Detecting Crowdturfing "Add to Favorites" Activities in Online Shopping

Ning Su
DCST, Tsinghua University
Beijing, China
sn-40@163.com

Yiqun Liu*
DCST, Tsinghua University
Beijing, China
yiqunliu@tsinghua.edu.cn

Zhao Li
Alibaba Group
Hangzhou, China
lizhao.lz@alibaba-inc.com

Yuli Liu
Qinghai University
Qinghai, China
liu-yuli@foxmail.com

Min Zhang
DCST, Tsinghua University
Beijing, China
z-m@tsinghua.edu.cn

Shaoping Ma
DCST, Tsinghua University
Beijing, China
msp@tsinghua.edu.cn

ABSTRACT

"Add to Favorites" is a popular function in online shopping sites which helps users to make a record of potentially interesting items for future purchases. It is usually regarded as a type of explicit feedback signal for item popularity and therefore also adopted as a ranking signal by many shopping search engines. With the increasing usage of crowdsourcing platforms, some malicious online sellers also organize crowdturfing activities to increase the numbers of "Add to Favorites" for their items. By this means, they expect the items to gain higher positions in search ranking lists and therefore boost sales. This kind of newly-appeared malicious activity proposes challenges to traditional search spam detection efforts because it involves the participation of many crowd workers who are normal online shopping users in most of the times, and these activities are composed of a series of behaviors including search, browse, click and add to favorites.

To shed light on this research question, we are among the first to investigate this particular spamming activity by looking into both the task organization information in crowdsourcing platforms and the user behavior information from online shopping sites. With a comprehensive analysis of some ground truth spamming activities from the perspective of behavior, user and item, we propose a factor graph based model to identify this kind of spamming activity. Experimental results based on data collected in practical shopping search environment show that our model helps detect malicious "Add to Favorites" activities effectively.

KEYWORDS

Online Shopping; Crowdsourcing Manipulation; Spam Detection

ACM Reference Format:
Ning Su, Yiqun Liu, Zhao Li, Yuli Liu, Min Zhang, and Shaoping Ma. 2018. Detecting Crowdturfing "Add to Favorites" Activities in Online Shopping. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186079

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186079

1 INTRODUCTION

Online shopping sites, such as Amazon and Taobao, have become popular platforms for people to find and buy items. For these sites, user behavior data plays an important role in the optimization of their personalized recommendation and shopping search results [13, 18]. When shopping online, users sometimes want to save some potentially interesting items for future purchase activities. For this situation, most online shopping sites provide an "Add to List" (in Amazon) or "Add to Favorites" (in Taobao, hereafter referred to as "A2F") function for users. While bringing convenience to users, online shopping sites can also benefit from this kind of behavior data. For example, sometimes the amount of A2F, also called popularity, is regarded as a facet of item ranking in shopping search result pages. This information can also be used in the default ranking process of shopping search engines [13].

Nowadays, with the wide usage of crowdsourcing systems, some online sellers try to manipulate the ranking of shopping search results by increasing their items' popularity with the help of crowd

*Corresponding author

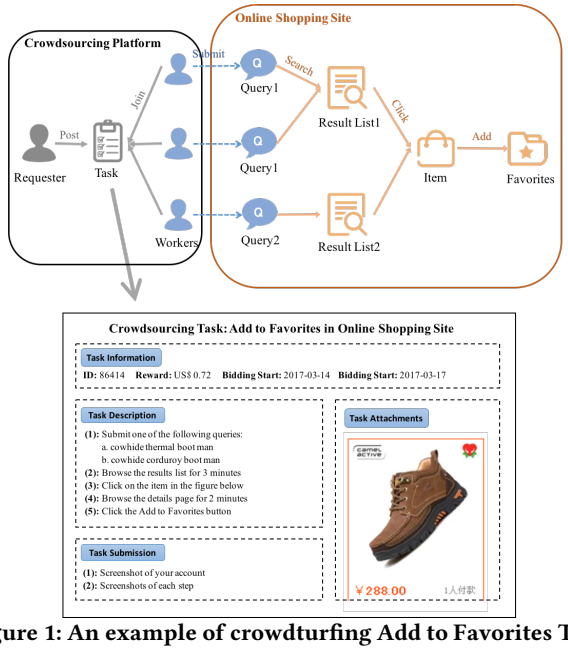


Figure 1: An example of crowdturfing Add to Favorites Task

workers and to boost sales. As shown in Figure 1, a malicious online seller posts a task on a crowdsourcing platform to increase his/her item's A2F amount (popularity). In this task, crowd workers need to follow some particularly designed guidelines to disguise themselves as normal users. First, crowd workers need to submit a specific query to the target shopping search engine and click on the item which the malicious seller wants to prompt. Then, crowd workers need to stay in the item details page for a while, usually 2 minutes at least, and then click the "Add to Favorites" button. To simulate a more realistic online shopping scenario, some tasks may even require crowd workers to browse the results list for a while and click on a random number of non-target items.

After accepting these crowdturfing tasks, crowd workers should first take a screenshot of their account ID in online shopping site, and then follow the guidelines to perform the tasks, with a screenshot at each step. The task requesters only approve those submissions that meet their requirements according to the screenshots, and pay the remuneration. These spamming activities affect the ranking strategy and recommendation mechanism of online shopping sites. Meanwhile, they will mislead normal users, because occasionally malicious sellers are trying to prompt low-quality or even fake items with these crowdturfing efforts.

In this paper, we aim to detect the above-described newly-appeared spamming activities in online shopping. Compared to prior works, many challenges arise regarding this detection task: (1) These spamming activities are composed of a series of user behaviors, including search, browse, click and add to favorites, which are more complex and more challenging to be detected. So we need to track and analyze users' whole behavior sequences. This is quite different from content-based spamming activities, such as deceptive product reviews [9, 23], promotion campaigns in Community Question Answering (CQA) [14, 26] and promotional microblog posts [4]. And this is also different from fake likes in OSNs [1, 8], which only need a simple user action. (2) Since these spamming activities are performed by crowd workers, they are very similar to normal ones and difficult to be detected even with manual efforts. (3) Compared to posting spam product reviews and organizing CQA campaigns, these activities are private and hardly noticed by the public. So there is a lack of effective indicators such as "review helpfulness" or "selected as best answer" [17]. (4) Some tasks set requirements of crowd workers' account in online shopping site (e.g. the accounts should be registered at least 2 years ago). Therefore, these crowd workers in spamming activities are normal users for most of time. They will also carry out some normal A2F activities by themselves (see Section 4.3). Meanwhile, with the increasing popularity of deceptive items, a number of normal users may also be attracted by them and contribute normal A2F activities. In other words, both spam users and items have a part of normal records. This also brings more challenges to the detection process.

