ReviewSec: A Tool for Online Review Analysis

Yongbo Zeng, Yihai Zhu, Yan (Lindsay) Sun

University of Rhode Island, Kingston, RI
Email: {yongbozeng, yhzhu, yansun}@ele.uri.edu

***Abstract*—Online review plays an important role when people are making decisions to purchase a product/service. It is shown that the sellers can benefit from boosting the reviews of their products/services, or downgrading the reviews of their competitors.**

Dishonest behavior on reviews can seriously affect both buyers and sellers. In this work, we propose an algorithm that contains a two-step analysis to detect whether a product's reviews have been manipulated. The first step is called consistency analysis, which detects the variation in rating values. In the second step, we introduce a novel angle to detect dishonest reviews, called Equal Rating Opportunity (ERO) principle. We propose the ERO analysis using the ANOVA method. Furthermore, we develop a web-based system, referred to as ReviewSec, to conduct real-time on-demand review manipulation detection. The ReviewSec system includes three modules: 1) crawler, which download reviews data from e-commerce website; 2) detector, consisting of the consistency analysis and ERO analysis; and 3) web-based interface. We believe that with the assistance of the ReviewSec system, online shoppers can understand product reviews in a better way and thereby reduce the risk of being misled by untruthful reviews.

I. Introduction

Online reviews are posted by people who have experience of using the products/services. A review usually consists of a rating score and a piece of comment describing the feedback of using the product/service. More and more people are relying on online reviews when evaluating the quality of products, hotels, restaurants, and even vacation packages. The reason includes that the description of the product/service may not be accurate, and that the other clues such as touching and trying out may not be available. The online review systems, also referred to as *online reputation systems*, allow users to post reviews for products/services, and aggregate these reviews to assign each product/service with a reputation score that indicates the quality (e.g. number of stars in Amazon). Online reputation systems can help people evaluate the quality of products/services before transactions, and hence greatly reduce the risks of people's buying behaviors.

However, not all the reviews are honest. Driven by the huge profits of online markets [1], attacks that attempt to mislead users' buying decisions through dishonest reviews are gaining popularity. Sellers at the online marketplace boost their reputation by trading with collaborators [2]. Firms post biased reviews to praise their own products or bad-mouth their competitors' products [3]. Review manipulation can overly inflate or deflate products' reputation scores, crash users' confidence in online reputation systems, eventually undermine reputation-centric online businesses and lead to economic loss. Particularly, there are some situations in which the review manipulation is even more damaging. For example, in *Black Friday*, people may have to make a rush decision because the 'unusual' discount will expire quickly. Another example is that the hotels and restaurants cannot be returned after people using them.

In the literature, researchers protect reputation systems from several angles, for example 1) increasing the cost of acquiring multiple user IDs [4], 2) endogenous discounting of dishonest reviews by analyzing the statistic features of the reviews [5], 3) exogenous discounting of dishonest ratings by introducing reputation evaluation of users [5]–[7], and 4) studying correlation among users to detect dishonest reviews [8], [9]. In this paper, we argue that review manipulation detection should also be conducted on-demand, for individual products/services upon the request from users, because the e-commerce sites may not have strong incentive to detect review manipulations. On the other hand, the detection should have low implementation cost and better accessibility so that third-parties can provide independent opinions.

The development of on-demand review manipulation detection system is challenging. *First*, on-demand service requires real-time response, which means the amount of data to be acquired and processed should be small enough to guarantee a reasonable latency. *Second*, the result presented to the users must be easy to understand and help users to make informative decisions quickly. *Third*, the ground-truth is unavailable when evaluating the detection system. There are mainly two methods proposed by researchers to evaluate detection schemes from two angles. One is adding artificial dishonest reviews, and the other is using expert opinions.

