation module. If rater $i \in R$, let r_i denote the rating from rater i and T_i denote the current trust value of rater i. In

Procedure 1 Computing Trust in Raters

For each rater i, initialize $S_i = 0$, and $F_i = 0$
for $k = 1 : K$ **do**
  for each rater i **do**
    Set $n_i = f_i = 0$,
    Considering all products being rated during time $\hat{t}(k-1)$ and $\hat{t}(k)$, determine:
      n_i: the number of ratings that is provided by rater i
      f_i: the number of ratings from rater i and being marked as suspicious
    calculate $F_i = F_i + f_i$ and $S_i = S_i + n_i - f_i$.
    calculate trust in rater i at time $\hat{t}(k)$ as: $(S_i + 1)/(S_i + F_i + 2)$.
  end for
end for

addition, each rater provides only one rating for one object and R_{ag} denotes the aggregated rating. Then,

$$R_{ag} = \frac{1}{\sum_{i:i\in R} max(T_i - 0.5, 0)} \sum_{i:i\in R} r_i \cdot max(T_i - 0.5, 0) \tag{7}$$

IV. Attack data analysis

A. Experiment Setup

We collected 251 valid submissions of unfair rating data in the rating challenge. Three observations are made. First, more than half of the submitted attacks were straightforward and did not exploit the features of the underlying defense mechanisms. These attacks are effective against majority-rule-based defense methods. Second, among the attacks that exploit the underlying defense mechanisms, many of them are complicated and unexpected. Third, according to a survey after the challenge, most successful participants either generated unfair rating manually or modified computer generated unfair rating data manually. As we expected, this data set covers a broad range of attack possibilities.

It is important to point out that the participants did not know the specific algorithms used for rating calculation. Through the analysis on real user attack data and interviews after the challenge, we have found that most of the participants simply tried many different ways to attack. A small portion of participants tried to figure out the underlying rating calculation algorithms and their attacks are specifically against the underlying rating algorithms. Thus, the collected attack data is diverse enough for achieving the main purpose of this work: identifying features that can be used to model attack, utilizing these features to

build attack models and attack generator.

In this section, we first analyze the attack data set from multiple angles. Then an attack data generator based on the analysis is presented in Section IV-E.

Three unfair rating detection systems are considered.

- P-scheme: the proposed system described in Section III. The parameters of the proposed system are as follows. The initial trust value of all raters is 0.5. The window size of the MC detector, H-ARC/L-ARC detectors, HC detector, and ME detector are 30 (days), 30 (days), 40 (ratings), and 40 (ratings), respectively. In the H-ARC and L-ARC, $threshold_a = 0.5m$ and $threshold_b = 0.5m + 0.5$, where m is the mean of the rating values in the time window.
- SA-scheme: using simple averaging for rating aggregation and not applying any unfair rating detection.
- BF-scheme: using the beta-function based filtering technique proposed in [2] to remove unfair ratings. Then, the trust value of rater i is calculated as $(S_i + 1)/(S_i + F_i + 2)$, where F_i is the number of ratings (from rater i) that have been removed, and $F_i + S_i$ is the total number of ratings provided by rater i. This is a representative majority-rule-based scheme.

The above defense schemes are tested against all attack data collected in the rating challenge. When using the P-scheme, the maximum MP value that the attackers achieve is about 1/3 of the maximum MP value when using the other two schemes. Thus, *from the defense points of view, the proposed P-scheme has significant advantage over the traditional majority-rule based schemes.* Since this paper only focuses on attack data analysis, we do not report this comparison results. We would like to point out that studying the attacks against the P-scheme is essential because of its significance as a new and powerful defense technique.

B. Unfair Rating Value Analysis

Each individual unfair rating is uniquely determined by the rating value and the time when this rating is provided. We examine the rating values in this subsection, and the time in the next subsection. Furthermore, the most effective unfair ratings might be correlated with the fair ratings. The issues related to correlation will be investigated in subsection IV-D.

