Proceedings of the Thirteenth International AAAI Conference on Web and Social Media (ICWSM 2019)

TOmCAT: Target-Oriented Crowd Review Attacks and Countermeasures

Parisa Kaghazgaran, Majid Alfifi, James Caverlee
Texas A&M University
College Station, TX 77843
{kaghazgaran, alfifi, caverlee}@tamu.edu

Abstract

Online platforms like Amazon, Yelp, and Regulations.gov give a voice to masses of users through reviews, comments, and ratings. However, this crowd-based feedback is susceptible to manipulation. To tackle this problem, most previous efforts have only indirectly sought to uncover targets of attacks by focusing on manipulation at the review or user level. Instead, this paper focuses on the challenge of countering target-oriented crowd attacks. We introduce a unique ground truth dataset of Amazon products that have been targeted for attack and identify two target-oriented attack patterns: (i) *promotion attacks* and (ii) *restoration attacks*. With these attacks in mind, we propose the TOmCAT detection framework based only on the timing and sequencing of product ratings. Although TOmCAT succeeds in uncovering targets of manipulation with high accuracy by addressing existing attacks, strategic attackers potentially can create hard-to-detect behavioral patterns by undermining timing-based footprints. Hence, we further propose a complementary approach to TOmCAT called TOmCATSeq which is resistant against strategic manipulation.

1 Introduction

User review aggregators like Amazon, Yelp and Netflix play a central role in forming our decision to buy a product or use a service (Duan, Gu, and Whinston 2008). In addition, policy makers rely on online platforms like Regulations.gov and FCC.gov as a means for citizens to voice their opinions about public policy issues. Alas, showcasing the opinions of fellow users has a dark side – these reviews and comments are often targets of manipulation (Jindal and Liu 2008; Post 2018; Kumar and Shah 2018). For example, a recent study of comments posted to the FCC about repealing net neutrality found that over one million pro-repeal comments were likely faked; in contrast, a majority of the legitimate comments were in favor of keeping net neutrality (Kao 2017). These attacks on online reviews can sway ratings of products, manipulate opinions and perceived support of key issues, and degrade our trust in online platforms.

Many previous efforts have typically focused on either detecting *fake review writers* (Hooi et al. 2016; Kumar et al. 2018; Shin et al. 2017) or *fake reviews* (Mukherjee et al. 2013b; Ott, Cardie, and Hancock 2013). While these fake review writers and fake reviews serve as a building block of an attack, the ultimate goal is often to manipulate a specific *target*. For example, the target of an attack could be a product (e.g., an item on Amazon), place (e.g., a restaurant on Yelp), service (e.g., web hosting service), or issue (e.g., a net neutrality post on FCC.gov). Knowing which products (or services, etc.) are targets of an attack, we can deploy more resources to defend the target from ongoing threats (e.g., require additional user verification or enlist more human moderators) and develop more robust defensive countermeasures for future threats (e.g., by learning patterns of the types of issues targeted).

And yet, it has traditionally been challenging to identify which products (or places, services, issues) are actually targets of attacks without a gold standard dataset. In this paper, we build a unique ground truth of *Amazon products* that have been targeted by crowd review manipulation attacks.

The main aim of this paper is to study the challenge of countering *target-oriented crowd attacks* in order to detect the target of attacks, in a complementary direction to approaches that focus on building blocks of attacks like detecting fake reviews or fake review writers. Concretely, our main contributions are:

1. We introduce the review manipulation dataset of Amazon *products* (Section 3.1) and this dataset is available for research purposes upon request.
2. We identify two target-oriented attack patterns over this dataset: (i) *promotion attacks*, wherein a crowd seeks to manipulate the product rating of a new product; and (ii) *restoration attacks*, wherein a crowd seeks to counteract a low rating from a legitimate reviewer. (Sections 3.2)
3. With these attacks in mind, we develop a **T**arget-**O**riented **C**rowd **AT**tack detection framework called TOmCAT comprising two components: (i) it first formulates crowd attack footprints into a suite of features based only on timing and sequence of product ratings; (ii) it then embeds attack footprints in a 3-layer neural network, where we find a good success in uncovering target products on our original Amazon dataset. (Sections 4.1, 4.2 and 4.3)

4. We show that TOmCAT outperforms six unsupervised and supervised baselines that originally were proposed to detect manipulation at the user/reviewer level. We find that review manipulation behaves differently at user and product levels. (Section 4.4)
5. We validate TOmCAT over three additional domains – Yelp, the App Store, and an alternative Amazon dataset – where we find that TOmCAT can effectively detect manipulation patterns in other domains. (Section 4.5)
6. Although TOmCAT can uncover target products with high accuracy by addressing existing attacks, strategic attackers can potentially create hard-to-detect behavioral patterns by undermining timing-based footprints. Inspired by recent advances in recurrent neural networks, we further propose a complementary approach to TOmCAT called TOmCATSeq.[1] We believe this is the first work to leverage RNN models on rating sequences for review manipulation detection. The initial evaluation of TOmCATSeq shows promising results against strategic manipulation and opens new doors for future study. (Section 5)

Note that we restrict our approach from taking advantage of historical reviewer behavior, reviews, and network properties, to emulate a scenario in which a powerful attacker can nullify the discriminatory power of these signals. For example, reviewers may have little or no history, say by using multiple user accounts to write fake reviews, degrading the impact of network characteristics on uncovering manipulation. Also, fake reviews may also imitate real (non-fake) reviews, limiting the power of linguistic detection. And as advances in AI continue, machine generated fake reviews will increasingly be difficult to detect (Yao et al. 2017).

2 Related Work

Many methods have been proposed to identify individual fake reviews. These methods often focus on the content of reviews themselves, from the perspective of bag of words features (Ott, Cardie, and Hancock 2013; Jindal and Liu 2008; Sandulescu and Ester 2015) and structural features (Li et al. 2011; Piskorski, Sydow, and Weiss 2008; Li et al. 2014; Harris 2012) like length of reviews, part-of-speech-tagging, and proportion of certain pronouns.

Another direction aims to identify users who engage in manipulation tasks. This has been extensively studied by researchers mainly by leveraging behavioral signals such as skewed rating distributions (Hooi et al. 2016; Shah et al. 2016) and dense inter-arrival times between successive reviews (Shah et al. 2016; Hooi et al. 2016; Ye, Kumar, and Akoglu 2016).