To tackle these challenges, we first exploit a number of crowdturfing tasks to form the dataset , and look into the corresponding user behavior log records that are highly likely to be spamming activities. Then we analyze the attributes of these crowdturfing A2F activities from the perspective of behavior, user and item. By integrating attributes and correlations (user-based and item-based) with a factor graph model, we conduct a discriminative model to detect spamming A2F activities. Through experimental comparisons with competitive baselines, we empirically show that our framework is robust and effective.

Our study has the following contributions:

- We specify the problem of crowdturfing A2F activities in online shopping. To our best knowledge, we are among the first to investigate this type of deceptive activities.
- Through simultaneously locating crowdturfing A2F tasks and collecting user behavior log from online shopping sites, we create a dataset which contains both normal user behavior data and a number of ground truth spamming activities.
- We provide a comprehensive analysis of these spamming activities from the perspective of behavior, user and item.
- We propose a novel detection framework that can effectively detect spamming activities.

2 RELATED WORK

Three lines of research works are highly related with the detection of crowdturfing A2F activities: individual spam detection, collusive spam detection and crowdsourced manipulation.

2.1 Individual Spam Detection

With the development of E-Commerce, opinion spam (i.e., deceptive reviews) has attracted much attention. It is firstly presented by Jindal and Liu in [9], in which they analyze Amazon data and identify three types of spam. With manually labeled training examples, they further train a supervised learning model to detect fake reviews, which is called opinion spam. Ott et al. [23] reports that the number of deceptive reviews grown across multiple consumer-oriented review sites. They find that deceptive opinion spam is growing in general, but with different growth rates across communities. Yoo and Gretzel [36] manually compared the psychologically relevant linguistic differences between collected truthful and deceptive hotel reviews. However, the results suggest that it might be difficult to distinguish between deceptive and truthful reviews based on syntactic features. Feng et al. [7] regard the opinion spam as a distributional anomaly. They find distinguishing patterns between ordinary and fake reviews from product review ratings and the time windows when reviews are posted. In [24], Ott et al. use n-gram and part-of-speech(POS) tag features for supervised learning on a gold-standard fake review dataset through Amazon Mechanical Turk. The problem of review spammer detection has also been widely studied in [16, 20, 25]. These research studies identify several features related to rating behaviors and model these features to detect spam reviewers. Lu et al. [20] use a probability graph model to detect fake reviews and review spammers simultaneously on a large labeled dataset.

Besides fake review and review spammer detection in the review systems, spam detection has also been studied in other platforms, such as Community Question Answering (CQA) portals [1, 2, 5, 8, 14, 26] and online social networks (OSNs) [10, 11, 28]

Compared with this line of research, our work aims to deal with a newly-appeared collusive spamming activity which involves crowdturfing activities. However, some of the features adopted in existing individual spam detection may also be inspiring our researches.

2.2 Collusive Spam Detection

A related line of research focuses on collusive spam detection [3, 21, 33–35]. Lu et al. [19] exploit contextual information about reviewers' identities and social networks for improving review quality prediction. They find that social context is useful to find groups of review spammers. Mukherjee et al. [21] are among the first to study spammer groups in review communities. They find that labeling fake reviewer groups is much easier than labeling individual fake reviews or reviewers, and propose a novel relation-based methods to detect spammer groups on the labeled dataset. In [34], Xu et al. propose two novel methods to cluster reviewers and detect collusive spammers, using both individual and collusive indicators. Besides, collusive spam detection is also studied in online social networks. For example, Cao et al. [4] investigate the individual-based and group-based user behavior of URL sharing in social media toward uncovering these organic versus organized user groups.

We can see that most of these collusive spam detection efforts focus on the opinion spam. These deceptive reviews will affect users' judgment about items or services directly. Our research is the first work to specify the problem of crowdturfing A2F activities. Different from previous work, these spamming activities will first affect the ranking strategy and recommendation mechanism of online shopping sites, and then affect users through them.

2.3 Crowdsourced Manipulation

Recently, with the wide usage of crowdsourcing systems, many researchers begin to study the crowdsourced manipulation problem which aims to spread manipulated contents to target sites. Wang et al. [31] find that not only do malicious crowdsourcing systems exist, but they are rapidly growing in both user base and total revenue. They estimate that about 90% of all tasks on two Chinese crowdsourcing platforms are malicious tasks. Lee et al. [12] analyze the types of malicious tasks and the properties of requesters and workers on Western crowdsourcing sites. They further propose and develop statistical user models to automatically differentiate among regular social media users and workers. In [6], the authors link crowdsourced deceptive review tasks to target products. They use a Conditional Random Field model to cluster reviewers and embed the results of this probabilistic model into a classification framework for detecting crowd manipulated reviews. Liu et al. [17] study collusive spamming activities on CQA platforms. They propose a combined factor graph model, using various extracted attributes and correlations to learn to infer whether a question or an answer is deceptive. In [30], the authors formalize the crowd fraud detection problem in Internet advertising, and carefully analyze the behaviors of crowd fraud.

We can see that this line of research provide valuable insights in how crowd workers are organized to finish complex spamming tasks. However, none of them aim to solve the crowdturfing A2F threats exposed to online shopping sites. Compared with previous spamming activities, crowdturfing A2F activities are more subtle and more difficult to be perceived by users. In addition, these spamming activities are composed of a series of user behaviors, which are much more complex and more difficult to be detected than opinion spam.

Table 1: User behavior log record

Field	Description
User id	User's digital id
Search timestamp	The timestamp of query submitting
Query	Submitted query
Ranking type	The selected result ranking type (0 for default)
Item id	Item's digital id
Page number	The page where the item is located
Click timestamp	The timestamp of the click
Dwell time	The time of the user staying in the details page
Shop id	Shop's digital id
Seller id	Seller's digital id
Add to Cart	Whether the user add the item to the cart
Previous clicks	Other clicks before this one (including item id and click timestamp)

3 DATA COLLECTION AND ANNOTATION

Since there is no public available dataset for the problem of spamming A2F activities in online shopping, we aim to collect data first to construct a dataset that can enable us to provide insights and evaluate our algorithms.

3.1 Data Collection

To collect data, we first locate a number of crowdturfing A2F tasks in a crowdsourcing platform as the seed set. Then we collect the related user behavior log based on the seed user set.