When examining the rating values for one product, we define the difference between the mean of

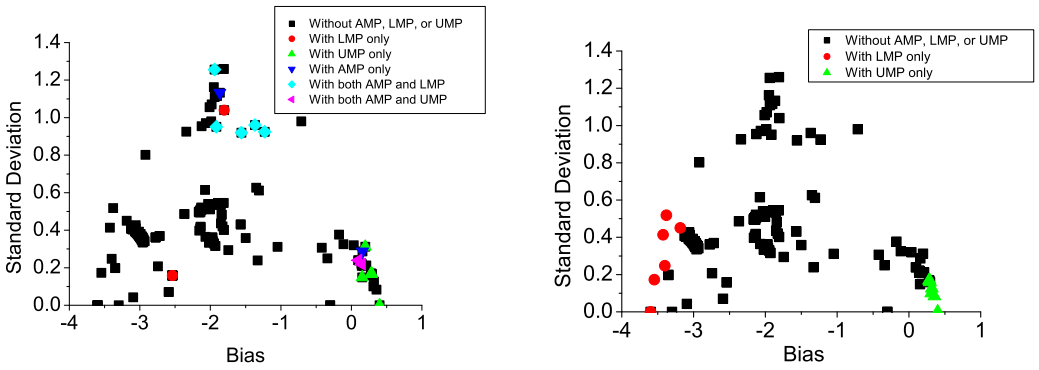

Fig. 3: (a) P-scheme for product 1 (b) BF-scheme for product 1

all unfair ratings and the mean of fair ratings as *bias*. When the bias value is positive, the purpose of the unfair ratings is to boost the final rating score. When the bias value is negative, the purpose is to downgrade the final rating score. The original rating value is between 0 and 5, and the mean of fair ratings is around 4 in the rating challenge. Thus, the bias is between -4 and 1. Besides bias, another important feature is the variance of the unfair rating values.

We are particularly interested in the unfair rating data that generate large MP values. The following data points are selected.

- We first compare the overall MP values. If one submission generates the top 10 overall MP values, this submission is marked as AMP. We only mark the AMP points with P-scheme because the SA-scheme and BF-scheme don't consider the correlation between different products.
- Then, for each product k, we compare the MP value gained from this product among all the submissions that have *negative* bias for product k. If one submission makes top 10, this submission is marked as UMP for product k.
- Finally, for each product k, we compare the MP value gained from this product among all the submissions that have *positive* bias for product k. If one submission makes top 10, this submission is marked as LMP for product k.

In Figure 3, the horizontal axis is the unfair rating bias for the first product and the vertical axis is the standard derivation of the unfair rating values. Each dot represents one submission. We use different colors to differentiate the different types.

- Black: without AMP, LMP, or UMP marks
- Blue: with AMP mark only

- Red: with LMP mark only
- Green: with UMP mark only
- Cyan: with both AMP and LMP marks
- Pink: with both AMP and UMP marks

Due to space limitation, we only show the results for product 1. The results for other products are similar. In Figure 3, the MP values are calculated with P-scheme and BF-scheme. Since the SA-scheme's performance is very similar with BF-scheme, we do not show the performance. By investigating the relationship between bias, variance, and MP values, we made an important observation.

- For each unfair rating detection algorithm, the submissions that generate large MP values mostly locate in a specific region on the variance-bias plot. This is especially obvious when the bias is negative (i.e. downgrading).

For negative bias, we can roughly divide the variance-bias plot into three regions: (R1) large bias, small to medium variance; (R2) medium bias, small to medium variance; and (R3) medium bias, medium to large variance. When the P-scheme is used, the submissions with large MP values are concentrated in region R3. When the other two schemes are used, the submissions with large MP values are concentrated in region R1. This directly leads to a better understanding on different unfair rating detection schemes.

- The signal-based unfair rating detection scheme can detect unfair ratings with small-to-medium variance, but cannot handle the unfair ratings with large variance very well. That is, to cheat the signal-based scheme, the attacker should insert unfair ratings with large variance and medium bias. The intuitive explanation is that the large variance can weaken the features presented in the unfair ratings and therefore make the detection based on signal features less effective.
- The BF-scheme is not very effective. Comparing Figure 3 (a) and (b), we can see that the unfair ratings with large bias make the largest MP. As long as those ratings are not very few, the BF-scheme cannot judge whether they are far from the majority's opinions. This is true for the most of the majority-rule based methods.