In a related direction, some efforts try to find groups of fraudulent reviewers, often by finding synchronized behaviors. One approach is graph-based, where users are nodes and their relationships are edges (Jiang et al. 2014; Prakash et al. 2010; Kumar et al. 2018; Akoglu, Chandy, and Faloutsos 2013). An embedding-based method is proposed in (Kaghazgaran, Caverlee, and Squicciarini 2018) to identify fraudulent users who are distant in a local graph but close in an embedding space. A separate direction detects dense blocks in a ratings matrix (Shin et al. 2017), to find clusters of coordinating raters.

Finally, crowd attacks have been explored in Facebook (Cao et al. 2014) with a SynchroTrap system to deal with coordinated Likes on selected pages and in Twitter (Viswanath et al. 2015) with the Stamper method to detect targeted hashtags. These recent trends in similar domains for detecting malicious crowds motivate us to tackle this problem in online review platforms wherein a crowd seeks to compromise the rating of a product or service.

[1]The naming is inspired by sequencing nature of RNN models

Table 1: Summary of Our Dataset

	# Products	# Reviews
Primary Targets	533	33 k
Secondary Targets	3,467	2.6 M
Randomly Sampled	4,000	0.7 M

3 Target-Oriented Crowd Attacks

We say that a coordinated attack that aims to manipulate a specific target is a *target-oriented crowd attack*. Such coordinated attacks have historically been difficult to identify in order to build a solid ground truth and evaluate corresponding countermeasures. How can we be sure that a product has actually been targeted?

3.1 Building Ground Truth

Here, we adopt an approach that samples evidence of manipulation launched from crowdsourcing platforms where a paymaster has tasked crowd workers to write a few fake reviews on a specific product on Amazon (Kaghazgaran, Caverlee, and Alfifi 2017). We refer to such products as *target products*. Similar efforts (Wang et al. 2012; Song, Lee, and Kim 2015; Lee, Tamilarasan, and Caverlee 2013) have been used to uncover other crowd-based attacks. In particular, we monitored the crowdsourcing platform RapidWorkers beginning in July 2016 for evidence of Amazon-related attacks. In contrast to our previous work that focuses on fake review writers and study their behaviors (Kaghazgaran, Caverlee, and Squicciarini 2018), this work provides a new dataset of products that have been targeted for attack. The previous work is based on a seed set of 300 products to label fake review writers. Here, we extend the initial seed set and develop an expansion methodology to gain a rich set of target products.

Primary Target Products. In total, we were able to identify 900 tasks targeting 533 unique products. We call these initial products the *primary target products*. By linking those products to Amazon, we then collected their associated reviews – that is ~33k reviews. Next, we explored the activity history of ~14k reviewers and crawled their reviews, i.e. ~580k reviews.

Identifying Suspicious Reviewers. Not all reviewers on a target product are suspicious. To identify suspicious reviewers we use a similar methodology to that introduced in (Jindal and Liu 2008) in which users with a certain fraction

⋆ *DON'T DO IT*, **February 15, 2018**
This is an inferior product. They are miniature, at least half the size they should be. I don't know anyone with eyes this small....
⋆⋆⋆⋆⋆ *Very Very good Product*, **February 16, 2018**
Very Nice product. Love it. Looking very cute. Best deal at best price. Very must satisfied with product and service.
⋆⋆⋆⋆⋆ *They really last for a good ..*, **February 17, 2018**
These lashes are so easy to use and they really last for a good while. There's a learning curve involved to apply them, ...
⋆⋆⋆⋆⋆ *Eyelashes look very natural*, **February 17, 2018**
I bought it for my girlfriend and she loves it. its a little difficult to learn how is the right way to put it on, but when you got .....
⋆⋆⋆⋆⋆ *Looks extremely real!*, **February 17, 2018**
I bought this for my girlfriend and I told her to test whether i could tell if they are her real eyelashes for fake one, but .....

Figure 1: Restoration Attack Example

⋆⋆⋆⋆⋆ *best shower head I have gotten*, **August 30, 2017**
This multi-functional head helps us relax ourselves during the bath and gives the feel of taking a spa. I am using it everyday...
⋆⋆⋆⋆⋆ *Best product...*, **August 30, 2017**
The design of the product is amazing and I absolutely love the water flow combinations.The water flow settings can be...
⋆⋆⋆⋆⋆ *very high quality product*, **August 30, 2017**
The water flow settings can be changed easily and a very high quality product.It has a setting for a very full vigorous...
⋆⋆⋆⋆⋆ *Multifunction shower.....All in one*, **August 31, 2017**
Multifunction shower is very useful because Three Settings Water Flow Control For Your Pleasure. In this product ...
⋆ *Don't waste your Money*, **March 10, 2018**
Very cheap quality. After a few days, it became loose and detached from the hose. The plumbing tape even didn't help..

Figure 2: Promotion Attack Example

of duplicate or semi-duplicate reviews are labeled as spammers. The authors showed that a spam reviewer may use duplicate reviews on different products, rather than different spammers duplicating a review on the same product. We calculate the Jaccard function between each pair of reviews written by a reviewer and take the maximum value as its *Self-Similarity Score*. Formally, the self-similarity score of reviewer v with review collection of R_v is defined as:

$$SSS(v) = max\{Jaccard(r_i, r_j) | \forall r_i, r_j \in R_v \wedge i \neq j\}$$

$$Jaccard(r_i, r_j) = \frac{|r_i \cap r_j|}{|r_i \cup r_j|} \times 100$$

We found 9,659 suspicious reviewers.

Secondary Target Products. To expand our dataset and gain a richer set of target products, we explore the products which received reviews from suspicious reviewers. To be conservative, we adjust a minimum threshold on the number of reviews generated by suspicious reviewers. In other words, we pull out products with ($\geq n$) reviews from these reviewers and label them as targets. Through our expansion methodology, we were able to identify 3,467 products which we call *secondary target products*. Crawling reviews associated with secondary target products, a corpus of 2.6M reviews is collected. In our experiments, we conservatively set the threshold such that target products were targeted by many suspicious reviewers (which helps mitigate any errors in our original labeling of suspicious reviewers). Varying n from 5 to 10 changes the number of products from ~7,500 to ~1,700. To make a trade-off between the choice of n and the number of products with reviews in our expanded dataset, we set n to be 7.

Randomly Sampled Products. Since our proposed detection model is a supervised approach, we sample about ~4,000 products randomly from the Amazon dataset introduced in (McAuley et al. 2015) with 0.7M reviews. A summary of the dataset can be found in Table 1.