In this work, we propose an algorithm that utilizes the rating change interval detection method and the analysis of variance method to detect whether a product's reviews are manipulated. The rating change interval detection, also referred to as consistency analysis, aims to analyze how the rating values change with time. We introduce the **Equal Rating Opportunity (ERO)** principle and propose to use ERO analysis to examine the satisfaction of the ERO principle. In order to address the first challenge, we revise the adopted consistency analysis by using random sampling. We also develop a web-based system, referred to as ReviewSec, to provide a user-friendly interface for individual users, which could address the second challenge. We also perform a few case studies to demonstrate the proposed algorithm and the ReviewSec system. In the

future, real users could be involved to evaluate the system.

The rest of this paper is organized as follows. Related work is discussed in Section II. The detection algorithm is presented in Section III, followed by the ReviewSec system and case studies in Section IV.

II. Related Work

In order to protect online reputation systems, researchers propose many protection schemes that can be roughly put into 4 categories. The **first** category is increasing the cost of acquiring multiple user IDs by binding user IDs with IP addresses [4]. The **second** category is endogenous discounting of dishonest reviews [5]. Dishonest ratings are directly differentiated from normal ratings based on the statistic features of the rating values. In a Beta-function based approach [10], a user is determined as a malicious user if the estimated reputation of the product rated by him/her lies outside q and $1-q$ quantile of his/her underlying rating distribution. An entropy based approach is proposed in [11]. The **third** category is exogenous discounting of dishonest ratings. Users are assigned trust scores based on their review history, and the quality of their reviews are discounted according to their trust scores. In [7], a user's trust is obtained by cumulating his/her neighbors' beliefs through belief theory. The **fourth** category is studying correlation among users to detect dishonest ratings [8], [9]. The proposed scheme has both category 2 and category 4 features, and the detection algorithm is from a new angle.

Many research results did not turn into practical systems. This is probably because of the potential liability concerns of major e-commerce companies, as well as the gap between research and practical constraints. Currently, there are only a few existing online systems providing review analysis services. For example, there is a website called "ReviewPro" [12], whose major business is to provide professional suggestions to hotel owners. By analyzing the customers' reviews on a hotel, ReviewPro can provide analytical reports with "strategies" to climb TripAdvisor rankings and earn 5-star reviews. Another practical system is "TRUSTYOU" [13], which provides review analysis services on hotels. For hotel owners, it provides services to market the reputations and increase businesses. For individual users, it provides services to analyze the hotel's quality, by summarizing online reviews and generating a trust score for the hotel. What we propose in this work is different from these existing services. First, our work focus on detecting review manipulation, instead of finding patterns for reputation promotion purpose. Second, our work can provide on-demand real time service, whereas ReviewPro and TRUSTYOU can only offer analysis of a pre-determined list of hotels.

III. Review Manipulation Detection Algorithm

A. Overview

The proposed ReviewSec system has mainly three modules.

1) The first module is the crawler. It downloads web pages from the e-commerce sites, parses review information and stores the information in the database.

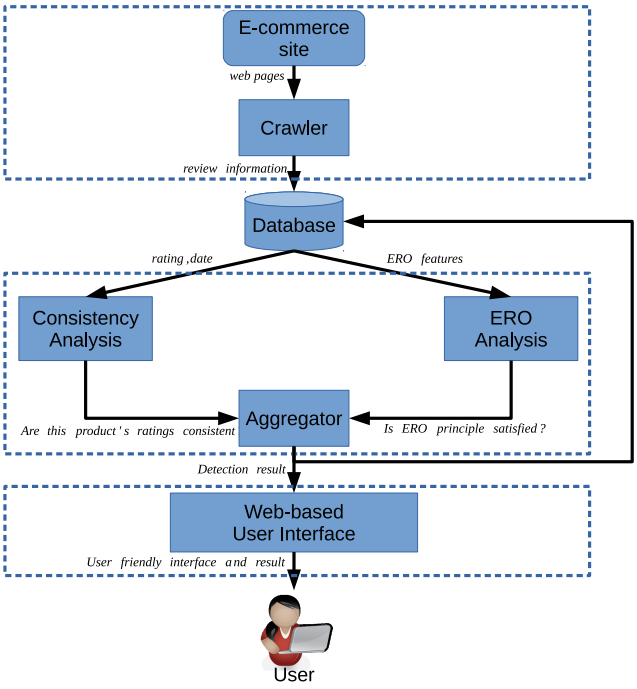

Fig. 1: The ReviewSec system overview.