As mentioned before, in some popular crowdsourcing platforms, such as Zhubajie.com and RapidWorkers.com, the crowd worker who participates in a crowdturfing A2F task is required to submit a specific query to the shopping search engine and take a screenshot of his/her account ID. This provides a chance to acquire ground truth spamming activities for us. We first locate the crowdturfing A2F tasks in a crowdsourcing platform using manual searching and filtering of the search results. All the queries and crowd workers' account IDs are manually extracted according to the submitted screenshots. Through this way, we obtain 60 tasks during 10 days that contain 113 spam users and 296 unique spam queries (each task may provide more than one query). Meanwhile, we also extract the spam shops manually for later use. We do not extract the spam items because the required items are presented in the form of pictures in the tasks (see Figure 1), and the item descriptions may change frequently.

Based on the common assumption that "spam users tend to post spam contents" in the content-based spamming activities [20, 32], we make two similar assumptions in this problem before the collection process: (1) spam users tend to add spam items to their favorites, and (2) spam items tend to be added to favorites by spam users. We will briefly verify these two assumptions through some examples in Section 4. With the collected account IDs and these two assumptions, we begin to collect the corresponding user behavior log from the target online shopping site, which is considered as one of the most popular e-commerce websites and has a large number of A2F activities each day. The collection process consists of three steps.

Step one: We extract these spam users' behavior log during the period from March 8 to April 13, 2017 (covering the active time for all 60 tasks). Each behavior log record represents an interaction

Table 2: Dataset Statistics

Spam (+)	Normal (-)	Suspicious (?)	All
5,333	156,192	4,110,696	4,272,221

session triggered by shopping search and aimed to finish a A2F activity. Table 1 lists all the fields we extract from the online shopping site.

Step two: Based on the first assumption that spam users tend to add spam items to their favorites, we collect all the items in step one, and then expand the dataset by extracting the user behavior log related to these items during the period.

Step three: Based on the second assumption that spam items tend to be added to favorites by spam users, we identify all the users in step two, and then expand the dataset again by extracting the behavior log of these users during the period.

After these three steps, we constructed a dataset which covers 81,778 users, 1,544,996 items and 4,272,221 user behavior log records. We believe that a large number of users in this dataset may be involved in the crowdturfing A2F activities. However, not all the items in this data set are spam targets because even spam users also perform normal A2F activities.

3.2 Data Annotation

According to our assumption, each user behavior log record in this dataset has a certain probability of spam. However, as mentioned before, both spam users and spam items have a number of normal records. For both algorithm designing and evaluation purposes, we need to annotate the data, spot spam behavior log records, as well as normal ones. Due to the huge size of behavior log and the similarity between spamming activities and normal ones, it is sometimes difficult to ascertain which records are spam and which are normal with manual effort. However, we believe that it is still possible to identify a number of ground truth spamming activities and normal activities based on the crowdturfing task designs. Following are our annotation method:

Spam(+): We first identify a number of ground truth spam behavior log records. Considering the fact that the queries provided by the crowdturfing tasks are usually quite specific and similar to the name of the target items so that crowd workers can find these low-popularity items quickly in the results list, we regard interaction sessions initialized with a spam query as the ground truth spam log records (i.e., spamming activities). With 296 unique spam queries (see Section 3.1), we spot 5,333 spam behavior log records.

Normal(-): Considering the fact that high-quality items will naturally attract a large number of A2F activities, there is no need for their sellers to prompt them with crowdturfing tasks. Therefore, to spot normal behavior log records, we first count the amount of A2F activities for each item in the dataset, and then extract popular items with more than 500 A2F activities. Since the number is much larger than the maximum trading volume (about 100-150) of all crowdturfing tasks during this period, we regard their log records as normal ones. In total, we extract 179 items and 156,192 normal log records. Meanwhile, we count the shops appeared in these log records, and find none of them is spam shop extracted in Section 3.1. This also verifies the rationality of our method.

Suspicious(?): We consider the rest 4,110,696 unlabeled log records are suspicious records. Our goal is to detect spamming A2F activities in these log records.

Table 3: Comparisons of behavior attributes between spamming and normal A2F activities

	Spam	Normal
Add to Cart	0.06	0.08
Rank the results	0.24	0.29
With previous clicks	0.20	0.02
On weekends	0.32	0.26

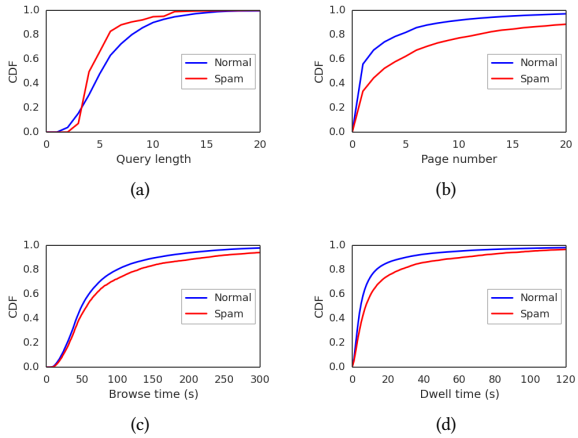


Figure 2: Comparisons of behavior attribute distributions between spamming and normal A2F activities

The statistics of the automatically annotated dataset as described above are shown in Table 2.

4 SPAMMING A2F ACTIVITIES ANALYSIS

Based on the annotated dataset, we make a comparative analysis of the crowdturfing A2F activities. Our analysis will be in three aspects: behavior, user and item.

4.1 Behavior Analysis

We first make a comparative analysis on behavior attributes between spamming and normal A2F activities according to the annotated log records. Table 3 depicts the comparisons between the proportions of spam and normal log records containing the corresponding attributes. As we can observe, only 6% spamming activities (i.e., spam log records) and 8% normal activities contain add to cart operation. It is predictable because the items that users add to their favorites are not what they want to buy immediately, and few tasks require this operation. When searching for items, users will use different ranking strategies provided by shopping search engine to find better items efficiently. Compared to the normal activities, fewer spamming activities contain this operation. As for "*with previous clicks*", we find that in about 20% spamming activities, users have clicked on other items before clicking the target item, while only 2% normal activities have previous clicks on other items. This is because crowdturfing tasks usually require workers to click on a random number of non-target items, as mentioned before. Besides, we also look into when these activities happen. From the table, we can see more spamming activities happen on weekends compared to normal ones. However, except the attribute of "*with previous clicks*", the difference between spamming and normal activities is

Table 4: Comparisons of user attributes between spam and other users

	Spam		Other	
	Mean	Median	Mean	Median
Number of A2F	70.3	42	122.7	65
Number of Add to Cart	42.2	18	62.8	25
Number of purchase	7.0	3	16.6	10
Number of item reviews	4.6	1	9.9	4
Add to Cart / A2F	0.70	0.63	1.35	0.47
Purchase / A2F	0.17	0.04	0.51	0.15

Table 5: Comparisons of item attributes between spam and other items

	Spam		Other	
	Mean	Median	Mean	Median
Number of A2F	288.7	145	1763.7	288
Number of Add to Cart	300.9	102	964.1	143
Number of purchase	71.3	30	633.5	49
Number of item reviews	44.7	23	253.4	25
Add to Cart / A2F	1.13	0.98	1.85	1.51
Purchase / A2F	0.32	0.20	0.97	0.35

very small. This further indicates that spamming activities are very similar to normal ones and difficult to be detected.