3.2 Dataset Analysis

This section provides an analysis into crowd review manipulation attacks in our dataset.

Crowd Attack Characteristics: We observe that these crowd attacks demonstrate the following characteristics:

Relatively small campaign size. A group of workers who target a specific product form a crowd campaign. A majority of the campaigns are small, soliciting between 5 to 10 reviews in total. This suggests that crowd campaigns do not leave obvious patterns of synchronized behavior that could aid in their detection.

High-quality reviews. Our fake reviews demonstrate proper grammar and other evidence of being generated by actual people (and not bots), meaning that existing methods that rely on signals of poor review quality may be ill-suited to uncover such fake reviews.

Non-duplicate reviews. Duplicate reviews on the same product are not reimbursed by paymasters. This provides more evidence that these reviews may appear legitimate, and hence be challenging to detect through content-based methods. Note that suspicious reviewers might still pollute different products with similar reviews.

Together, these observations suggest that traditional text-based and NLP-based techniques may face challenges, as well as techniques that rely on clearly anomalous behavior (e.g., dozens of positive reviews arriving in a few minutes).

Crowd Attack Types: In our dataset, we identify two prominent types of crowd attacks:

Restoration Attacks: If a product receives a low-rate review (1 or 2 stars in a 5-star rating system) it might be targeted by a crowd attack with highly positive reviews to help restore the overall rating. Figure 1 shows an example of this attack wherein the target product is first rated low by a 1-star review and then receives a series of 5-star reviews.

Promotion Attacks: In other cases, crowd attacks target newborn products, i.e. immediately after the product is first introduced to Amazon. These early fake reviews aim to promote the product and encourage actual consumers to make a purchase. Figure 2 shows an example of such a scenario wherein the review thread is initiated by several fake reviews. However, it receives a low star review a few months later presumably from a true customer.

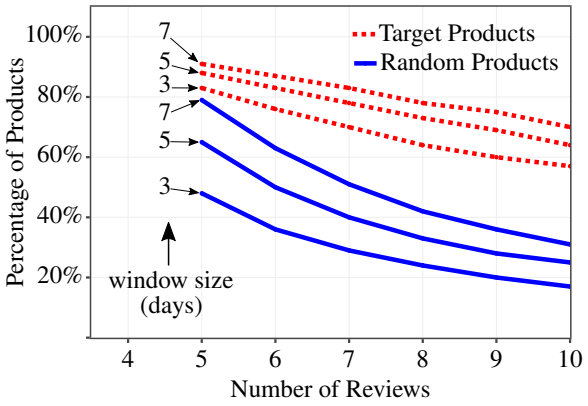

Figure 3: Target products tend to receive higher number of reviews in short time intervals.

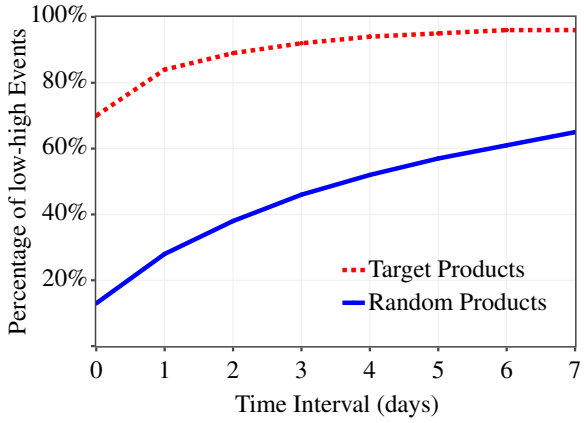

Figure 4: Target products react to low-rate review faster

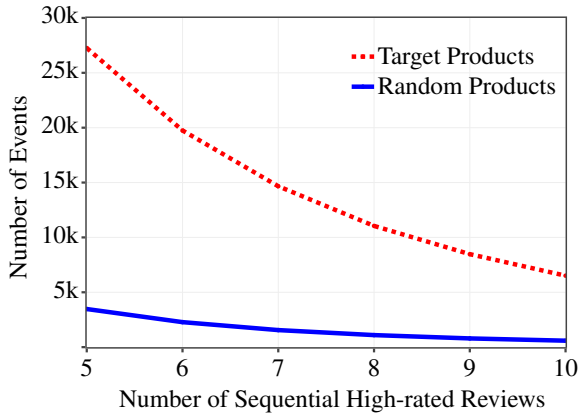

Figure 5: Target products tend to have more series of high-rating reviews immediately following a low-rating review.

Initial Observations: Given crowd attacks characteristics and types, we analyze the rating behavior of target products versus randomly selected products from three dimensions. Products are described by their time-series ratings. More formally, for each product p, we have an ordered sequence of ratings as $R_p = (r_1, .., r_n)$ where r_i happens earlier than r_{i+1}.

1. Dense Review Behavior: First, we investigate how many reviews may turn up during a specific window of time w under the two classes of products. By sliding w over a sequence of reviews, we measure the number of reviews that are written in this interval. Referring to Figure 1 which gives an example of restoration attack, fake reviews were written within a time window of 3 days. Therefore, we set the value of w to be 3, 5 and 7 days. Also, since we observe a campaign size n is in the range of 5 to 10 reviews, we examine what portion of products receive 5 to 10 reviews within w days. Figure 3 summarizes the review behavior across different values of w and n. Interestingly, a large portion of target products demonstrate dense review behavior. For example, 78% of target products have received 7 reviews within 5 days while only 40% of random products display such behavior.

2. Low High Rate Behavior: The purpose of restoration attacks is to rebuild the trust in a product that has been shaken by a negative review. Thus, we investigate low-high rate events, e.g., a 5-star review showing up immediately after a 1/2-star review. In total, we found $111,870$ and $29,205$ number of such events in target and random products respectively. That is, target products are almost four times more vulnerable to this event. However, the existence of this kind of event is not a strong indicator of anomalous behavior as it could happen naturally due to consumers with different tastes evaluating a single product significantly differently (Hu, Pavlou, and Zhang 2006).

To control for this, we measure how fast different products react to a negative review by gauging the inter-arrival time between sequential low and high ratings. Figure 4 shows what portion of these events happens in less than a specific time window w. We set w to be between 0 and 7 days. Interestingly, 71% of events occur in the same day as the negative review is written in target products while in random products, only 13% of such events happen in the same day.