The above observations are made for the downgrading attack (i.e. negative bias). In the rating challenge, we have observed that the boosting attack (i.e. positive bias) is not as effective as the downgrading attack in terms of generating large MP values. This is because the mean of the fair ratings is high (which is true

for most popular products on commercial websites) and there is no much room to further boost the rating values. As a consequence, the variance-bias plot for positive bias does not have a high 'resolution' such that the different regions can be clearly identified. In this paper, we focus on the downgrading attack, and will examine the details of the boosting attack in the future work.

From above discussion, we can see that *bias and variance are two important features directly related to the strength of the attack*. This enables us to design a heuristic algorithm to find the best region on the variance-bias plot to generate unfair ratings for different algorithms from the attacker's points of view. Identifying these regions is critical for the design and evaluation of defense algorithms.

The heuristic algorithm is described in Procedure 2.

Procedure 2 Heuristic Unfair Rating Value Set Generator

1: Set $interested - area$ as the entire area on the variance-bias plot
2: Set $Flag = true$.
3: **while** $Flag = true$ **do**
4: Divide the $interested - area$ into N subareas
5: **for** each subarea **do**
6: Randomly generate m set of unfair rating data using the bias and variance values represented by the center point of the subarea,
7: Test the rating aggregation systems and determine the maximum MP value resulting from the m set of unfair ratings.
8: **end for**
9: Set the $interested - area$ as the subarea with the largest MP.
10: **if** the $interested - area$ is smaller than a threshold **then**
11: Flag=false
12: **end if**
13: **end while**
14: Output the $interested - area$

In Figure 4, it demonstrates the process of Procedure 2. The initial $interested - area$ ranges from 0-4(bias) and 0-2(standard deviation), and $N = 4$, $m = 10$. After 4 rounds, it outputs the area with the center point of (-2.325,1.56). The most significant result is that the MP value gained from the generator is larger than any of the submission during the challenge, which demonstrates that this heuristic generator can generate more powerful attack automatically.

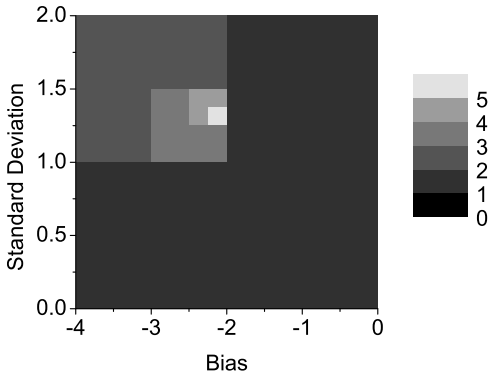

Fig. 4: Optimum region searching

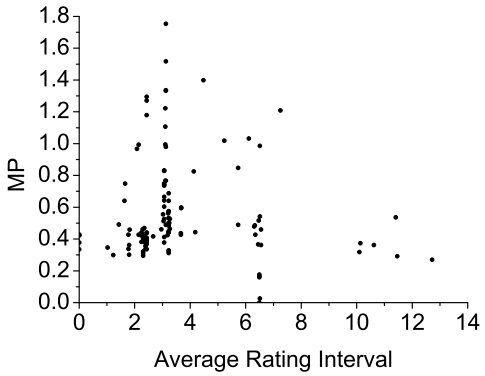

Fig. 5: Time Analysis for unfair ratings of product 1

C. Time Domain Analysis

Attack duration, which is defined as the time period when dishonest raters provide their ratings, is an important feature of attack data in the time domain. In the rating challenge, there is a large variation in the attack duration, ranging from 10 days to the entire rating duration. In this subsection, we examine the average *unfair rating interval*, defined as the total number of unfair ratings divided by the attack duration. Given the same number of unfair ratings, longer is the attack duration, larger is the unfair rating inter arrival time.

