Detecting Collusive Cheating in Online Shopping Systems through Characteristics of Social Networks

Jianwei Niu[1], Lei Wang[1], Yixin Chen[2], Wenbo He[2]
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China[1],
McGill University, Canada[2]
niujianwei@buaa.edu.cn, wangleildmc@gmail.com, chenyixinabel@gmail.com, wenbohe@gmail.com

***Abstract*—Detecting the collaborative cheating in an online shopping system is an important but challenging issue. In this paper, we propose a novel approach to detect the collusive manipulation on ratings in Amazon, an online shopping system. Rather than focusing on rating values, we believe the online shopping and rating activities have nontrivial attributes in terms of social network connections. Our major contributions include: (a) We build a virtual social network based on users' ratings and comments, and detect the collusive cheating based on the social network activities. (b) We investigate the properties of disconnected components in a wide range of social networks, such as the longevity and final size of the disconnected components before they join the giant connected component or merge with other disconnected components. (c) We apply our proposed collusion detection algorithm to detect the possible collusive cheating on the ratings based on the data we crawl from Amazon, and the experimental results validate our approach.**

I. Introduction

With the increase of the popularity and scale of the online shopping systems such as Amazon, eBay and Taobao, reputation systems have been widely deployed to protect the honest buyers and the reputable sellers through the customer ratings, reviews, and the user recommendations and referrals. However, the reputation systems are in general vulnerable to many types of attacks [6], where selfish users try to manipulate the rating. A typical example is that a group of users collaboratively subvert the rating of a given product or service. Another example is so-called Sybil attack [1] where a user creates a large number of identities and uses them to gain a large influence [9]. With the desire to promote their own products, users have strong motivation to cheat by providing unfair ratings.

Detecting the unfair ratings introduced by collaborative manipulation (from either a single Sybil attacker or a group of collusive users) is an important but challenging issue. Previous efforts have been based on the investigation of rating values [18]. For example, if a rating value is far away from the majoritys opinion, or if we observe a sudden change of the rating values in a short time period, there is likely to be an attack manipulating the rating scores. The rating-value-based detection schemes usually make several assumptions such as (1) the number of manipulated ratings is less than the number of honest ratings; (2) the bias of the manipulated ratings is sufficiently large; (3) there is no bursty rating input.

In this paper, we propose a novel approach to detect the collusive manipulation on ratings in real world. Rather than focusing on rating values, we believe that the online shopping and rating systems have the common attributes as the online social networks in terms of social connections. Hence, we first model the normal behavior of the honest users in online shopping and rating systems as a user in a virtual social network, and we identify characteristics of normal users from a wide range of online social networks. We then check if a user's behavior in the virtual social network deviates from the observed normal behavior to identify suspicious users for future scrutiny. The contributions of this paper are summarized as follows:

(1) We build a virtual social network based on users' ratings and comments, and detect the collusion based on the characteristics of the virtual social networks. We observe that collusive users trying to promote or badmouth a specific group of products, usually by using a special account to do so and the account may not be used frequently. So in the virtual social network, if we connect two users when they make similar comments or ratings on the same product or user, the collusive users are likely to form disconnected components. By modeling the collusive users in this way, we employ the characteristics of online social networks to detect suspicious users in the online shopping and rating systems.

(2) We investigate the characteristics of the disconnected components in a wide range of social networks, such as the longevity and size of the disconnected components. We observe that the disconnected components exhibit similar characteristics in different social networks, hence we obtain a generic model for normal social network users. Then, we detect the collusion by checking if the disconnected components in the virtual social network behave significantly different from the model.

(3) To evaluate the proposed approach, we crawl the Amazon data including 7,659,294 transactions and 123,384 users. We applied the proposed collusion detection algorithm to detect the possible collusive manipulation on the ratings, and found 836 suspicious users. After the closer investigation by people, 704 of detected users are considered cheating. We believe that the proposed approaches have a great impact on collusion detection in online shopping systems.

The rest of this paper is organized as follows. Section II introduces the related work in this field. Section III describes how to model an online shopping and rating system as a virtual social network. In Section IV, we study a wide range of online social networks in the real world and show the characteristics of them. We propose a collusion detection algorithm in Section V,

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

and evaluate it in Section VI. We conclude this paper in Section VII.