Rough Estimation of Attack Types: 3,258 products experience low-high rate events. Considering same day occurrence of such event as baseline, we can say 2,660 out of ,3467 target products are targeted by restoration attacks. Similarly, 211 of the target products do not experience low-high rate events where we can say they are only targeted by promotion attacks.

3. Sequential High Ratings Behavior: In this analysis, we count the number of 5-star reviews immediately following a low-rated review. We set the size of these blocks of 5-star reviews to be between 5 and 10 with respect to the crowd campaign size. We can see from Figure 5 that the existence of such blocks in target products is about 5 times more likely than among their randomly selected peers.

4 Proposed TOmCAT Model

Inspired by these findings, we propose in this section the **T**arget-**O**riented **C**rowd **AT**tack detection framework (TOmCAT for short). The key intuition is to model targets based only on the timing and sequencing of product ratings, without access to historical reviewer behavior, reviews, and network properties. We introduce here the overall framework and a series of crowd attack footprints.

TOmCAT Structure. It is based on a neural network with three fully-connected layers. The input layer is fed attack footprints (as we describe next). The output layer has a single unit with labels 0 or 1 for each class of products (target vs. non-target). We use $ReLU(x) = max(x, 0)$ as the activation function in the hidden layers, a common choice in the literature (LeCun, Bengio, and Hinton 2015). The activation function for the output layer is a Sigmoid function which represents the classification result. Further, we adopt standard L2 regularization and gradient descent optimization.

We formulate two types of crowd footprints: *Micro features*, wherein rating behavior is modeled for a given product; and *Macro features*, wherein the deviation of rating behavior from a base-model is measured.

4.1 Micro Features

Micro features codify rating patterns on individual products as follows:

Speed of Low-High Rate Events (SLH): In restoration attacks, we assume that target products receive high rate reviews faster to facilitate their rating recovery in the aftermath of a negative review. This feature is modeled as the average of inter-arrival times (IATs) between sequential low and high rate reviews.

$$SLH(p) = avg(IAT(r_i, r_{i+1}) | r_i \in \{1, 2\} \wedge r_{i+1} \in \{5\})$$

Sequential High Ratings (SHR): Next we aim to capture how many 5-star reviews turn up after a negative review. This feature is modeled as the average number of sequential 5-star reviews following a low star review. The intuition is that while in target products the number of such reviews is limited to crowd campaign size, this feature does not carry any constraints in randomly sampled products.

$$SHR(p) = avg(k | r_i \in \{1, 2\} \wedge r_{i+1}, .., r_{i+k} \in \{5\})$$

Ratio of High Rating Reviews (RHR): As fake reviews are generated rapidly while in a normal situation reviews arrive more randomly, this feature measures the ratio by dividing SHR feature by its duration.

$$RHR(p) = \frac{SHR(p)}{IAT(r_i, r_{i+k})}$$

Variance of Inter-arrival Times (VIT): This last micro feature measures how inter-arrival time varies among ratings associated to a specific product. The intuition is that target products at some points of their life receive fake reviews rapidly and then reach an equilibrium state in which they no longer exhibit abnormal behavior. We model this behavior by taking the standard deviation between the median and maximum values of inter-arrival times for each product. This feature has relatively greater value in target products.

$$VIT(p) = STD(median(IATs), max(IATs))$$

4.2 Macro Features

Our macro features consider ratings in the context of a neighborhood of related products to measure the deviation from a base-model.

Base-model: We use an Amazon dataset (McAuley et al. 2015) including ~83M reviews associated with ~1.5M products spanning from May 1996 to July 2014 to build the base-model. For example, the average rating distribution over all products in this dataset could be treated as base rating behavior. However, different products do not follow similar distributions, e.g., due to different quality, so relying on a single baseline does not provide a fair comparison. Therefore, we apply k-means clustering on base distributions to cluster similar samples together and scatter distant ones in different clusters.

Measuring Deviation: We use Kullback-Leiber (KL) divergence to compute the relative entropy between two probability mass distributions (PMD) similar to proposed approaches in (Shah et al. 2016; Viswanath et al. 2015). Formally, the KL-divergence between base-model (M) and a distribution (P) of a given product attribute is defined as:

$$KL(P, M) = \sum_{i=1}^{n} P_i \times \log_2 \frac{P_i}{M_i} \tag{1}$$

For example, if M and P are the probability mass distributions of ratings, then M_i and P_i indicate the probability of rating i in the corresponding distributions where $1 \leq i \leq 5$. The 0 value for KL-divergence means two distributions are identical while larger values indicate higher discrepancy between them. We refer to the value of KL-divergence as a product's anomaly score.

Adding the notion of clustering, a direct approach for calculating an anomaly score would be to aggregate KL-divergence between P and each cluster as follows:

$$\sigma(p) = \sum_{k=1}^{K} \rho_k \times KL(P, M_k) \tag{2}$$

Where K, M_k and ρ_k indicate the number of clusters, probability mass distribution of the center of the cluster k and proportion of samples in cluster k, respectively. However, the output of the function is dominated by large clusters. For example, if a new sample is closer to a cluster with small size and far from large clusters, then the anomaly score becomes large even it is similar to one of the base behaviors. To address this issue, we modify the distance function to reinforce the impact of distance rather than cluster size inspired by *Inverse Distance Weighting* methods.

$$\sigma(p) = \begin{cases} 0, & \text{if } \exists k, KL(P, M_k) = 0 \\ \dfrac{1}{\sum_{k=1}^{K} \frac{\rho_k}{KL(P, M_k)}}, & \text{otherwise} \end{cases} \tag{3}$$

Table 2: Performance Evaluation of TOmCAT using Feedforward Neural Network (NN) in Different Settings.

	Target Products			Random Products			
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Accuracy**
Macro Features	83	81	82	81	83	82	**82**
Micro Features	69	83	75	86	74	79	**77**
Macro + Micro Features	**86**	**83**	**84**	**82**	**85**	**84**	**84**
w/o Early Ratings	81	82	81	82	81	82	**81**
w/o Temporal Features	70	69	70	69	70	70	**69**

We introduce three attribute distributions where their deviations from the base-model form our macro features.