In Figure 5, the horizontal axis is average rating interval, and the vertical axis is the MP value for product 1 when using the P-scheme. Each dot in the plot represents one unfair rating submission. Obviously, with the same inter arrival time, different submissions yield different MP values because the specific attack methods are different. However, it is seen that there exist the best arrival rate, which can produce the largest MP values. In Figure 5, this best interval is approximately 3 days.

It is noted that the specific value of the best interval is affected by two factors: the way to calculate

the MP value and the unfair rating detection algorithm. In this experiment, the MP value is calculated monthly and the top two MP values are counted. Therefore, without using any unfair rating detection algorithm, the attacker should distribute all 50 unfair ratings within a two month period of time, which makes the best inter arrival time smaller than 1.2. When using the signal-based detection algorithm, the attack with high arrival rate can be easily detected, and the attack with very large interval does not have much impact on the MP values. This is why the moderate arrival rate leads to the highest MP value.

In this paper, we only conducted the simple analysis along the time domain. Even this simple analysis yields an interesting result. That is, the signal-based defense methods response differently to different unfair rating arrival rate. As a consequence, *in the evaluation of unfair rating detection algorithms, it is necessary to adjust the unfair rating arrival rate in a fine scale.*

D. Rating correlation

In the current literature, the unfair ratings are always assumed to be independent of each other and also independent of fair ratings. Is this assumption valid?

To answer this question, we calculate the correlation between fair ratings and the unfair ratings. Here, the correlation means the relationship between the fair ratings (background ratings) and the unfair ratings (in one attack). The unfair ratings are from different user IDs, and these user IDs are under the control of the same attacker. From the real user attack data, we didn't see obvious correlation between unfair ratings and fair ratings. We believe that *the assumption of no correlation between unfair ratings and fair ratings is valid for the current attackers.*

However, can the correlation enhance an attack? We conduct the following heuristic experiment to explore this question.

- We choose the real user unfair rating data that generate top 10 MP values.
- We change the order in which the unfair ratings are given to create correlations between unfair ratings and fair ratings with Procedure 3.
- We also change the order randomly 5 times to compare the results.
- With this method, we generate 60 sets of new unfair rating data.
- We compare the generated unfair rating data with the original unfair rating data from real users, in terms of the MP values.

Procedure 3 Heuristic Correlation Algorithm

1: Put all the rating values in rating value set
2: Put all the rating time in rating time set.
3: **while** rating time set is not empty **do**
4: Set $MinT$ as the minimum time in rating time set
5: Set $NearV$ as the fair rating value whose rating time is just before $MinT$
6: Set $MaxV$ as the rating value that has the maximum difference with $NearV$ in rating value set
7: Match $MinT$ and $MaxV$ together
8: Remove $MinT$ out of rating time set
9: Remove $MaxV$ out of rating value set
10: **end while**

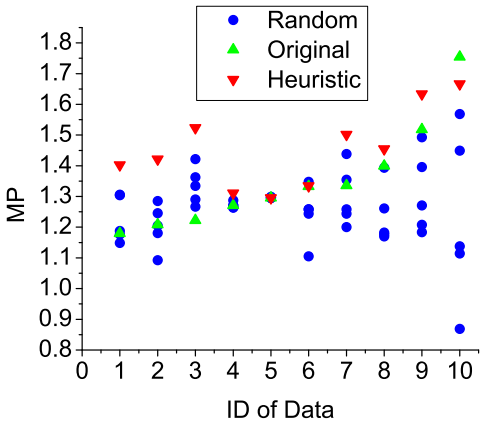

Fig. 6: Comparison of different order strategies

The comparison results are shown in Figure 6. In most of the cases, the generated unfair rating data yield higher MP than the original rating data under the P-scheme, which indicates the *correlation* can improve attack strength. Although the current attackers have not correlated unfair ratings with fair ratings, we believe this is a potential threat in the future. Therefore, it is necessary to evaluate the defense algorithms against unfair ratings in correlation.