2) The second module is the review manipulation detection, which consists of two analysis algorithms, the consistency analysis and the ERO analysis.
3) The third module is the web-based user interface.

Figure 1 shows the diagram of the proposed ReviewSec system.

In this section, we present the second module, review manipulation detection. Particularly, the consistency analysis and ERO analysis are discussed in detail. The first module and the third module are discussed in Section IV.

B. Consistency Analysis

Most of the e-commerce sites maintain an average rating score (e.g. number of stars in Amazon) for each product to provide users an overview of the product's quality. This rating score is an important filtering criterion when users select products to view and to purchase. Therefore, the rating score is highly likely to be the target of dishonest reviews. We argue that if the rating scores of a product are inconsistent with time, it may indicate the possibility of review manipulation. Note that in order for a manipulation to be effective, it must cause large enough change in the average rating score.

In the literature, there are several approaches to detect the variation of average ratings. In this work, we adopt the one proposed in [14] called CUSUM.

1) Detection Function: The ratings of a product is organized according to the time when the review was posted. Let $x[n]$ denote the n^{th} rating of a product. Here, the index of the rating is the order of the arrival of the review. For example, $x[1]$ is the earliest rating of the product, $x[2]$ is second rating received, etc. The true average rating of the product is μ_0,

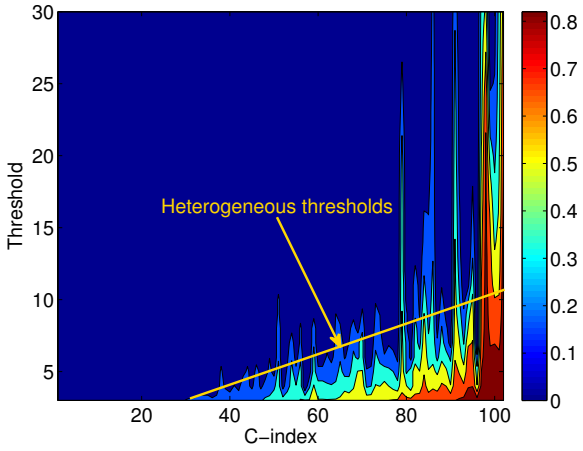

Fig. 2: Detection Threshold Selection.

and the change to be detected when the variation of average rating exceeds a range ν. In other words, if $\bar{x}_n > \mu_0 + \nu$ or $\bar{x}_n < \mu_o - \nu$, the variation of average rating is observed, where $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x[i]$ is the average rating. The detection functions are:

$$\begin{cases} g_n^+ = \max(g_{n-1}^+ + x[n] - \mu_0 - \nu/2,\ 0) \\ g_n^- = \max(g_{n-1}^- - x[n] + \mu_0 - \nu/2,\ 0) \end{cases} \tag{1}$$

where g_n^+ indicates the positive changes (upgrading), g_n^- indicates the negative changes (downgrading), and $g_0^+ = 0$ and $g_0^- = 0$ for initialization.

Rating inconsistency is observed when g_n^+ or g_n^- exceeds a threshold $\bar{h}$, and the inconsistency interval is from time t_a to time t_b.

$$\begin{cases} t_a = \arg\min_n \{g_n^+ \geq \bar{h} \cup g_n^- \geq \bar{h}\} \\ t_b = t_a + \arg\min_{\Delta t} \{g_{t_a+\Delta t}^+ < \bar{h} \cap g_{t_a+\Delta t}^- < \bar{h}\} \end{cases} \tag{2}$$

Note that there might be multiple inconsistency intervals observed for one product.