Rating Distribution: Since crowd review manipulation jobs solicit highly positive reviews, we assume that the rating distribution for target products should be skewed to high ratings. Considering 5-star rating system, rating distribution P_r is the probability mass function $[P_{r_1}, P_{r_2}, P_{r_3}, P_{r_4}, P_{r_5}]$ where P_{r_i} is the probability of observing i-star rating calculated as:

$$P_{r_i} = \frac{|r_i|}{\sum_{j=1}^{5} |r_j|}$$

Since crowd reviews often appear in a row, investigating inter-dependency behavior provides significant information about the existence of manipulation. Inspired by this property, the remaining macro features model inter-dependency features.

Inter-arrival time Distribution: Since crowd reviews often turn up in short period of time, their inter-arrival times deviate from the base-model. Similar to time-stamp, inter-arrival time is not categorical data, requiring further care. We adapt the approach proposed in (Hooi et al. 2016) to discretize the value space of continuous attributes. The intuition is that if maximum value of IATs (5 years in our case) is larger than minimum value (0 day) with an order of magnitude, then the value space is split logarithmically into d buckets:

$$d = \log_{base} max(IAT)$$

In our experiments, we set the value of the logarithm base to be 2, and as a result the number of buckets is 15.

Inter-arrival rating Distribution: Since a low-rate reviews can trigger restoration attacks and crowd reviews are highly rated, there should be a significant difference between two sequential ratings. We model this property as inter-arrival rating (IAR). The intuition is that the rating gap in target products reviews is higher than random products. On a 5-star rating system, possible values for IAR are $[-4, -3, -2, -1, 0, 1, 2, 3, 4]$ and thus, the probability mass function contains 9 discrete values.

4.3 TOmCAT Evaluation

We next evaluate the impact of different crowd footprints on the TOmCAT approach. Do macro features work well? Or micro features? We also explore the impact of features based on the first few reviews (which may be significant for detecting promotion attacks) and temporal-based features.

Experimental setup: For setup, the feed-forward neural network parameters such as hidden layers' dimension, regularization parameter λ and learning rate α are chosen via parameter tuning and we report the best results. Hidden layers are set to be 5 and 7 dimensions. λ and α are set to be 0.8 and 0.02 respectively. We apply normalization on input data to facilitate the convergence process before feeding them into the neural network. We train the network over 30% of samples and then evaluate its performance over the remaining 70% samples, where the results are averaged over 20 runs.

Results: Table 2 reports the results of TOmCAT for different types of crowd footprints. We report Precision, Recall and F1 score for each class of products and overall accuracy. We consider all of the micro features, all of the macro features, and both macro and micro features. As we expected, the aggregation of micro and macro features performs better in identifying target products with 86% recall and 84% accuracy. In contrast, TOmCAT identifies target products with 83% and 69% recall in the presence of only macro features and only micro features respectively. It is evident that a comprehensive detection framework boosts performance.

Further, we also consider a special case where we drop all features based on the first few reviews of all products. Extracting features from the complete series of reviews can successfully model restoration attacks but it can miss promotion attacks, specifically ones targeting new products. By definition, these scenarios create circumstances that are not yet optimal for the model to detect. Therefore, we execute our feature extraction methodology only on the first n reviews in addition to the complete series of reviews. In the experiments, we set the value of n to be 5 to cover the majority of this type of attack considering crowd campaign size typically varies from 5 to 10 fake reviews. We see in Table 2, fourth row, that performance metrics drop in the absence of early ratings features confirming the real impact of addressing this scenario. For example, recall in recognizing target products decreases from 86% to 81%.

4.4 Comparison With Baselines

In this section, we compare TOmCAT to six existing baselines which are originally designed to identify spam-behavior at the user level and we are eager to evaluate their performance at the product (target) level. These works aim to identify fraudulent users using rating and temporal features. It should be noted that network-based approaches are beyond the scope of this paper. To evaluate unsupervised approaches, we use precision @k by varying k from 100 to

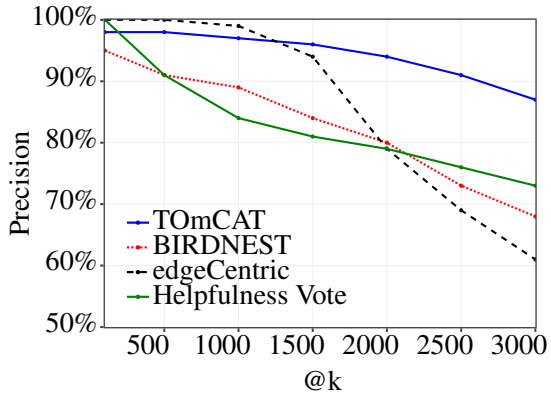

Figure 6: Comparison with unsupervised approaches: TOmCAT captures target products with higher precision

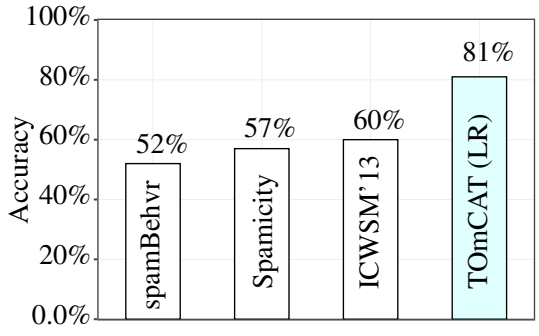

Figure 7: Comparison with supervised approaches: TOmCAT captures target products with higher accuracy

3,000. To adapt our model with precision @k, we examine the anomaly scores obtained from Sigmoid activation function at the output layer of the feedforward neural network and then sort the products in descending order.

- *Helpfulness Vote* (Mukherjee et al. 2013a): On Amazon users can provide feedback to the reviews via helpfulness votes. We assume that target products receive fewer helpfulness votes. This approach uses the average of helpfulness votes of reviews of each product and ranks them in ascending order based on their helpfulness score.
- *BIRDNEST* (Hooi et al. 2016) models temporal gaps and rating deviations.
- *edgeCentric* (Shah et al. 2016) also models temporal gaps and rating deviations.

Figure 6 illustrates that TOmCAT is superior to its alternatives especially as k increases acknowledging its capability in identifying most target products. For example, precision varies from 98% to 87% for different values of k. These results indicate that careful feature modeling can be important for defending against crowd attacks.

We further compare TOmCAT with the following supervised approaches.

- *SpamBehavior* (Lim et al. 2010) uses the average of rating deviation of individual ratings from overall rating as a feature.
- *Spamicity* (Mukherjee et al. 2013a) takes review-burstiness and maximum reviews per day as features.
- *ICWSM'13* (Mukherjee et al. 2013b) describes each user (or product in our setting) as its fraction of positive reviews, maximum reviews per day, and average rating deviation.