II. Related Work

It is usually very important to analyze the properties of networks, especially the static properties reflect characterizations of a network. The power-law degree distribution [2] and the small-world phenomenon [16] are two typical properties gaining widespread focuses. In addition, Leskovec et al. [10] found that with the adding of nodes the number of edges in a network increases at rate between the linear growth and the quadratic growth while the density of the network will increase gradually.

There are also a few publications on the disconnected components in networks in previous studies. In [7] Kumar et al. classified the connected components into three types and found that the disconnected components in the middle regions always have star-shaped structures. McGlohon et al. [12] found that in the evolutionary process of a network, there exists a "gelling point", where disconnected components merge into a giant component and the diameter of the network spikes.

There is a rich variety of research literature about reputation systems and Sybil defense schemes. In [4], a comprehensive survey of reputation systems is conducted. An analytical framework is proposed to decompose, analyze, and compare a few influential reputation systems, to further evaluate their strengths and weaknesses. In [5], the authors use 2-gram based review content comparison to detect duplicate or near-duplicate reviews, and then focus on outlier reviews.

In [11], TAUCA (joint Temporal And User Correlation Analysis), an anomaly detection scheme is proposed. The scheme exploits time domain information usually ignored by other schemes, under the assumption that abnormal time intervals between reviews of a product will appear in attacks. In [18], it is observed that Sybil-defense schemes based on majority rule assumption are incapable of detecting smart attacks under certain scenarios.

In [19], by verifying an implicit assumption in [20] that social networks are fast mixing, the authors improve the performance of the Sybil defense scheme. [17] argues that the improvement in [19] is still away from the theoretical lower bound and the assumption about the small cut, in both [20] and [19] is questionable. In [13], it is realized that the inherent inequity of the trust (e.g., co-authors vs. social friends) in social networks can be exploited to defend against Sybil attacks. It is supported by an algorithmic property, the mixing time. In [15], the fundamental thoughts underlying ostensible different Sybil defense schemes based on social networks are examined.

III. Model a Rating System as a Virtual Social Network

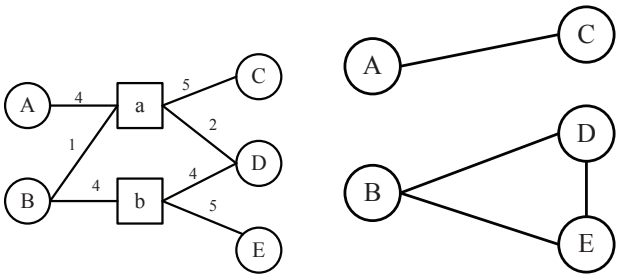
A social network can be modeled as a graph with a set of nodes and edges between them. A node represents a user, and an edge between two nodes is created if the two nodes are friends, or they contact with each other via email, or they just take part in the same activity. In other words, an edge between two nodes in a social network means the two nodes share a certain similarity. While in an online shopping and rating system, each user can be regarded as a node in a network, and the two users or nodes are connected if they give similar comments or rating scores[1] to a product. In this way, we can model an online shopping and rating system as a virtual social network as follows.

Let $Reviews$ be the set of all reviews. For each $review \in Reviews$, let $review_{id}$ be the product ID, $review_{score}$ be the score given by a user, $review_{user}$ be the user's name who posted the review, $review_{date}$ be the date of the review. For $reviewA, reviewB \in Reviews$, if $reviewA_{id}$ equals to $reviewB_{id}$ and $|reviewA_{score} - reviewB_{score}| \leq 1$, we think $reviewA_{user}$ and $reviewB_{user}$ have resemblances. So an edge between $reviewA_{user}$ and $reviewB_{user}$ is created if it did not exists before and its timestamp is set to t_{max} which equals to the larger one of $reviewA_{date}$ and $reviewB_{date}$. If the edge between $reviewA_{user}$ and $reviewB_{user}$ with the timestamp ts has been created before, the timestamp of the edge is updated to t_{max} if ts is larger than t_{max}. The timestamp information can help us to obtain the structure of the virtual social network at any time point by removing all the nodes and edges which appear later than this time point.