Figure 2 further shows the comparisons of behavior attributes between spamming activities and normal ones, in terms of query length, page number, browse time (time interval between search and click) and dwell time (in details page). As Figure 2(a) indicates, the query length of spamming activities is concentrated at 4-6 (about 75%), while for normal activities, the distribution of query length is more uniform. This is because the queries provided by the crowdturfing tasks need to contain a certain number of keywords to match their items, while too many words may lead to unnecessary typing errors. But for normal activities, the length of the query varies with user intent. From Figure 2(b), we observe that users view more pages to find the required item in spamming activities, which indicates that the spam items in these tasks are not so popular and ranked at a low position. Despite the use of the specific queries, crowd workers still can not find these items in the first few pages. As for time-related attributes, spamming activities' browse time and dwell time are usually longer than those of normal ones (shown in Figure 2(c) and 2(d)), due to the corresponding requests in the crowdturfing tasks.

4.2 User Analysis

We now look into the user attributes. Since we only spot normal activities in Section 3.2 and there is no effective method to spot normal users or normal items, we compare user attributes between spam users extracted in the crowdsourcing platform and other users in our dataset, and so does for item attributes. According to the user id, we collect users' information from the online shopping site during the period from March 8 to April 13, 2017.

As shown in Table 4, spam users have relatively fewer A2F, Add to Cart, purchase and item reviews. It indicates that spam users spend less time on normal online shopping activities. Since these attributes are related to users' active time on the shopping site, which varies from user to user, we calculate the ratio of Add to Cart and purchase to A2F. We can see that spam users showing a smaller consumer demand compared to other users, which indicates that these users have less purchasing power.

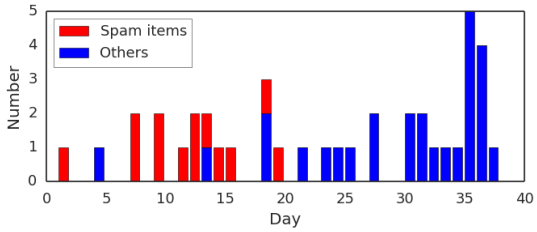
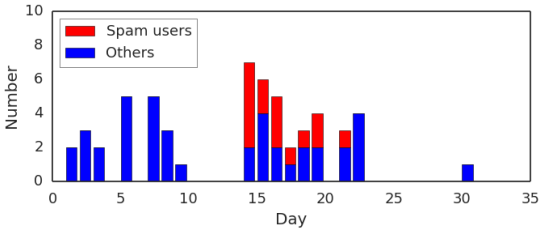
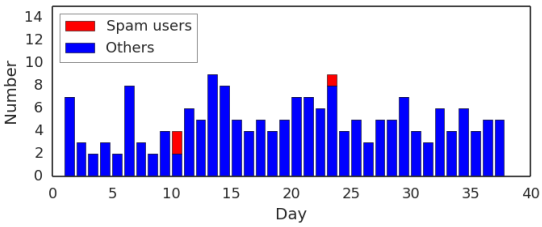


Figure 3: A2F activities of a spam user



(a) Spam item



(b) Popular item

Figure 4: A2F activities of a spam item and a popular item

We also look into the continuity of spam users' A2F activities in our dataset from a case study. As we can see in Figure 3, this spam user adds a number of spam items to his/her favorites in a continuous period of time (the first half month). It indicates that spam users tend to add spam items to their favorites (Assumption 1), and these spamming activities are continuous.

4.3 Item Analysis

As mentioned in Section 3.1, we don't extract the spam items in the tasks. Therefore, to get spam items, we extract activities performed by spam users with a spam query in the dataset, and spot the corresponding items as spam items. With 296 spam queries and 113 spam users, we spot 58 spam items in total. We compare item attributes between spam items and other items in our dataset.

As shown in Table 5, spam items have fewer A2F, add to cart, purchase and item reviews. It indicates that spam items are usually low-quality items that cannot attract normal users. Since these attributes are related to the exposure of the items, we also calculate the ratio of Add to Cart and purchase to A2F. We can see that these two attributes of spam items are much lower than those of others, indicating the lower demand for these items. In other words,

users are less likely to buy these spam items, which validates the necessary of identifying these items and avoid them being ranked at high positions in the results list.

Similarly, we look into the continuity of items' A2F activities in our dataset. Figure 4(a) shows a spam item's A2F activities. We can see that all the spam users' A2F activities are concentrated within a short period of time, which is probably the active time of the crowdturfing task. It indicates that spam items tend to be added to the favorites by spam users (Assumption 2), and these spamming activities are continuous and concentrated. Meanwhile, there are also concentrated A2F activities happen in the first 9 days. Therefore, we have reason to doubt that these activities in the first 9 days may be performed by another group of crowd workers in another crowdsourcing platform. Figure 4(b) shows A2F activities of a popular item (as mentioned in Section 3.2). The number of A2F activities is stable over time. Besides, there are some activities performed by spam users, indicating that spam users will also carry out normal A2F activities.

4.4 Summary

From the above analysis, it is clear that some behavior attributes between crowdturfing A2F activities and normal activities are asymmetric. Besides, we can find that there are certain differences between spam users/items and normal ones. We also observe that the spamming activities of spam users/items are continuous and concentrated. Based on these findings, we construct a factor graph model to detect spamming A2F activities in next Section.

5 SPAMMING A2F ACTIVITIES DETECTION

In this section, we propose a novel Activity Factor Graph Model (AFGM) to incorporate all the information about behaviors, users and items for better predicting spamming A2F activities. We first sample a part of nodes as the training set and the remaining as the test set, then our model infers each of the remaining node's probability of spam. Our goal is to train a partially labeled factor graph model.

5.1 Model Framework

Factor graph assumes that observation are cohesive with attributes and correlations. It has been successfully applied in a number of spam detection works [17, 20].