For a fair comparison, we applied Logistic Regression (LR) as the classifier used in these baselines on corresponding feature sets. We report accuracy as the appropriate metric for balanced datasets. As shown in Figure 7, TOmCAT outperforms its alternatives significantly with 81% accuracy.

Table 3: Alternative Datasets Description

	# Items	# Reviews
Amazon (Jindal and Liu 2008)	13,449	1,573,555
App store	2,858	304,450
Yelp	174,567	5,261,669

We can conclude that baselines that are originally targeted at the reviewer level may miss some of the subtle behaviors evidenced at the product level. Also, they tend to capture users with clear patterns of spam-behavior while target products do not show such clear anomalous signals and they contain mixed legitimate and non-legitimate behaviors as they receive reviews from actual users as well.

4.5 Evaluation Over Other Datasets

Here, we are interested to evaluate the effectiveness of our approach on three other datasets: from Yelp, the App store, and another existing Amazon dataset (Jindal and Liu 2008). Our goal is to detect manipulation patterns left by crowd attacks at the target level using only timing and sequence of ratings. Hence, detection at either reviews or reviewers level which aims to investigate textual content or network behaviors are ruled out. Table 3 shows the summary of each dataset in terms of number of items and number of reviews. It should be noted that we re-apply our feature extraction methodology and build the base-model over these datasets. Since the main challenge here is lack of ground truth, we use complementary information e.g., an item could be a target if it received reviews from suspicious reviewers, to evaluate the results.

Amazon (Jindal and Liu 2008): This dataset has been used for review spam detection since 2007. It includes information about reviewers as well which we use as the support information to evaluate our model. We filter out products with fewer than 50 reviews and end up with 13,449 items. **Findings:** Ranking items by their anomaly scores (Sigmoid values), we find significant number of items with Sigmoid score of 1 (~2,000 items) meaning they are extremely simi-

lar to our ground truth. For evaluation, we pulled out 100 of these items randomly and investigated their reviewers. The idea is that if they have a review written by a suspicious reviewer, then they are potentially a target of manipulation. We follow the same method to identify suspicious reviewers as explained in section *Identifying Suspicious Reviewers*. We observe that **95 out of 100** top-ranked items by our approach do indeed receive at least one review from suspicious reviewers. **App store:** This dataset has been introduced in (Li et al. 2017) where it follows similar strategy described in this paper to identify target mobile Apps that includes 100 primary target Apps and in total 2,858 Apps. **Findings:** Evaluating 100 of top-ranked Apps, we find *47 of them are subset of primary target Apps*. We also investigated reviewers associated with these Apps and observed **86 out 100** Apps received at least one review from suspicious reviewers.

Yelp: This dataset is released as part of round 11 of the Yelp challenge in January 2018 [2]. **Findings:** Interestingly, ~19,000 items are ranked top with anomaly score of 1. We pulled out 100 of the top items randomly for further evaluation. Unfortunately, the Yelp dataset only provides information about items not reviewers, so previous method to identify suspicious reviewers does not apply here. Therefore, we explored other available meta-data to support our findings. For example, **16 out of 100** businesses are closed now and **59 and 79 of the items** received fewer than 10 and 20 helpfulness votes respectively while on average items on Yelp receives 42 helpfulness votes.

In summary, TOmCAT performs well in uncovering target products in our original Amazon dataset. This framework is also extrapolated on other review platforms. Furthermore, TOmCAT outperforms the baselines. Despite its success in addressing *existing crowd attacks*, it may perform poorly against strategic attackers who aim to nullify timing-based features. This motivates us to build upon TOmCAT and propose TOmCATSeq complementary approach benefiting from minimal features.

5 Complementary TOmCATSeq

As manipulators are constantly evolving their strategy to circumvent new detection methods, the core TOmCAT footprints may lose their power against potential hard-to-detect attack behaviors. Indeed, we find empirically that inter-arrival time plays an important role in detecting manipulation in our dataset. But what if attackers undermine the power of these footprints?

To test this attack strategy, we consider TOmCAT without any temporal features (keeping all other configurations the same). As reported in Table 2(w/o Temporal Features) there is a significant drop across various performance metrics. For example, recall/precision of target products and overall accuracy drop from 86/83/84 to 70/69/69 respectively. This indicates the importance of careful consideration of strategic attacks as the TOmCATSeq strive to address such scenarios.

[2]https://www.yelp.com/dataset/challenge

Costs Borne By Attackers. This suggests that attackers may be able to subvert TOmCAT, though at some cost. For example, reviews launched by crowd attacks typically show up in a short time window. However, prolonging inter-arrival time between fake reviews to mimic base-model distributions and conceal anomaly patterns may affect the ultimate goal of crowd attacks in several ways: (i) *Slowing Down the Recovery Period:* As there is a delay between fake reviews, the overall rating affected by negative ratings will be rebuilt slowly, meaning that potential customers may be discouraged by the low overall rating. (ii) *Receiving Legitimate Low Ratings:* Since fake reviews are posted at a slower rate to avoid detection, other legitimate low ratings have an opportunity to arrive, diminishing the impact of the crowd attack.

TOmCATSeq Structure. Regardless of the negative consequences to attackers, such strategic attacks pose serious challenges to the ongoing success of TOmCAT and models built on similar crowd footprints.The main idea of *TOmCATSeq* is to exploit only rating patterns via a novel Rating-based Bidirectional LSTM model. It has two main advantages: First, since it is an end-to-end model, it avoids the need for carefully designed features, instead learning effective representations directly from the input sequence data. Second, it only focuses on rating behavior regardless of their temporal characteristics, meaning that attacks on the timing of ratings are powerless.

Long Short-Term Memory networks known as LSTM is special type of recurrent neural networks (RNN) introduced in (Hochreiter and Schmidhuber 1997). The chain-like nature of RNN naturally fits time-series data as in our ratings scenario. Despite the advances in language modeling, speech recognition, machine translation and computer vision using RNN, the power of deep learning in detecting review manipulation has not been explored yet. In a nutshell, RNNs at each time-step i transfer the corresponding input x_i along with the information obtained from previous hidden state h_i to a new hidden state h_{i+1}. LSTM is popular form of RNNs due to its capability in remembering long-term dependencies. Traditional RNN-based models do not perform well on long sequences since they only reintroduce information a few steps back from the current step. LSTM solves this problem by designing a more sophisticated hidden states.