2) Detection Threshold Selection: As we discussed in the previous subsection, rating inconsistency is observed when the detection function exceeds the threshold $\bar{h}$. It is argued that a uniform threshold for all products is not applicable, and heterogeneous thresholds should be used [14]. Next, we briefly describe the CvT features, abbreviation for Change Interval versus Threshold, presented in [14] as background, and then present the modified procedure used in ReviewSec.

From (2), we can easily observe that the change period $[t_a, t_b]$ depends on the threshold $\bar{h}$. Percentage of Change Interval (PCI) is used, which is a function of $\bar{h}$:

$$PCI(\bar{h}) = \frac{total\ \ length\ \ of\ \ all\ \ detected\ \ change\ \ periods}{length\ \ of\ \ the\ \ rating\ \ series} \tag{3}$$

In this paper, when we discuss specific product, we also use $PCI(\bar{h}, p)$ to denote the PCI for product p when threshold is $\bar{h}$.

CvT feature is used to represent how PCI value changes along with the threshold. By visualizing the CvT feature, we can get the threshold selection metrics. For example, Figure 2 is a visualization figure of CvT from real data, which contains ratings for 114 products from Amazon. One integer in the x-axis indicates one product, the y-axis indicates the threshold $\bar{h}$, and the color indicate the PCI value. The products are reorganized on the x-axis, which we discuss later.

It is seen that the PCI value drops as $\bar{h}$ increases, but the PCI of some products drops much slower than that of other products. These products are corresponding to the spikes on the CvT figure. In [14], all data is used to generate the CvT figure. In ReviewSec, to reduce the data size, we randomly select N products, denoted by $p_1, p_2, \ldots, p_N$, to generate CvT as follows.

1) A starting threshold $\bar{h}_0$ is selected, and $PCI(\bar{h}_0, p_i)$ for $i = 1, 2, \ldots, N$ is calculated.
2) Reorganize those N products in the ascending order of $PCI(\bar{h}_0, p_i)$ and use the order number as the index of each product, which is referred to as *C-index*. For example, if p_5 has the 10^{th} lowest $PCI(\bar{h}_0)$, the C-index for p_5 is 10.
3) For each product, calculate the PCI values for different thresholds.
4) Then the contours of the PCI values are generated using *contour* function in Matlab.

Based on the contour figure, heterogeneous thresholds are selected. A line is fitted to be parallel to the contours, according to the 1^{st} order regression model. Let $\bar{h} = a_0 + a_1 c$ represent the line, where a_0 and a_1 are the fitted coefficients and c is the C-index.

For a given product p, the detection threshold is selected as follows. 1) Calculate $PCI(\bar{h}_0, p)$ and find the closest C-index by comparing $PCI(\bar{h}_0, p)$ with $PCI(\bar{h}_0, p_i)$ $i = 1, 2, \ldots, N$. 2) Plug the C-index into the fitted line to calculate the threshold, denoted by $\bar{h}_d$. If the threshold is less than $\bar{h}_0$, the detection threshold is $\bar{h}_0$.

3) Consistency Analysis: After the calculation of detection functions (i.e. g_n^+ and g_n^-) and threshold selection, $PCI(\bar{h}_d)$ is calculated and used to represents the consistency level of the product. Obviously, $PCI(\bar{h}_d) = 0$ indicates ideal consistency, and $PCI(\bar{h}_d) > 0$ means the ratings are volatile. In the web-based system, we convert the PCI value to the displayed value, which is $1 - PCI(\bar{h}_d)$.