In this work, we formalized our problem into an Activity Factor Graph Model(AFGM). Figure 5 shows the graphical representation. The set of activity nodes $V = \{A_1, A_2, ..., A_N\}$ in network G is mapped to a factor node set $Y = \{y_1, y_2, ..., y_N\}$ in activity factor graph. The activities in G are partially labeled, thus Y can be divided into two subsets Y_L and Y_U, corresponding to the labeled(the training set) and unlabeled(the test set) activities respectively. Using the known factor node set in the training set, AFGM infers how likely an unknown node is to be spam. Based on the findings in Section 4, we define the following four types of factors:

- **Behavior attribute factor:** $f_b(y_i|b_i)$ represents the posterior probability of y_i, given the behavior attribute vector b_i.
- **User attribute factor:** $f_u(y_i|u_i)$ represents the posterior probability of y_i, given the user attribute vector u_i that are extracted from user U_i.
- **Item attribute factor:** $f_p(y_i|p_i)$ represents the posterior probability of y_i, given the item attribute vector p_i that are extracted from item(product) P_i.
- **Correlation factor:** Based on the finding that spam users/items' spamming activities are continuous and concentrated, we have two intuitions that (1) A2F activities performed by the same user in a small period of time may have a correlation, and (2) activities on the same item in a small period of time may have a correlation. Therefore, we have two correlation factors:
  - $g_u(y_i, C_u(y_i))$ denotes the user-based correlations between the activities, where $C_u(y_i)$ is the set of user correlated factor nodes to y_i in the graph.
  - $g_p(y_i, C_p(y_i))$ denotes the item-based correlations between the activities, where $C_p(y_i)$ is the set of item correlated factor nodes to y_i in the graph.

Given the activity network G, the formation probability of the activities in the AFGM defines as follow:

$$P(Y|G) = \frac{1}{Z}\prod_i f_b(y_i|b_i)\cdot f_u(y_i|u_i)\cdot f_p(y_i|p_i) \cdot g_u(y_i, C_u(y_i))\cdot g_p(y_i, C_p(y_i)) \tag{1}$$

where Z is the normalized factor, which sums up the formation probability $P(Y|G)$ over all the possible labels of all the activities. The objective of our model is to maximize this formation probability.

5.2 Model Inference

The factors in our model can be instantiated in different ways. Following previous work [29], we use exponential-linear functions and define the three attribute factors as

$$f_b(y_i|b_i) = exp\left\{\lambda_b^T \Phi_b(y_i, b_i)\right\} \tag{2}$$

$$f_u(y_i|u_i) = exp\left\{\lambda_u^T \Phi_u(y_i, u_i)\right\} \tag{3}$$

$$f_p(y_i|p_i) = exp\left\{\lambda_p^T \Phi_p(y_i, p_i)\right\} \tag{4}$$

where $\lambda_b, \lambda_u, \lambda_p$ are weighting vectors, and Φ_b, Φ_u, Φ_p are vectors of feature functions. Similarly, the correlation factors can be defined as

$$g_u(y_i, C_u(y_i)) = exp\left\{\sum_{y_j \in C_u(y_i)} \varphi_u^T \Theta_u(y_i, y_j)\right\} \tag{5}$$

$$g_p(y_i, C_p(y_i)) = exp\left\{\sum_{y_j \in C_p(y_i)} \varphi_p^T \Theta_p(y_i, y_j)\right\} \tag{6}$$

where φ_u, φ_p are weighting vectors, and Θ_u, Θ_p can be defined as vectors of indicator functions. Learning AFGM is to estimate a parameter configuration $\theta = (\lambda_b, \lambda_u, \lambda_p, \varphi_u, \varphi_p)$, by maximizing the formation probability $P(Y|G)$.

For presentation simplicity, we concatenate all factor functions in Eq 2-6 for a factor node y_i as

$$\mathbf{s}(y_i) = (\Phi_b(y_i,b_i)^T, \Phi_b(y_i, u_i)^T, \Phi_b(y_i, p_i)^T, \sum_{y_j \in C_u(y_i)} \Theta_u(y_i, y_j)^T, \sum_{y_j \in C_p(y_i)} \Theta_p(y_i, y_j)^T)^T \tag{7}$$

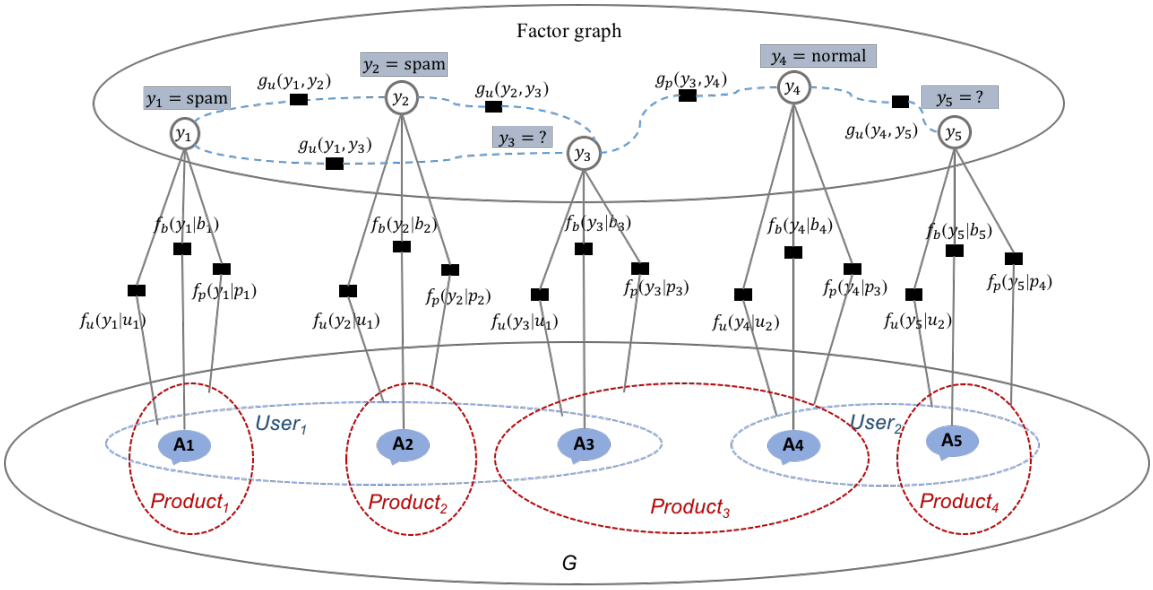


Figure 5: Graphical representation of the AFGM

Then, the formation probability in Eq 1 can be written as

$$\begin{aligned} P(Y|G) &= \frac{1}{Z}\prod_i exp\left\{\theta^T \mathbf{s}(y_i)\right\} \\ &= \frac{1}{Z} exp\left\{\theta^T \sum_i \mathbf{s}(y_i)\right\} \\ &= \frac{1}{Z} exp\left\{\theta^T \mathbf{S}\right\} \end{aligned} \tag{8}$$

where $\mathbf{s}$ is the aggregation of factor functions over all factor nodes, i.e. $\mathbf{S} = \sum_i \mathbf{s}(y_i)$.