Usually, the online rating systems frown on the collusive cheating and have their own policies preventing the collusion among users, though they are transparent to users. For example, once you are involved in collusion and get caught, Amazon may remove your reviews, and even stop selling your products. In this case, users involved in collusion typically have separate accounts for regular listing and purchasing activities and for posting unfair ratings. Hence, even if the collusion is gotten caught, their regular activities will not be affected. With this observation, we know that after modeling an online shopping and rating system as a virtual social network, the users involved in collusion are relatively isolated from the rest of the network. However, the studies on social networks have shown that disconnected components are usually connected to the giant component [8] [3] soon after their appearing. It implies that the characteristics of the disconnected components in online social networks is helpful to judge if an isolated group of users in online shopping and rating system is normal.

Let's consider an example in Figure 1, where we demonstrate how to detect the collusion by modeling of the online shopping systems and characterizing the social networks. In Figure 1 (a), each circle represents a user in the rating system and each block represents a product. An edge between a user and a product means the user gave a rating to the product. The virtual social network is obtained by the above modeling procedure, and it is shown in Figure 1 (b). User B, C and D gave relatively high scores to product b while the average score of b is only 1.3. What's more, most of scores they gave have great deviations from the products' average scores, so the connected component

[1] In Amazon, the rating score of a product ranges from 1-star to 5-star. 4-star or 5-star implies the user is satisfied with the product while 1-star or 2-star means the customer is disappointed at the item.



(a) *Rating system* (b) *Virtual social network*

Fig. 1. Model a rating system as a virtual social network

TABLE I
THE DATASETS WE STUDY

Name	$\|N\|$	$\|E\|$	Type
enron	87, 273	61, 148, 072	Directed
wikipedia	30, 997	201, 727	Directed
flickr	230, 292	3, 314, 001	Bipartite
youtube	3, 826, 344	4, 835, 698	Directed

consisting of B, C and D remains a quite long time without any node added into it, meaning the component stays in a relatively isolate state. The users B, C and D are highly suspicious.

IV. CHARACTERIZING SOCIAL NETWORKS

A. Datasets Description

To investigate the characteristics of the disconnected components in social networks, we collected four datasets of the social networks. The *enron*[1] dataset describes an email network consists of emails sent between employees of Enron. The *wikipedia*[2] dataset contains users and pages from the *Wikipedia*[3], connected by edit events. The *flickr*[4] dataset describes the social network of *Flickr*[5] users and their friendship connections. The *youtube*[6] dataset describes a friendship network of the famous website *YouTube*[7].

Table I shows the details about the datasets we use, which include the number of nodes and edges, and the type of network for each dataset. In each dataset, there are lists of connections (edges) and timestamps (creation times) for each connection.

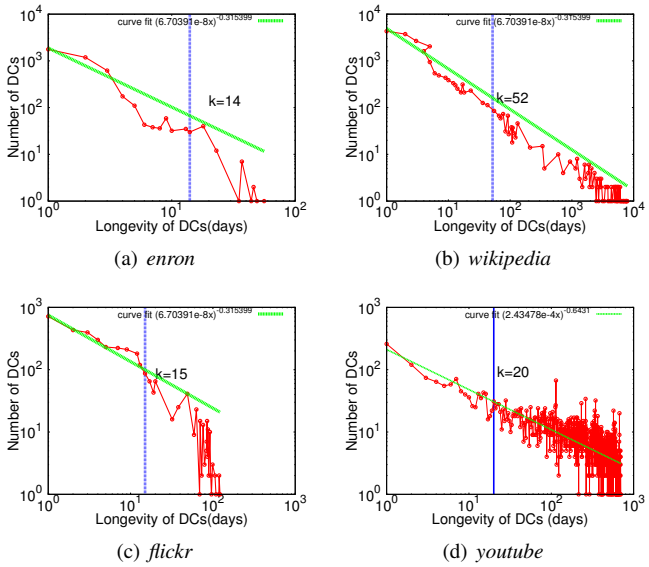
B. Interested Characteristics

As we know, the connected components in a network can be divided into two classes: the giant connected component and disconnected components. The Giant Connected Component (GCC) contains a significantly large fraction of nodes while a Disconnected Component (DC) is defined as a small component that is not connected to any other components in the network.

As we mentioned in the previous section, we want to propose some properties which could describe the frequency of the nodes in a connected component contact with other connected components. If two nodes from two different connected components are connected, the two components are merged into a new connected component and they are regarded as "dead". A connected component which keeps "alive" for a very long time implies the nodes in it rarely contact with the nodes from other connected components.