C. Equal Rating Opportunity Analysis

1) Equal Rating Opportunity Principle: The consistency detection, which is based on the statistics of ratings, can only be used to find products that are suspected to be under review manipulation attack, but is lack of the capability to accurately detect such manipulation. This is because the average rating can change without any manipulation. For example, when a restaurant changes the chief, a seller changes his/her attitude toward consumer complaints, and the manufactory fixes a defect of the product, the ratings for the restaurant/seller/product could change. The ratings is also related to price. Consumers

tend to be more tolerant if they purchase deeply discounted products. If the price changes dramatically, the ratings may change. Therefore, after the consistency analysis gives us a set of suspicious products, we must apply a more informative analysis to confirm the review manipulation.

We are inspired by the *Equal Employment Opportunity Policy*, adopted by many employers. One example of such policy statement is as follows.

"All employment decisions at the company are based on business needs, job requirements and individual qualifications, without regard to race, color, religion or belief, national, social or ethnic origin, sex (including pregnancy), age, physical, mental or sensory disability, HIV status, $\cdots$."

We introduce an **Equal Rating Opportunity (ERO) Principle**, as follow.

ERO Principle. *"The normal rating values should be primarily based on the quality of the product or services, without regard to whether the review is posted on weekdays or weekend, posted during daytime or night time, long or short, from reviewers on east coast or west coast $\cdots$."*

Review features that apply to ERO principle are referred to as **ERO features**. Of course, this ERO Principle is not very strict. For example, it is possible that a product is more favorable in east coast than in west coast. However, for general products, the factors mentioned in the above statement, besides the quality, should play marginal roles in the rating values.

2) ERO Analysis: Many features of the reviews can be used as ERO features, such as "day of the week", "time in a day", "review text length", "geographical location", etc. Let $Y[n] = \{y_1[n], y_2[n], \ldots, y_k[n]\}$ denote the features of a review, where n is the time index (similar as the variable n in $x[n]$), and $y_1, y_2, \ldots, y_k$ are different features. For example, y_1 is "day of the week", and y_2 is "review text length".

Based on the ERO principle, the "day of the week" (i.e y_1) feature of the 1-star reviews and the "day of the week" feature of 5-star reviews should be similar. If not, ERO principle is not satisfied and we argue that the manipulation of reviews is detected.

In this work, we employ the analysis of variance (ANOVA) method to check whether EOR principle is satisfied upon each given ERO feature.

3) ANOVA and Correlation Analysis: Analysis of variance (ANOVA) is used to test whether or not the null-hypothesis can be rejected at a certain alpha (confidence) level. The *null-hypothesis* indicates that there is no significant difference between the group means, while the *alternative hypothesis* indicates that there is a difference between at least two of the group means.

ANOVA has been used in the analysis of experimental data. For example, a manufactory uses ANOVA to test whether the products of multiple machines are statistically identical in terms of mean value.

In this work, we formulate the ERO analysis by first introducing the hypotheses to be tested

$$\begin{cases} \mathcal{H}_0 : ERO\ feature\ means\ are\ identical\ for\ ratings \\ \mathcal{H}_1 : ERO\ feature\ means\ are\ not\ identical\ for\ ratings \end{cases} \tag{4}$$

When $\mathcal{H}_0$ is rejected, it follows that the conclusion is $\mathcal{H}_1$, which means ERO principle is not satisfied. In other words, the influence of the ERO features on the rating values is non-negligible.

TABLE I: ERO analysis example

Group pair	Conf. level for product I	Conf. level for product II
1 vs. 2	0.9716	0.9966
1 vs. 3	0.0351	0.9989
1 vs. 4	0.4950	0.9804
1 vs. 5	0.4629	*0.9513*
2 vs. 3	*0.0090*	0.9995
2 vs. 4	0.1770	1.0000
2 vs. 5	0.1342	1.0000
3 vs. 4	0.3652	0.9963
3 vs. 5	0.1791	0.9758
4 vs. 5	0.9984	0.9987
ERO satisfaction	**0.0090**	**0.9513**