Traditionally, LSTM is used for text data in which the input at each time-step is a one-hot vector of the corresponding word in the sequence. In rating data, we only have 5 possible different ratings/words so, the vocabulary size is very efficient while that of language models has more than 10K words (Ahn et al. 2016).

We also leverage bidirectional LSTM which provides the capability of taking information from both earlier and later in the sequence. Crowd attacks properties motivate the practice of bidirectional LSTM in our problem. To figure out the occurrence of a manipulation attack, it is not sufficient to just investigate the first part of the sequence, since the model requires more information than just observing one or more low-rate reviews as an attack trigger. Thus, information from the other direction of the sequence containing

Table 4: TOmCATSeq Performance Evaluation

Target Products			Random Products			
Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Accuracy**
94	74	83	63	90	74	**79**

a series of fake reviews is also required. For this purpose, bidirectional LSTM adds a backward recurrent layer. Hence, TOmCATSeq can detect abnormal rating patterns by memorizing events from past and future time-steps.

The input to the network is the rating sequence of products representing each rating value as a one-hot vector. We treat 1 and 2 stars similarly so the vector representation of each rating has only 4 dimensions. We tried with 5 dimensional vectors and results are similar or slightly worse. We can relate this to the fact that ratings 1 and 2 are treated equally as negative feedback.

TOmCATSeq comprises three layers as *Embedding layer*, *Bi-LSTM layer*, and *fully connected layer*. The Embedding layer ecodes the input one-hot vectors into embedding vectors. The output of the last time-step in the Bi-LSTM layer is fed into the fully connected layer with Sigmoid as activation function. Furthermore, to prevent over-fitting, we use *dropout* (Srivastava et al. 2014) regularization technique. In training stage, dropout sets a portion of Bi-LSTM hidden units to zero with a probability determined by dropout rate.

TOmCATSeq is a binary classifier that learns to classify a rating sequence belongs to a specific product as target or non-target. More formally, the goal is to learn a classification function $f(R_p, \theta)$ that determines the label of product p, given a set of model parameters θ. In training stage, the model is fed training instances (R_p, l_p), where R_p is rating sequence of product p and l_p is the actual label acquired from the ground truth. We consider the binary cross entropy (LeCun, Bengio, and Hinton 2015) as loss function.

5.1 TOmCATSeq Evaluation

Here, we present the evaluation results of TOmCATSeq.

Experimental setup: We build TOmCATSeq using the Keras framework (Chollet 2017). The model hyper-parameters are tuned using grid search. The number of epochs, the size of batches, the size of embedding layer, and the hidden size of BiLSTM are selected from [5, 6, 7, 8, 9, 10], [4, 8, 16, 32], [8, 16, 32, 64], and [8, 16, 32, 64] respectively. The learning rate and the dropout rate are selected from [0.01, 0.001, 0.0001] and [0.0, 0.2, 0.5, 0.7, 0.9] respectively. The parameters of the network are optimized by employing Adam optimizer (Kingma and Ba 2014) though we have also tried a suite of different optimizers as *Adamax, SGD, RMSprop, Adagrad, Adadelta and Nadam*. The weights are initialized based on various number of distributions as *uniform, normal and zero*. We set the maximum sequence length to be 150 empirically. Therefore, in the training stage we limit ourselves to items with at most 150 ratings and address shorter ones with zero padding. In the inference stage, sequences longer than 150 ratings are chunked into multiple sequences. However, sequential

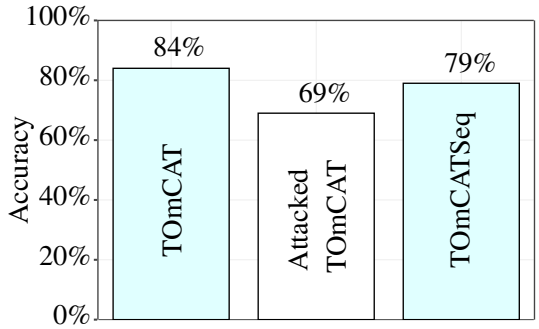

Figure 8: Impact of Removing Temporal Features on TOmCAT and TOmCATSeq Performance

chunks are not mutually exclusive but they have some overlap. If that were the case we would miss the crowd attacks happening on the edge of two chunks because their ratings are divided between two different sequences. To overcome such trivial but destructive cases, we set the overlap to be 20 ratings (as crowd campaign size is typically between 5 to 10, it assures to cover crowd attacks as many as possible). In the evaluation stage, if at least one chunk associated to a product is recognized as a target then the corresponding product is considered as a target product.

Results: We report the evaluation results on testing data in Table 4. Briefly, 94% of target products have been captured by TOmCATSeq and 74% of products that are recognized manipulated are actual target products. Putting it all together, Figure 8 demonstrates the performance of TOmCAT (84%) and how it is affected (69%) when we model attack footprints using only ratings data. Finally, TOmCATSeq is only 5% worse than when we have all the features and is able to recover the accuracy (from 69% to 79%) by relying on an end-to-end RNN-based model fed only rating sequences.

Discussion: The initial evaluation of TOmCATSeq shows promising results relying on rating data while temporal features are left out. This reinforces the capability of TOmCATSeq in the face of strategic attackers who may adopt a normal temporal distribution to circumvent the detection model. It tells us that even if the attack footprints are not explicit enough to be formulated as handcrafted features, TOmCATSeq which is built over BiLSTM can perceive hidden patterns left by crowd attacks. This also demonstrates the value of our ground truth which can leverage the power of RNN-based models while unsupervised approaches lack

the capability to uncover *difficult-to-detect* review manipulation. However, TOmCATSeq has lower precision compared to TOmCAT (74% v.s. 83%) meaning it misclassifies some non-target products. Of course, strategic attackers may seek to undermine approaches like TOmCATSeq, so we are engaged in a continuing effort to defend against future attacks. For example, adaptive attackers (Wu et al. 2017) can still circumvent TOmCATSeq by following legitimate rating behaviors. In our ongoing work we aim to incorporate other information such as textual information into the detection model to address adaptive attacks. Finally, the work in this paper can be extended by studying how both TOmCAT and TOmCATSeq can be combined together to work as a single end to end system.