In [14], the authors proposed the *longevity* and *final size* of disconnected components. We think these two properties can help us to identify collusive cheating in a rating system.

[1]http://konect.uni-koblenz.de/networks/enron
[2]http://konect.uni-koblenz.de/networks/edit-frwikibooks
[3]http://www.wikipedia.org
[4]http://konect.uni-koblenz.de/networks/flickr-growth
[5]http://www.flickr.com
[6]http://socialnetworks.mpi-sws.org/data-wosn2008.html
[7]http://www.youtube.com



(a) *enron* (b) *wikipedia*
(c) *flickr* (d) *youtube*

Fig. 2. Longevity distribution of DCs in each dataset.

C. Longevity and Final Size of Disconnected Components

We first study the longevities of DCs. The results are shown in Figure 2. Figure 2 sees an apparent decaying trend from all curves, which means that the short-lived DCs account for a significantly large fraction among all the DCs. In addition, we notice that there is often a "emanative" point k where the number of DCs whose longevity is longer than a certain value k begins to oscillate remarkably.

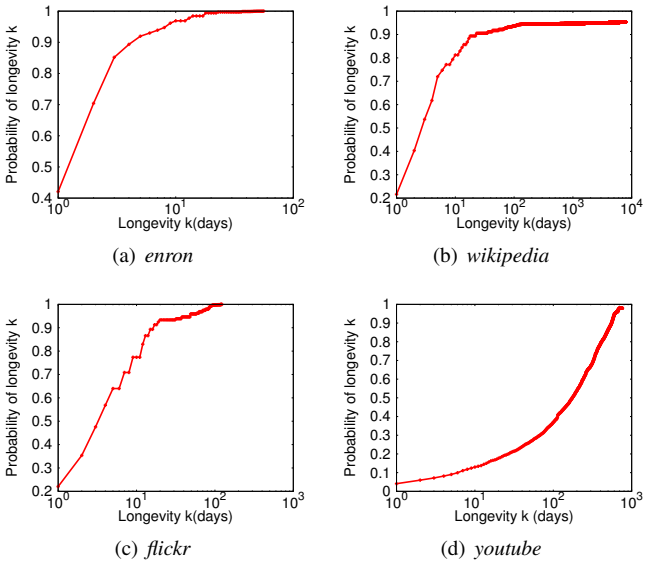
Now we plot the cumulative distribution function (CDF) of DCs' longevity in social networks and the results are displayed in Figure 3. The curves show most of DCs are short-lived as well.

We now focus on the final size of the DCs and the results are displayed in Figure 4. From Figure 4, we see that with the increase of final sizes of DCs, the slopes of curves decrease gradually, which means the number of dead DCs with a small size is much greater than the number of dead DCs with a large size.

All the observations in this section keep consistent in the observations in [14].

V. COLLUSION DETECTION ALGORITHM

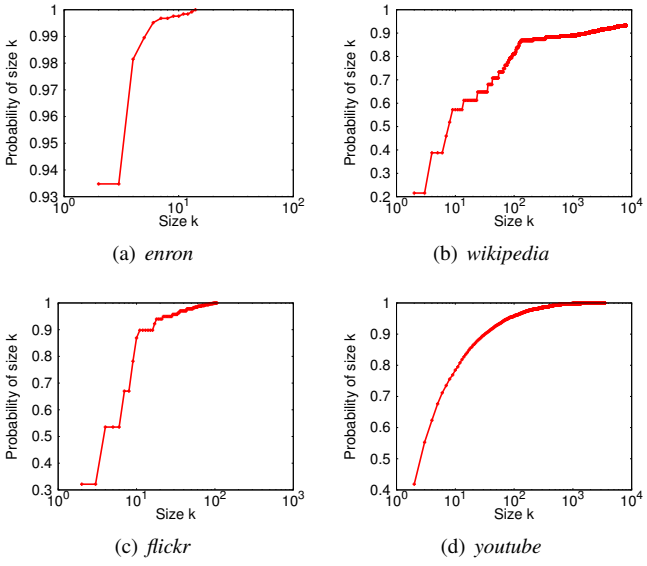
In this section, we focus on the detection of collusive nodes in rating systems. As mentioned in Section III, a rating system can be modeled as a virtual social network.