Since the factor node set Y is partially labeled, to calculate the formation probability, we define $Y|Y_L$ as a labeling configuration given the known labels Y_L. Further, we can define the log-likelihood objective function as

$$\begin{aligned} O(\theta) &= log(\sum_{Y|Y_L} P(Y|G)) \\ &= log(\sum_{Y|Y_L} \frac{1}{Z} exp\left\{\theta^T \mathbf{S}\right\}) \\ &= log(\sum_{Y|Y_L} exp\left\{\theta^T \mathbf{S}\right\}) - log(Z) \\ &= log(\sum_{Y|Y_L} exp\left\{\theta^T \mathbf{S}\right\}) - log(\sum_{Y} exp\left\{\theta^T \mathbf{S}\right\}) \end{aligned} \tag{9}$$

We adopt a gradient descent algorithm [29] to solve the log-likelihood objective function. The gradient for each parameter θ is

$$\begin{aligned} \frac{\partial O(\theta)}{\partial \theta} &= \frac{\partial (log(\sum_{Y|Y_L} exp\left\{\theta^T \mathbf{S}\right\}) - log(\sum_Y exp\left\{\theta^T \mathbf{S}\right\}))}{\partial \theta} \\ &= \frac{\sum_{Y|Y_L} exp\left\{\theta^T \mathbf{S}\right\} \cdot \mathbf{S}}{\sum_{Y|Y_L} exp\left\{\theta^T \mathbf{S}\right\}} - \frac{\sum_Y exp\left\{\theta^T \mathbf{S}\right\} \cdot \mathbf{S}}{\sum_Y exp\left\{\theta^T \mathbf{S}\right\}} \\ &= E_{Y|Y_L,G}(\mathbf{S}) - E_{Y|G}(\mathbf{S}) \end{aligned} \tag{10}$$

where $E_{Y|Y_L,G}(\mathbf{S})$ represents the expectation of $\mathbf{S}$ given the known label Y_L, and $E_{Y|G}(\mathbf{S})$ represents the expectation of $\mathbf{S}$ over all the possible labels.

Since it is intractable to calculate $E_{Y|Y_L,G}(\mathbf{S})$ and $E_{Y|G}(\mathbf{S})$, we use loopy belief propagation (LBP) algorithm [22] to achieve an near-optimal solution. Specifically, we perform LBP process twice in each iteration, one time for estimating the marginal probability of unknown nodes (i.e. $p(y|Y_L, G), y \in Y_U$) and the other for the marginal probability of all nodes (i.e. $p(y|G)$). With the marginal probabilities, $E_{Y|Y_L,G}(\mathbf{S})$ and $E_{Y|G}(\mathbf{S})$ can be obtained by summing over all corresponding nodes. Finally with the gradient, we update each parameter with a learning rate α:

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla_\theta \tag{11}$$

Based on learned parameters θ, we again use LBP algorithm to calculate the marginal probability of each factor node in the test set Y_U. Then, the marginal probability is taken as the prediction confidence, i.e., the activity node's probability of spam or normal.

6 EXPERIMENTS

6.1 Experimental Setup

Feature. According to Section 4 and Section 5, we give our features used in factor construction. All the attribute features involved are listed in Table 6. For both user and item attribute features, we only use two ratio attributes because we think quantitative attributes are biased for different users or items as mentioned before, and these two ratios can better reflect the quality of users and items. It is worth noting that the first four are binary attributes and the rest are continuous-valued attributes. To simplify representation for continuous-valued attributes, we discretize these continuous attribute space over some number of H intervals, each H is tuned according to the corresponding attribute distribution [27]. Thus, each continuous-valued attribute can take on values from $\{1...H\}$, i.e. convert to one of H attributes.

Table 6: Attribute Features List

Cat.	No	Description
Behavior attribute	1	Add to Cart
	2	Rank the results
	3	With previous clicks
	4	On weekends
	5	Query length
	6	Page number
	7	Browse time
	8	Dwell time
User attribute	9	Add to Cart / A2F
	10	Purchase / A2F
Item attribute	11	Add to Cart / A2F
	12	Purchase / A2F

As for correlation factors, since spam users/items' spamming activities are continuous and concentrated, we consider that continuous N_u/N_p activities of same user/item have a strong relationship. Therefore, for the activity node A_i, we add factor nodes of previous $N_u - 1$ activities performed by the same user into $C_u(y_i)$. Similarly, for item-based correlation, we add factor nodes of previous $N_p - 1$ activities on the same item into $C_p(y_i)$. In our work, we set both N_u and N_p to 3.

Dataset. As mentioned in Section 3, our goal is to detect spamming A2F activities in suspicious log records. Due to the fact that spamming activities are very similar to normal ones, we can not label these log records as "Spam" or "Normal" manually. Therefore, it is difficult to evaluate the performance of our algorithm. To solve this problem, we randomly select 80% of the spam log records (about 4K records), together with all the normal log records as the training set (Y_L), and leave the rest 20% of the spam log records for the evaluation. We use the five-fold cross validation to split spam records and examine the performance of detection model.

For the suspicious log records, we extract the items with no more than 10 records and remove their records, because we consider that the low number of related log records in the dataset indicates that these items are unlikely to be spam items. Even if they are spam items, their harm to the online shopping site is negligible due to the low number of the spamming activities. In this way, we removed 2,495,066 log records. Therefore, our test set (Y_U) consists of the remaining 1,615,630 suspicious records and 20% of the spam records.