There are different variants of ANOVA. In the ReviewSec system, the one we adopt is *One-way ANOVA for multiple groups and single factor*. ERO features are analyzed independently. For a given ERO feature, we use Matlab function *anova1* taking the feature as observed sample and the rating as group tag. Another Matlab function *multcompare* is used to calculate the confidence level of hypothesis $\mathcal{H}_0$ for each pair of group, denoted by $\alpha_{i,j}$, where i means i^{th} feature and j means j^{th} pair. The group pair includes all combinations of groups, for example rating=1 vs. rating=2, rating=1 vs. rating=3, etc. The ERO satisfaction upon i^{th} feature, i.e. y_i, is

$$S_i = \min_j \alpha_{i,j} \tag{5}$$

$S_i = 1$ means the ERO principle is absolutely satisfied upon ERO feature y_i, while $S_i = 0$ means the ERO principle is not satisfied at all. We demonstrate the process of ERO analysis using an example.

The ERO feature is "day of the week". Rating can have a value in 1,2,...,5. Observed sample is the "day of the week", which can have a value in Monday, Tuesday, ..., Sunday. There are 10 pairs of groups, as listed in the 1^{st} column in Table I. The confidence level for product I and product II are listed in column 2 and 3 respectively in Table I. The ERO satisfactions are listed in the last row. Obviously, product II satisfies the ERO principle for this ERO feature, while product I does not.

IV. ReviewSec System and Results

In this section, we first describe the other two modules of the proposed ReviewSec system, the crawler and the web-based user interface. Then we present a case study on real data to demonstrate the proposed ReviewSec system.

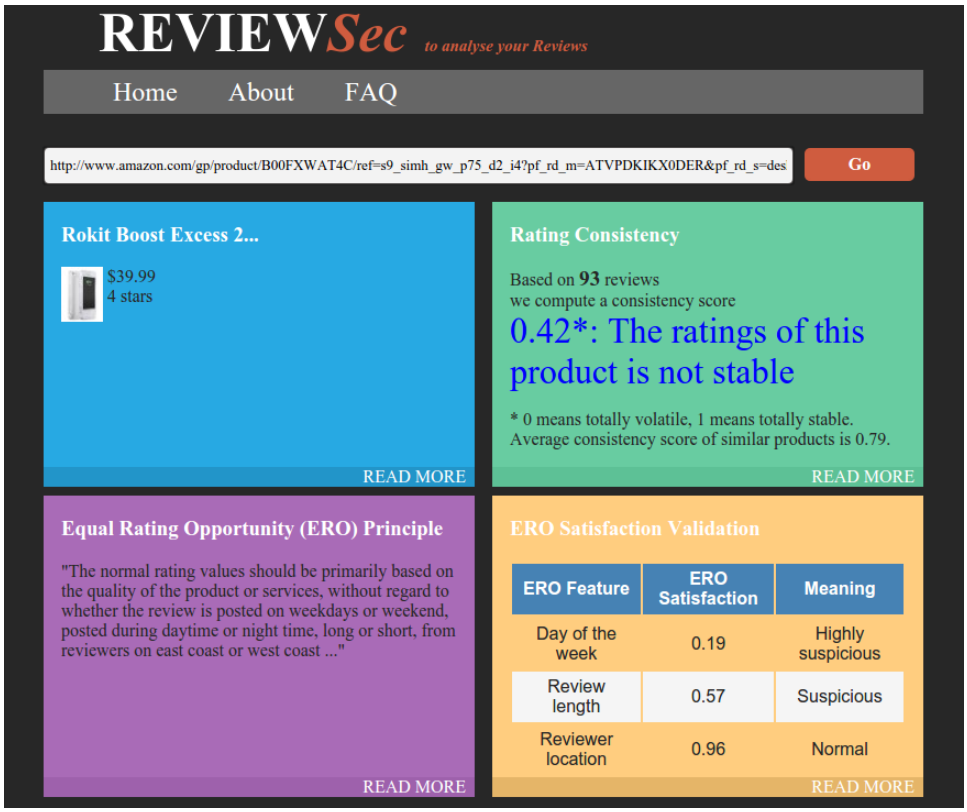

Fig. 3: An example of the system response.