(a) *enron* (b) *wikipedia*

(c) *flickr* (d) *youtube*

Fig. 3. The CDFs of DCs' longevities.



(a) *enron* (b) *wikipedia*

(c) *flickr* (d) *youtube*

Fig. 4. The CDFs of DCs' final sizes

According to the observations, the majority of DCs in a social network die with quite short longevities and small sizes. Long longevity of a DC implies the isolation of nodes in it. For users represented by these collusive nodes, they only give comments or ratings to specified products and stay in a relatively isolated state, increasing the longevity of the group they belong to. In addition, since popular products have tons of reviews, the sizes of the groups constituted by users giving these reviews are relatively large. However, in our experience, collusive nodes, or users, tend to give reviews or comments to unpopular products to obtain unfair benefit. So the sizes of the groups consisting of collusive nodes should be small. Therefore, it is more likely that the nodes in the long-lived and small-size DCs are collusive.

We tag each node an attribute p which indicates the probability of a suspicious node to be a normal node. The initial value of p is a constant $init$ (an empirical constant for a specific system), ranging from 0 to 1. The suspicious node is considered as a real collusive node whose p is smaller than a threshold α.

Using the method from Section IV we can get the longevities and final sizes of all DCs at any time point. As we mentioned before, the component with quite long longevity and small size is regarded as isolated one and the nodes in them are considered to be suspicious. That is to say a node is regarded suspicious only when the connected component it belongings to has the longevity larger than a threshold β and the final size smaller than a certain threshold γ.

Now we discuss how to determine the values of β and γ. For a certain network, we can obtain the CDFs of DCs' longevities and final sizes using the method mentioned in Section IV. Let $f(x), g(x)$ denote the CDFs of DCs' longevities and final sizes respectively. For a certain β and a certain γ, we can get $p_1 = f(\beta), p_2 = g(\gamma)$. The value of α, which is the threshold judging whether a suspicious node is a real collusive one or not, is determined by the longevity and final size of the DC. So we can set

$$\alpha = (1 - p_1) * p_2. \quad (1)$$

Each suspicious node is set to be collusive by comparing a certain number α ($0 < \alpha < 1$) with p. If α is larger, the node is regarded as a collusive node. The value of p will be decreased each time when the node is examined to be a suspicious node. So the probability $(1 - p)$ that a node is a collusive node increases if the node is checked as a suspicious one.

With these considerations, our collusion detection algorithm is displayed in Algorithm V.1 where we set the time interval of the dataset is day. At the beginning, the probability of being a normal node (we use p to denote it) for the all nodes in the network is $init$. At the end of each day, we can calculate the longevities and final sizes of all DCs. The nodes in the DCs whose longevities are larger than β, and whose final sizes are smaller than γ are considered suspicious. The suspicious node is regarded as a collusive node if its p is smaller than α.

VI. Evaluation

A. Data Collection

We have evaluated our proposed collusion detection approach on Amazon, where users have enough motivation to provide biased or fraudulent ratings or comments for their own benefits. We designed a data crawler, to collect data from Amazon. The crawler searches products in Amazon by Deep-First-Search (DFS) and related information about the products is collected and stored in a database. In order to make the crawler more efficient, a few rules were defined to form a filter, filtering out links not worth crawling. The crawler had worked for seven days, and crawled 42,828 productions, 7,659,294 users' ratings about these products.

B. Characteristics of the Virtual Social Network

We crawled a dataset from Amazon and built a virtual social network based on the crawled dataset. To validate our proposed approach on collusion detection by modeling an online shopping system as a virtual social network, we found

Algorithm V.1 ***Collusion Detection*** Algorithm

Input: G-the virtual social network, p_1-the percentage of normal DCs in all DCs, p_2-the percentage of DCs which give comments to the unpopular items, $init$-a constant between 0 and 1.
Output: $CollusiveNodesList$-collusive nodes list.