6.2 Baseline Methods

Since we are among the first to investigate the spamming A2F activities in online shopping, there is a lack of effective detection models for this problem. Therefore, we compare our proposed model (AFGM) with three widely-used methods for classification in many fields. Meanwhile, to investigate our proposed features, we also add some simplified models as our baseline methods. Details are given below:

- **Support Vector Machine:** Given all behavior attribute features, user attribute features and item attribute features, we can represent each log record with an attribute vector and train a Support Vector Machine (SVM) classification model, based on the training set. With the learned model, we can get the spam probability of each log record in our test set.
- **Logistic Regression Classifiers:** Similarly, we train a Logistic Regression (LR) models with all the attribute features. Then we use our trained LR classifier to infer the spam probability of each log record in the test set.
- **Random Forest Classifiers:** We also train a Random Forest Classifiers (RF) with all the attribute features. We use our learned RF model to infer unlabeled records and compare performance with our approach.
- **Bipartite Graph** We take the idea of label propagation algorithm [15] to build a "user-item" bipartite graph based on the two assumptions mentioned before. In the bipartite graph, there is an unweighted edge between a user and its collected item, i.e. each edge means a A2F activity. The spam items mentioned in Section 4.3 are used as the labeled seed to drive the algorithm. The spam probability of each log record is calculated by the spam probability of the user and the item in the graph.
- **AFGM − UP:** Comparing to AFGM, it removes user attribute factors and item attribute factors, which only use features extracted from individual log records and their correlations. We construct this model to illustrate the necessity of user and item attributes.
- **AFGM − C_u:** It uses the proposed activity factor graph model, but the user-based correlations between activities are not integrated in it. Through this method, we want to analyze whether user-based correlations are useful for our model.
- **AFGM − C_p:** Similarly, to show whether item-based correlations is useful for our model, the item-based correlations are not used in this approach compared to AFGM.

6.3 Evaluation Metrics

Due to the difficulty of manual annotation for the test set, we use two metrics to evaluate our detection model AFGM and compare AFGM with baseline methods.

As mentioned in Section 6.1, the test set (Y_U) contains 20% of ground truth data. Considering the fact that a discriminative detection model should identify spam records, we focus on the spam probabilities of these ground truth data. We first sort all the activity log records in the test set by their spam probabilities given by the detection model. Then, we calculate the recall rate at top 1% , i.e.

$$Recall@Top\ 1\% = \frac{Number\ of\ spam\ records\ in\ the\ top\ 1\%\ records}{Number\ of\ spam\ records} \tag{12}$$

We do not use precision rate because suspicious log records in the test set have high probabilities of spam. Our goal is to detect spamming A2F activities in these records. Thus, it is unreasonable to regard these records as non-spam records when calculating precision rate. Meanwhile, we also use AUC metric to see whether the detection model can give these spam log records higher spam probabilities.

6.4 Experimental Results

Table 7 shows the performance of spam detection with different methods on our evaluation metrics. The best performance has been highlighted in bold.

As we can see, LR model achieves the worst performance on Recall@Top 1% (0.078), followed by SVM (0.121) and RF (0.166). The AUCs of these three baseline methods are around 0.7, which means

Table 7: Comparisons between our methods and baselines

	Recall@Top 1%	AUC
LR	0.078	0.689
SVM	0.121	0.682
RF	0.166	0.706
BG	0.247	0.699
AFGM − UP	0.580	0.899
AFGM − C_u	0.448	0.877
AFGM − C_p	0.334	0.757
AFGM	**0.617**	**0.903**

these widely-used methods are not appropriate for this problem. BG model achieves a better performance on Recall@Top 1% (0.247), which indicates correlations are more important than attributes in this detection task.

It can be easily found that all our models perform better than 4 baselines, and AFGM achieves the best performance on both Recall@Top 1% (0.617) and AUC (0.903). By comparing AFGM − C_u and AFGM − C_p with AFGM, we find that removing user-based or item-based correlations will decrease the performance to some extent. And it can explain why LR, SVM and RF models achieve bad performance because they do not use these correlations. Besides, AFGM − C_u performs better than AFGM − C_p, which indicates that item-based correlations are important than user-based correlations. It is reasonable because crowd workers are normal users for most time and will also carry out some normal A2F activities by themselves, while most activities for spam items are spamming. By comparing AFGM − UP and AFGM, we can find that individual log record contains enough information (including behavior attributes and their correlation) to detect spamming activities, while user and item attributes can further enhance performance.

Figure 6 shows the detection performance of different proposed models measured by recall rate at top k%. We can observe that nearly 80% of spamming activities can be acquired in the top 10% log records of the test set with AFGM and AFGM − UP. AFGM − UP achieves a very close performance with AFGM. This could be because the correlation factors contain enough information to cover the user and item attribute factors. Therefore, more effective user and item attributes may be needed to improve AFGM. The gap between AFGM − C_p and other models are large, which indicates that item-based correlations are relatively more important to detect spamming activities.

7 DISCUSSION

According to Table 7 and Figure 6, more than 60% of the ground truth spam records can be found in the top 1% test records. As mentioned in Section 3, each behavior log record has a certain probability of spam. Therefore, besides spam log records, online shopping site should pay attention to the records with high spam probability calculated by AFGM. For example, we can warn or punish the users and items, which appear in the top 1% test records twice or more. Or we can calculate a discount weight for spam items' popularities or remove spam records directly. However, it will have a worse effect if we regard a normal user/item as spam one and carry out the punishment. Therefore, more effort is needed to determine whether a user/item is really spam.

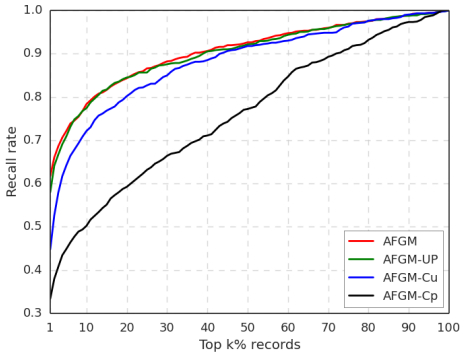


Figure 6: Comparisons of different proposed models measured by Recall@Top k%

One limitation of our work is that we regard interaction sessions on items with more than 500 records in the dataset as normal log records. This may ignore some niche items that also contain a number of normal A2F activities. Therefore, we need to further improve our annotation method to acquire a more complete labeled dataset.

8 CONCLUSIONS

In this paper, we investigate the crowdturfing "Add to Favorites" activities in online shopping. To look into this kind of newly-appeared malicious activities and make the detection, we create a dataset through simultaneously locating a number of crowdturfing tasks and collecting user behavior log from online shopping activities. With a comprehensive analysis of some ground truth spamming activities, we find some differences between spamming activities and normal ones in terms of behavior, user and item.