TABLE II: Data crawler

Product	Review	Reviewer
ASIN	Rating value	Customer ID
Average rating	# of helpful votes	Reviewer ranking
Category	# of total votes	# of helpful received votes
	Review date	# of total received votes
	Review text	# of reviews
	Verified review or not	# of verified reviews

A. Data to be crawled

In the ReviewSec system, we develop a data crawler to download data from Amazon. Generally, the crawler can download web pages from Amazon, parse the desired information and store it in the database. There are 3 types of crawlers in the system.

1) *Product crawler* crawls the information of a given product. In Amazon, products are indexed by the Amazon Standard Identification Number (ASIN). ASIN is used as the product ID in our system.
2) *Review crawler* crawls the reviews of the given product.
3) *Reviewer crawler* crawls the profile and history data of the given customer. In our system, we use the customer ID in Amazon as our user ID.

Table II shows the detail information we obtain.

ReviewSec is an on-demand review analysis tool. When a ReviewSec user requests the review analysis service (i.e. copy the URL of an Amazon product to ReviewSec interface and click "Go"), we will first check whether the database contains up-to-date information of this product. If yes, the previous analytical results can be directly sent to the web-based user interface module. If not, the crawlers are called to obtain data from Amazon and the newly obtained data are sent to the database and passed to the review manipulation detection module.

B. ReviewSec System

The outcome of the review manipulation detection module is a set of numerical values, as illustrated in Table I. Instead of directly presenting these results, we designed a web-based user interface, as shown in Figure 3.

Assume Alice finds a product in Amazon that she is interested in, but she does not know whether she can trust all these reviews. Alice can copy the URL of this product from her browser, paste it in the input text box in ReviewSec system, and click the red "Go" button. ReviewSec will crawl necessary data, perform analysis, and present the results in the four boxes.

The first box contains a brief introduction of the product. The second box contains the consistency analysis result, including the number of analyzed reviews and the consistency score that is $1 - PCI(\bar{h}_d)$. The third box is the introduction of ERO principle. The fourth box is the REO analysis results.

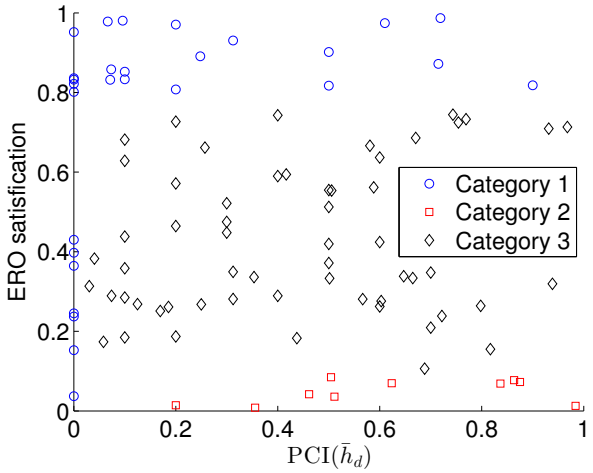

Fig. 4: Case study.

C. Case Study

In this section, we conduct case studies on real products from Amazon to demonstrate the process of review manipulation detection. The products are crawled from the *Kitchen & Dining* and *Toy & Game* categories.

First, we assume the database already contains some products. These products can be either based on ReviewSec user requests or randomly selected by the ReviewSec developer. Specifically, we randomly pick 114 products from those two categories. Following the procedure in III-B2, we set up the threshold selection line for the consistency analysis based on these existing products. In particular, the changing threshold ν is set to 0.6, and the starting threshold $\bar{h}_0$ is set to 3.