E. Attack Generator

Although the attack data from real human users can accurately evaluate the rating system, the data collection process is time-consuming and costly. When the system design is in its early stage, simulation is an effective way to evaluate and improve the system. Therefore, a good attack generator can greatly help the rating aggregation system design.

Based on the investigation on real user attack data, we get the necessary features to describe attacks against rating systems. These features include bias, variance, arrival rate, and correlation associated with unfair ratings. To assist the evaluation of current and future rating aggregation systems, we design an attack generator as illustrated in Figure 7. To use the proposed attack generator, the users need to provide (1) honest rating data, (2) rough range of attack parameters, and (3) the algorithm for rating aggregation. The basic attack parameters include mean, variance, and arrival rate of unfair ratings. Then, the generator will provide a sequence of unfair ratings that represent strong attacks against the honest ratings in the specified rating aggregation system.

The proposed attack generator is implemented in Matlab and gets a performance evaluation against the P-scheme. To compare the attacks from real users, we do the experiment like the participants in the rating challenge. Specifically, we downgrade the product 1, 3 and boost product 2, 4, setting the average time interval as 2 , 3 and 4 days, like the winner participants did in the rating challenge. We generate the rating time based on Poisson distribution. Figure 7 gives the MP results of the attacks from the generator. Based on the process of Procedure 2, the generator can give more and more powerful attacks by heuristically learning from the previous attack effects. Comparing Figure 5 and Figure 7, we see that the generator can give the similar attack performance as the real users did. However, the attacks from the generator cannot achieve the strongest attack (MP=1.8) from a real user. We investigate this strongest attack and find out that the participate used a complicated strategy to get the best performance. First, its rating time does not follow Poisson distribution; second, not 50 dishonest users all attack the 4 products. Instead, 20 out of dishonest users did not join the boost attack, which made the P-scheme not detect them very well. Although the generator does good jobs in most of the cases, it cannot completely replace the real users' test, which makes our rating challenge more meaningful.

V. Conclusion

In this paper, we exploit the features in unfair ratings against rating aggregation systems. Different from all previous efforts, the exploration is based on attack data collected from real human users through a rating challenge. The rating challenge is carefully designed such that the data collected can represent a broad range of attacks against different rating aggregation systems. For the rating challenge, we design a new signal-based unfair rating detection system, which not only outperforms the existing schemes but

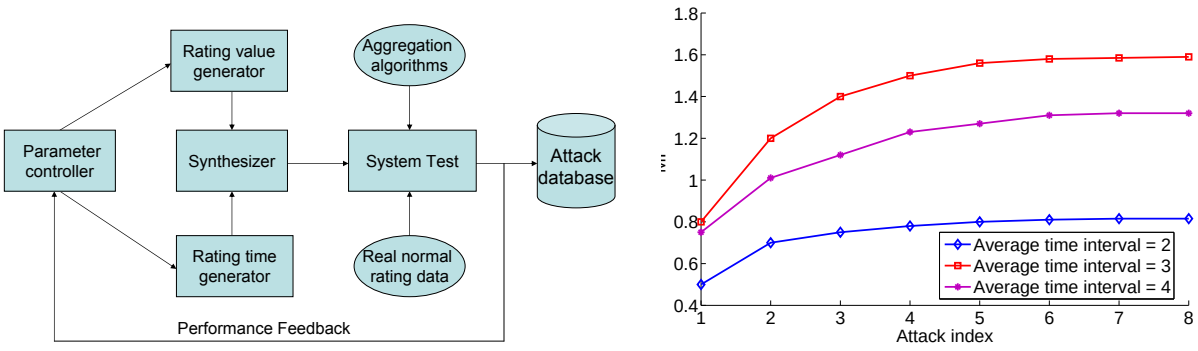

Fig. 7: Attack generator and performance

also encourages creative attacks from the participants.