6 Conclusion and Future Work

We have proposed the TOmCAT framework to uncover crowd review manipulation attacks on sites like Amazon. The proposed model – unlike previous efforts that focus on identifying manipulation at the review or user level – investigates manipulation in the aggregate at the target level. The main contributions are to: (i) model product review behavior through a set of micro and macro features inspired by the presence of restoration and promotion attacks; (ii) evaluate the effectiveness of our TOmCAT model on other online review platforms; and (iii) exploit hidden patterns of manipulation attacks to defeat strategic attackers by leveraging RNN on rating manipulation scenario for the first time. Our results are encouraging, indicating that our model can indeed discover many target products. In our ongoing work, we are expanding our coverage of crowd campaigns which launch via other channels like social media e.g., Facebook (Post 2018). We are also eager to further explore how linguistic features of product reviews may provide some insights into manipulation detection to complement our focus in this paper on the rating and temporal behavioral of the reviews.

References

Ahn, S.; Choi, H.; Pärnamaa, T.; and Bengio, Y. 2016. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*.

Akoglu, L.; Chandy, R.; and Faloutsos, C. 2013. Opinion fraud detection in online reviews by network effects. In *AAAI*.

Cao, Q.; Yang, X.; Yu, J.; and Palow, C. 2014. Uncovering large groups of active malicious accounts in online social networks. In *CCS*, 477–488. ACM.

Chollet, F. 2017. *Deep learning with Python*. Manning Publications Co.

Duan, W.; Gu, B.; and Whinston, A. B. 2008. Do online reviews matter? an empirical investigation of panel data. *Decision support systems*.

Harris, C. 2012. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on Artificial Intelligence*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

Hooi, B.; Shah, N.; Beutel, A.; Günnemann, S.; Akoglu, L.; Kumar, M.; Makhija, D.; and Faloutsos, C. 2016. Birdnest: Bayesian inference for ratings-fraud detection. In *SDM*.

Hu, N.; Pavlou, P. A.; and Zhang, J. 2006. Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *ICEC*.

Jiang, M.; Cui, P.; Beutel, A.; Faloutsos, C.; and Yang, S. 2014. Inferring strange behavior from connectivity pattern in social networks. In *PAKDD*.

Jindal, N., and Liu, B. 2008. Opinion spam and analysis. In *WSDM*.

Kaghazgaran, P.; Caverlee, J.; and Alfifi, M. 2017. Behavioral analysis of review fraud: Linking malicious crowdsourcing to amazon and beyond. In *ICWSM*.

Kaghazgaran, P.; Caverlee, J.; and Squicciarini, A. 2018. Combating crowdsourced review manipulators: A neighborhood-based approach. In *WSDM*.

Kao, J. 2017. More than a million pro-repeal net neutrality comments were likely faked, https://tinyurl.com/fcc-nn, last access: 02/26/2018.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, S., and Shah, N. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*.

Kumar, S.; Hooi, B.; Makhija, D.; Kumar, M.; Faloutsos, C.; and Subrahmanian, V. 2018. Rev2: Fraudulent user prediction in rating platforms. In *WSDM*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*.

Lee, K.; Tamilarasan, P.; and Caverlee, J. 2013. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. In *ICWSM*.

Li, F.; Huang, M.; Yang, Y.; and Zhu, X. 2011. Learning to identify review spam. In *IJCAI*.

Li, J.; Ott, M.; Cardie, C.; and Hovy, E. 2014. Towards a general rule for identifying deceptive opinion spam. In *ACL*.

Li, S.; Caverlee, J.; Niu, W.; and Kaghazgaran, P. 2017. Crowdsourced app review manipulation. In *SIGIR*.

Lim, E.-P.; Nguyen, V.-A.; Jindal, N.; Liu, B.; and Lauw, H. W. 2010. Detecting product review spammers using rating behaviors. In *CIKM*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

Mukherjee, A.; Kumar, A.; Liu, B.; Wang, J.; Hsu, M.; Castellanos, M.; and Ghosh, R. 2013a. Spotting opinion spammers using behavioral footprints. In *KDD*.

Mukherjee, A.; Venkataraman, V.; Liu, B.; and Glance, N. S. 2013b. What yelp fake review filter might be doing? In *ICWSM*.

Ott, M.; Cardie, C.; and Hancock, J. T. 2013. Negative deceptive opinion spam. In *HLT-NAACL*.

Piskorski, J.; Sydow, M.; and Weiss, D. 2008. Exploring linguistic features for web spam detection: a preliminary study. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*.

Post, T. W. 2018. How merchants use facebook to flood amazon with fake reviews, https://goo.gl/u4wqxk, last access: 25/04/2018.

Prakash, B. A.; Sridharan, A.; Seshadri, M.; Machiraju, S.; and Faloutsos, C. 2010. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD*.

Sandulescu, V., and Ester, M. 2015. Detecting singleton review spammers using semantic similarity. In *WWW*.

Shah, N.; Beutel, A.; Hooi, B.; Akoglu, L.; Gunnemann, S.; Makhija, D.; Kumar, M.; and Faloutsos, C. 2016. Edgecentric: Anomaly detection in edge-attributed networks. In *ICDMW*.

Shin, K.; Hooi, B.; Kim, J.; and Faloutsos, C. 2017. D-cube: Dense-block detection in terabyte-scale tensors. In *WSDM*.

Song, J.; Lee, S.; and Kim, J. 2015. Crowdtarget: Target-based detection of crowdturfing in online social networks. In *CCS*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*.

Viswanath, B.; Bashir, M. A.; Zafar, M. B.; Bouget, S.; Guha, S.; Gummadi, K. P.; Kate, A.; and Mislove, A. 2015. Strength in numbers: Robust tamper detection in crowd computations. In *COSN*.

Wang, G.; Wilson, C.; Zhao, X.; Zhu, Y.; Mohanlal, M.; Zheng, H.; and Zhao, B. Y. 2012. Serf and turf: crowdturfing for fun and profit. In *WWW*.

Wu, L.; Hu, X.; Morstatter, F.; and Liu, H. 2017. Adaptive spammer detection with sparse group modeling. In *ICWSM*.

Yao, Y.; Viswanath, B.; Cryan, J.; Zheng, H.; and Zhao, B. Y. 2017. Automated crowdturfing attacks and defenses in online review systems. In *CCS*.

Ye, J.; Kumar, S.; and Akoglu, L. 2016. Temporal opinion spam detection by multivariate indicative signals. In *ICWSM*.