Second, we crawl 102 products as testing data based on the following guideline. The number of reviews is greater than 50, the average rating is between 2.5 and 4.5, and the products have discounts from 15% off to 40% off. We apply the consistency analysis to get the PCI values. The ERO principle upon ERO feature "day of the week" is examined. The result is plotted in Figure 4. The x-axis is $PCI(\bar{h}_d)$, and the y-axis is the ERO satisfaction.

We can see that the products are classified into 3 categories.

1) Category 1: *Normal* product, whose ERO satisfaction is greater than 0.8 or PCI value is 0.
2) Category 2: *Highly suspicious* product, whose $PCI(\bar{h}_d)$ is non-zero and ERO satisfaction is less than 0.1.
3) Category 3: *Suspicious* product, which belongs to neither category 1 nor category 2.

In the future, we will conduct detailed investigation on these products that are detected as highly suspicious. For example, we may obtain experts opinions on these products. Furthermore, since ReviewSec system is ready, we would like to involve real users in testing and obtain their feedback. For instance, we would like to know whether ReviewSec can reduce the amount of time that the users study reviews before making purchasing decisions, and increase the confidence of the users in their decisions.

V. Conclusion

The ERO principle, which is a new angle to detect review manipulation, has been proposed in this work. Together with the consistency analysis, real-time data crawler, and the web-based user interface, we present ReviewSec, an on-demand review analysis service. Using ReviewSec, an individual user can check whether the reviews of a particular product are manipulated or not. This is particularly useful because many online review systems today are constantly under attack. A few case studies have been presented to demonstrate the proposed algorithm and the ReviewSec System. In the future, we look forward to perform testing with real user involvement.

References

[1] *Final Pre-Christmas Push Propels U.S. Online Holiday Season Spending Through December 26 to Record $30.8 Billion*. comScore, 2010. [Online]. Available: http://ir.comscore.com/releasedetail.cfm?ReleaseID=539354

[2] J. Morgan and J. Brown, "Reputation in online auctions: The market for trust," *California Management Review*, vol. 49, no. 1, pp. 61–81, 2006.

[3] A. HARMON, *Amazon glitch unmasks war of reviewers*. The New York Times, February 14, 2004. [Online]. Available: http://www.nytimes.com/2004/02/14/us/amazon-glitch-unmasks-war-of-reviewers.html

[4] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, "A calculus for access control in distributed systems," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 4, pp. 706–734, Sep. 1993. [Online]. Available: http://doi.acm.org/10.1145/155183.155225

[5] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007, emerging Issues in Collaborative Commerce. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923605000849

[6] P. Laureti, L. Moret, Y.-C. Zhang, and Y.-K. Yu, "Information filtering via iterative refinement," *EPL (Europhysics Letters)*, vol. 75, no. 6, p. 1006, 2006.

[7] B. Yu and M. P. Singh, "An evidential model of distributed reputation management," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, ser. AAMAS '02. New York, NY, USA: ACM, 2002, pp. 294–301.

[8] Y. Liu and Y. Sun, "Anomaly detection in feedback-based reputation systems through temporal and correlation analysis," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, Aug 2010, pp. 65–72.

[9] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, ser. EC '00. New York, NY, USA: ACM, 2000, pp. 150–157. [Online]. Available: http://doi.acm.org/10.1145/352871.352889

[10] A. Jsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, 2002, pp. 41–55.

[11] W. Jianshu, M. Chunyan, and G. Angela, "An entropy-based approach to protecting rating systems from unfair testimonies," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 9, pp. 2502–2511, 2006.

[12] *ReviewPro*. [Online]. Available: http://www.reviewpro.com/

[13] *TrustYou*. [Online]. Available: http://www.trustyou.com/

[14] Y. Liu, Y. Sun, and T. Yu, "Defending multiple-user-multiple-target attacks in online reputation systems," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, Oct 2011, pp. 425–434.