Based on the analysis of real attack data, we have discovered important features in unfair ratings. For example, the bias and variance greatly affects the strength of attacks; there exists an unfair rating arrival rate that maximizes the attack power; correlation between unfair ratings and fair ratings are not presented in current attacks but can improve the attack power, etc. Furthermore, we build attack models and develop an unfair rating generator. The models and generators developed in this paper can be directly used to test current rating aggregation systems, as well as to assist the design of future rating systems.

REFERENCES

[1] Z. Liang and W. Shi, "Analysis of ratings on trust inference in open environments," *Elsevier Performance Evaluation*, vol. 65, no. 2, pp. 99–128, 2008.

[2] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in Bayesian reputation systems," in *Proc. 7th Int. Workshop on Trust in Agent Societies*, 2004.

[3] J. Zhang and R. Cohen, "Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings," in *Proceedings of the 8th international conference on Electronic commerce*, 2006.

[4] Q. Zhang and T. Yu, "On the modeling of honest players in reputation systems," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2008.

[5] C. Dellarocas, "Strategic manipulation of internet opinion forums: Implications for consumers and firms," *Management Science*, October 2006.

[6] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000.

[7] M. Chen and J.P. Singh, "Computing and using reputations for internet ratings," in *Proceedings of the 3rd ACM conference on Electronic Commerce*, 2001.

[8] J. Weng, C. Miao, and A. Goh, "An entropy-based approach to protecting rating systems from unfair testimonies," *IEICE TRANSACTIONS on Information and Systems*, vol. E89-D, no. 9, pp. 2502–2511, September 2006.

[9] Y. Yang, Y. Sun, J. Ren, and Q. Yang, "Building trust in online rating systems through signal modeling," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2007.

[10] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," *Proceedings of the tenth international conference on Information and knowledge management*, pp. 310–317, 2001.

[11] L.-H. Vu and K. Aberer, "A Probabilistic Framework for Decentralized Management of Trust and Quality," *Eleventh International Workshop on Cooperative Information Agents (CIA 2007)*, 2007.

[12] Z. Despotovic and K. Aberer, "A Probabilistic Approach to Predict Peers' Performance in P2P Networks," *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004, Erfurt, Germany, September 27-29, 2004; Proceedings*, 2004.

[13] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," *The Icfain Journal of Management Research*, vol. 4, no. 2, pp. 48–64, 2005.

[14] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," *Proceedings of the 2nd ACM conference on Electronic commerce*, pp. 150–157, 2000.

[15] Z. Despotovic and K. Aberer, "P2P reputation management: Probabilistic estimation vs. social networks," *Computer Networks*, vol. 50, no. 4, pp. 485–500, 2006.

[16] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-topeer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, May 2007.

[17] M. Maheswaran and H. Tang, "Towards a gravity-based trust model for social networking systems," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2007.

[18] Y. Wang and J. Vassileva, "A review on trust and reputation for web service selection," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2007.

[19] H. Zhao and X. Li, "H-trust: A robust and lightweight group reputation system for p2p desktop grid," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2008.

[20] A. Gutowska and K. Buckley, "Computing reputation metric in multi-agent e-commerce reputation system," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2008.

[21] H. Wu, H. Chen, and C. Gao, "A trust management model for p2p file sharing system," in *Proceedings of IEEE ICDCS Workshop on Trust and Reputation Management*, 2008.

[22] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, July 2004.

[23] K. Fujimura and T. Nishihara, "Reputation rating system based on past behavior of evaluators," in *Proceedings of the 4th ACM conference on Electronic commerce*, 2003.

[24] "Netflix prize dataset," www.netflixprize.com/download.

[25] University of Rhode Island, "Etan rating challenge," www.etanlab.com/rating.

[26] D. H. McKnight and N. L. Chervany, "The meanings of trust," MISRC Working Paper Series, Technical Report 94-04, arlson School of Management, University of Minnesota, 1996.

[27] Y. Sun and Y. Yang, "Trust establishment in distributed networks: Analysis and modeling," in *Proceedings of IEEE ICC'07*, 2007.

[28] Steven M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*, Prentice Hall, 1998.

[29] M.H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley and Sons, 1996.

[30] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, June 2002.