$CollusiveNodesList \leftarrow \emptyset$
$Nodes \leftarrow$ all nodes appear in G
$\beta = f^{-1}(p_1)$
$\gamma = g^{-1}(p_2)$
/*$f(x), g(x)$ are the CDFs of DCs' longevities and final sizes*/
$\alpha = (1 - p_1) * p_2$
for all $node \in Nodes$ **do**
  $node.p = init$
end for
for all $i \in \{1, 2, ..., m\}$ **do**
  /*m is the length of time duration for the dataset*/
  CalculateDCLongevityAndFinalSize(i)
  /*to obtain all DCs' longevities and final sizes at i th day*/
  $S_i \leftarrow$ the set of DCs whose longevity is larger than β and final size is smaller than γ at i th day
  for all $e \in S_t$ **do**
    for all $node \in e$ **do**
      MakeDecision($node$,e)
    end for
  end for
end for
return
function MakeDecision($node, DC$)
/*Make sure whether $node$ is a collusive node*/
if $node.p \leq \alpha$ **then**
  $node$ is a collusive node
  $CollusiveNodesList \leftarrow CollusiveNodesList \bigcup node$
else
  $node.p = node.p * init$
end if

there is an obvious similarity between the virtual social network and the social networks studied in Section IV. Hence, it is valid to detect abnormal behavior (e.g., collusion) according to the characteristics of online social networks.

C. Experimental Evaluation

In a real world, judging whether or not a user is involved in collusion is quite subjective and can be different from person to person. For example, a user may be regarded as a collusive node if most of his ratings have a great deviation from the products' average scores. However, the reason for this may be that the user has different tastes from others. Moreover, the number of reviews in the dataset is too huge to manually examine these reviews' validity. Our collusion detection algorithm works as a filter, picking up a handle of collusive nodes from massive amount of users, which can be further investigated manually according to the information of reviews and products.

There are three parameters to be decided in our collusion detection algorithm. Now we discuss the effectiveness of algorithm under different cases.

First, we set $p_2 = 0.8$, $init = 0.75$ and analyze the performance of the algorithm when p_1 is assigned to different values. Since we can not obtain all the real collusive nodes (ground truth) in the dataset, it is impossible to calculate the false positive rate of our algorithm. We invited 20 volunteers to pick up the real collusive nodes according to their opinions from the collusive nodes detected by our algorithm. These volunteers contain 10 male students and 10 female students chosen from Beihang University. Then we can calculate the false negative rate of our algorithm. The result is shown in Table II. In general, the number of real collusive nodes detected by our algorithm decreases with the increase of p_1. However, the false negative rate decreases first, reaches the nadir and then increases with the increase of p_1. We think our algorithm can reach the best performance when $p_1 = 0.95$ in terms of the number of detected collusive nodes and false negative rate.

TABLE II
THE RESULT OF THE ALGORITHM WHEN p_1 VARIES

p_1	detected nodes	real collusive nodes	false negative rate
0.80	1,235	743	39.84%
0.85	1,076	742	31.04%
0.90	965	738	23.52%
0.93	922	730	20.82%
0.95	823	719	12.64%
0.98	751	612	19.44%

TABLE III
THE RESULT OF THE ALGORITHM WHEN p_2 VARIES

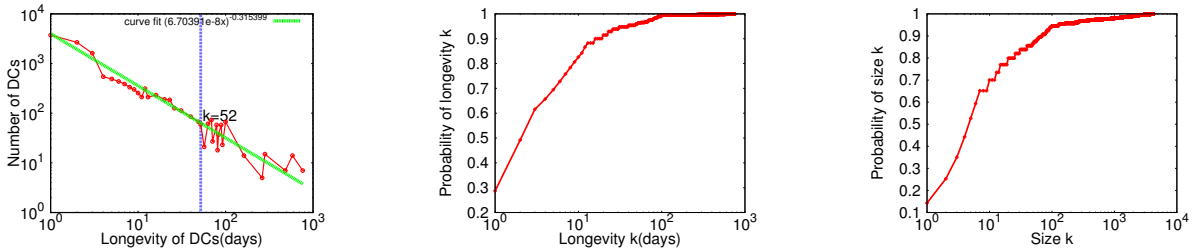
p_2	detected nodes	real collusive nodes	false negative rate
0.75	745	648	13.02%
0.80	823	719	12.64%
0.85	874	743	14.99%
0.87	853	752	11.84%
0.90	877	758	13.56%
0.95	903	787	12.85%

From the observation about the distribution of DCs' longevities, we know that the number of DCs which have longevities longer than a certain value k (e.g., 52) begins to oscillate remarkably. From 5(b) we see the percentage of DCs whose longevities are smaller than k is quite near to 0.95, which is thought to be the optimal value for p_1. So it is reasonable to hypothesize that the emanative point of CDF for DCs' longevities is consistent with the best assignment to β.