Given various extracted attributes (behavior-level, user-level and item-level) and correlations (user-based and item-based), we propose an activity factor graph model (AFGM) to infer whether a A2F activity is spamming. Experimental results on our dataset validate the effectiveness of the proposed model. More than 60% of the spam records can be found in the top 1% records of the test set. By comparing with some simplified models, we show that the features we use are helpful to the detection, while item-based correlations are most important except behavior attributes.

As future work, it is important to study how to detect spam users and spam items based on our result. Since the user and item attributes have limited contributions to our model, we need to find more effective indicators, which may also help us to evaluate different detection methods. Besides, the current model is built for detection spamming activities in a period of time. A timely detection model is also an interesting future research direction.

9 ACKNOWLEDGEMENTS

We thank Mr. Haifeng Lu for providing very useful suggestions for this paper. This work is supported by Natural Science Foundation of China (Grant No. 61622208, 61732008, 61532011), National Key Basic Research Program (2015CB358700) and Alibaba Group through Alibaba Innovative Research (AIR) Program.

REFERENCES

[1] Prudhvi Ratna Badri Satya, Kyumin Lee, Dongwon Lee, Thanh Tran, and Jason Jiasheng Zhang. 2016. Uncovering fake likers in online social networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2365–2370.

[2] Antoaneta Baltadzhieva. 2015. Question Quality in Community Question Answering Forums:a survey. *Acm Sigkdd Explorations Newsletter* 17, 1 (2015), 8–13.

[3] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. 33, 9 (2013), 119–130.

[4] Cheng Cao, James Caverlee, Kyumin Lee, Hancheng Ge, and Jinwook Chung. 2015. Organic or Organized?: Exploring URL Sharing Behavior. In *ACM International on Conference on Information and Knowledge Management*. 513–522.

[5] Cheng Chen, Kui Wu, V Srinivasan, and K Bharadwaj, R. 2012. The best answers? Think twice: Online detection of commercial campaigns in the CQA forums. (2012), 458–465.

[6] Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Squicciarini. 2015. Uncovering Crowdsourced Manipulation of Online Reviews. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 233–242.

[7] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012. Distributional Footprints of Deceptive Product Reviews.

[8] David Mandell Freeman. 2017. Can You Spot the Fakes?: On the Limitations of User Feedback in Online Social Networks. In *International Conference on World Wide Web*. 1093–1102.

[9] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 219–230.

[10] Kyumin Lee, James Caverlee, Zhiyuan Cheng, and Daniel Z. Sui. 2014. Campaign extraction from social media. *Acm Transactions on Intelligent Systems & Technology* 5, 1 (2014), 1–28.

[11] Kyumin Lee, Brian David Eoff, and James Caverlee. 2006. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. (2006).

[12] Kyumin Lee, Prithivi Tamilarasan, and James Caverlee. 2013. Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media.

[13] Beibei Li, Anindya Ghose, and Panagiotis G Ipeirotis. 2011. Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*. ACM, 327–336.

[14] Baichuan Li, Tan Jin, Michael R. Lyu, Irwin King, and Barley Mak. 2012. Analyzing and predicting question quality in community question answering services. In *International Conference on World Wide Web*. 775–782.

[15] Xin Li, Yiqun Liu, Min Zhang, Shaoping Ma, Xuan Zhu, and Jiashen Sun. 2015. Detecting Promotion Campaigns in Community Question Answering.. In *IJCAI*. 2348–2354.

[16] Ee Peng Lim, Viet An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *ACM International Conference on Information and Knowledge Management*. 939–948.

[17] Yuli Liu, Yiqun Liu, Ke Zhou, Min Zhang, and Shaoping Ma. 2017. Detecting Collusive Spamming Activities in Community Question Answering. In *The International Conference*. 1073–1082.

[18] Bo Long, Jiang Bian, Anlei Dong, and Yi Chang. 2012. Enhancing product search by best-selling prediction in e-commerce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2479–2482.

[19] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. 2010. Exploiting social context for review quality prediction. (2010), 691–700.

[20] Yuqing Lu, Lei Zhang, Yudong Xiao, and Yangguang Li. 2013. Simultaneously detecting fake reviews and review spammers using factor graph model. In *ACM Web Science Conference*. 225–233.

[21] Arjun Mukherjee, Bing Liu, and Natalie Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *International Conference on World Wide Web*. 191–200.

[22] Kevin Murphy, Yair Weiss, and Michael I. Jordan. 2013. Loopy Belief Propagation for Approximate Inference: An Empirical Study. (2013), 467–475.

[23] Myle Ott, Claire Cardie, and Jeff Hancock. 2012. Estimating the prevalence of deception in online review communities. (2012), 201–210.

[24] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. 1 (2011), 309–319.

[25] Vlad Sandulescu and Martin Ester. 2015. Detecting Singleton Review Spammers Using Semantic Similarity. (2015).

[26] Chirag Shah and Jefferey Pomerantz. 2010. Evaluating and predicting answer quality in community QA. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 411–418.

[27] Neil Shah. 2017. FLOCK: Combating Astroturfing on Livestreaming Platforms. In *The International Conference*. 1083–1091.

[28] Shankar S Siva. 2014. Survey Paper for WARNINGBIRD: Detecting Suspicious URLs in Twitter Stream. *International Journal of Advanced Trends in Computer Science & Engineering* 3, 5 (2014), 2319–7242.

[29] Wenbin Tang, Honglei Zhuang, and Jie Tang. 2011. Learning to Infer Social Ties in Large Networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 381–397.

[30] Tian Tian, Jun Zhu, Fen Xia, Xin Zhuang, and Tong Zhang. 2015. Crowd fraud detection in internet advertising. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1100–1110.

[31] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. 2012. Serf and turf: crowdturfing for fun and profit. In *International Conference on World Wide Web*. 679–688.

[32] Fangzhao Wu, Jinyun Shu, Yongfeng Huang, and Zhigang Yuan. 2015. Social spammer and spam message co-detection in microblogging with social context regularization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1601–1610.

[33] Chang Xu and Jie Zhang. 2016. Towards Collusive Fraud Detection in Online Reviews. In *IEEE International Conference on Data Mining*. 1051–1056.

[34] Chang Xu, Jie Zhang, Kuiyu Chang, and Chong Long. 2013. Uncovering collusive spammers in chinese review websites. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 979–988.

[35] Junting Ye and Leman Akoglu. 2015. Discovering Opinion Spammer Groups by Network Footprints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 267–282.

[36] Kyung Hyan Yoo and Ulrike Gretzel. 2009. Comparison of Deceptive and Truthful Travel Reviews. In *Information and Communication Technologies in Tourism, Enter 2009, Proceedings of the International Conference in Amsterdam, the Netherlands*. 37–47.