Then we set $p_1 = 0.95$, $init = 0.75$ and compare the effectiveness of our algorithm when p_2 is assigned to different values. We still investigate the false negative rate of our algorithm when p_2 varies. The result is displayed in Table III.

From Table III, we observe that the number of detected nodes by our algorithm is "U"-type distribution with the increase of p_2. At the same time, the false negative rate ranges within a small scale.

Let S_i denote the set of detected collusive nodes when $p_2 = i$ and $Diff_{i,j}$ denote $S_i - S_j$. The values of $Diff_{i,j}$ are displayed in Table IV. With the increase of p_2, the number of detected real collusive nodes by our algorithm increases generally. From Table IV we can see the values of $Diff_{i,j}$



(a) *Longevity distribution* (b) *CDF of longevity* (c) *CDF of size*

Fig. 5. The characteristics of the virtual social network corresponding to Amazon

TABLE IV
THE RESULT OF $Diff_{i,j}$

i \ j	0.75	0.80	0.85	0.87	0.90	0.95
0.75	0	2	2	3	5	8
0.80	73	0	13	17	16	25
0.85	97	37	0	34	43	58
0.87	107	50	43	0	56	67
0.90	115	55	58	62	0	69
0.95	147	93	102	102	98	0

are quite small when $i < j$, which means that S_j contains most nodes in S_i when $i < j$.

Limitations discussion. We acknowledge that there were a couple of limitations to our study. First of all, the recall rate and false positive rate of our algorithm are not available currently because of the lack of the ground truth on how many real collusive users exist in the Amazon rating system. Second, we have not conducted comparative experiments for our approach because, to the best of our knowledge, we are the first to investigate the collusive cheating in online shopping systems by analyzing characteristics of social networks.

VII. Conclusion

We made empirical observations and analysis on the longevities and final sizes of the disconnected components in four real networks. We proposed a model which models a rating system as a virtual social network and put forward a collusion detection algorithm to detect the collusive nodes in the virtual social network. We then validated our algorithm using the data we collected from Amazon , showing our observations and analysis are effective.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61170296 and 61190125), R&D Program (2012BAH07B01 and 2013BAH35F01), 973 Program (2013CB035503).

References

[1] J. R. Douceur. The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer, 2002.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[3] L. Fleming and K. Frenken. The evolution of inventor networks in the silicon valley and boston regions. *Advances in Complex Systems*, 10(01):53–71, 2007.

[4] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1, 2009.

[5] N. Jindal and B. Liu. Analyzing and detecting review spam. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 547–552. IEEE, 2007.

[6] A. Jøsang and J. Golbeck. Challenges for robust trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, 2009.

[7] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (SIGKDD 06)*, pages 611–617. ACM, 2006.

[8] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Link Mining: Models, Algorithms, and Applications*, pages 337–357. Springer, 2010.

[9] M. Lazzari. An experiment on the weakness of reputation algorithms used in professional social networks: the case of naymz. In *Proceedings of the IADIS International Conference e-Society*, pages 18–21, 2010.

[10] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th International Conference on Knowledge Discovery and Data Mining (SIGKDD 05)*, pages 177–187. ACM, 2005.

[11] 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*, pages 65–72. IEEE, 2010.

[12] M. McGlohon, L. Akoglu, and C. Faloutsos. Weighted graphs and disconnected components: patterns and a generator. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining (SIGKDD 08)*, pages 524–532. ACM, 2008.

[13] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM, 2011 Proceedings IEEE*, pages 1943–1951. IEEE, 2011.

[14] J. Niu, J. Peng, C. Tong, and W. Liao. Evolution of disconnected components in social networks: Patterns and a generative model. In *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, pages 305–313. IEEE, 2012.

[15] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 363–374. ACM, 2010.

[16] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998.

[17] W. Wei, F. Xu, C. C. Tan, and Q. Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1951–1959. IEEE, 2012.

[18] Y. Yang, Y. Sun, S. Kay, and Q. Yang. Securing rating aggregation systems using statistical detectors and trust. *Information Forensics and Security, IEEE Transactions on*, 4(4):883–898, 2009.

[19] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17. IEEE, 2008.

[20] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278, 2006.