Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Exploring groups of opinion spam using sentiment analysis guided by nominated topics

Jiandun Li [*], Pin Lv, Wei Xiao, Liu Yang, Pengpeng Zhang

School of Electronic and Information Engineering, Shanghai Dianji University, 201306, China

ARTICLE INFO

Keywords:
Review spam
Opinion spam
Spam group
Nominated topic
Aspect-oriented sentiment

ABSTRACT

Currently, it is common to see untruthful opinions (also known as review spam, fraud or shilling attack) that resemble each other explicitly or implicitly across multiple business-to-customer websites or opinion sharing communities. Unfortunately, these fake recommendations can be fabricated by individual spammers or results of a manipulation campaign. Considering its severe harmfulness in influencing a product's reputation, grouped spam is more urgent to detect than individual fraud. Most state-of-the-art techniques of labeling grouped spam, e.g., Frequent Itemset Mining (FIM) or Latent Dirichlet Allocation (LDA), are completely unsupervised and incapable of making good use of officially recommended topics, such as *appearance*, *speed* and *standby* are three suggested aspects along a cell phone product in JD.com. In this paper, we introduce a novel approach based on aspect-oriented sentiment mining that can identify spam groups supported by nominated topics. Experiments show that our method is effective and outperforms several state-of-the-art solutions with statistical significance on two metrics, content duplication and burstiness of time.

1. Introduction

As user-generated content, reviews are somewhat essential in portraying an online item's reputation. Recommendations promote sales, whereas complaints force the manufacturer to improve product quality (Hennig-Thurau, Gwinner, Walsh, & Gremler, 2004; Li, Wang, Yang, Zhang, & Yang, 2020; Mayzlin, 2006). However, driven by profit, opinion fraud (i.e., review spam) originated at the same time with the objective of overrating one's products or underrating opponents' items (Heydari, Tavakoli, Salim, & Heydari, 2015; Hussain, Turab Mirza, Rasool, Hussain, & Kaleem, 2019; Vidanagama, Silva, & Karunananda, 2019). Considering its timely effect in distorting a product's reputation, grouped spam (also known as collaborated spam) out of short-term manipulation campaigns is the severest attack so far; therefore, identifying spam groups or spammer groups is of great significance to today's Business to Customer (B2C) websites and opinion sharing communities. A spam group consists of several fake reviews, whereas a spammer group is composed of malicious reviewers. When user profiles are available (e.g., Yelp), many attributes (e.g., average overall rating, reviewing burstiness) could be adopted to unveil spammer groups. However, since professional spammers are used to registering new identities, account-centric attributes become less reliable; this fact also demonstrates that identifying grouped manipulators is difficult if it is not impossible. Nevertheless, exposing collaboratively spammed reviews is promising by which we can also get rid of malicious impact.

Compared to spam or spammer recognition, which suffers from the lack of ground-truth labels, identifying grouped review spam is affordable (Mukherjee, Liu, & Glance, 2012; Zhang, Wu, & Cao, 2018). Several solutions have been proposed and they typically have two steps: clustering reviews and labeling spam groups. However, their employed attributes and clustering models are different. Mukherjee et al (Mukherjee et al., 2012) applied the Frequent Itemset Mining (FIM) technique to find candidate spammer groups in which several customers overlapped on multiple items that they reviewed. Many attributes were introduced to label spammer groups, including group indicator set {*time window*, *deviation*, *content similarity*, *member content similarity*, *early time frame*, *size ratio*, *size*, *support count*} and individual indicator set {*rating deviation*, *content similarity*, *early time frame*, *member coupling*}. A similar study conducted by Xu and Zhang (C. Xu & Zhang, 2015) utilized more behavioral indicators to cluster reviewers, e.g. *rating consistency* and *temporal traits*.

Recently, graphs, especially bipartite graphs, are popular in shaping

[*] Corresponding author.
E-mail addresses: lijd@sdju.edu.cn (J. Li), lvp@sdju.edu.cn (P. Lv), xiaow@sdju.edu.cn (W. Xiao), yangliu@sdju.edu.cn (L. Yang), zhangpp@sdju.edu.cn (P. Zhang).

https://doi.org/10.1016/j.eswa.2021.114585
Received 24 August 2020; Received in revised form 3 January 2021; Accepted 6 January 2021
Available online 12 January 2021

reviewers' behaviors. Allahbakhsh et al. (Allahbakhsh et al., 2013) extended (Mukherjee et al., 2012) to address pure rating systems. They also borrowed FIM to generate candidate spam groups out of a signed bipartite graph. Related features that can discriminate these groups are *rating similarity*, *rating time*, *rating spamicity* and *member suspiciousness*. Ye and Akoglu (Junting Ye, 2015) first recognized spammed products by two network-based markers, called *neighbor diversity* and *self-similarity*, and then clustered reviewers by a two-hop subgraph inducing technique. Wang et al. (Wang, Hou, Song, Li, & Kong, 2017; Wang, Gu, Zhao, & Xu, 2018) generated spammer groups based solely on graph topology. When one indicator out of the set {*review tightness*, *neighbor tightness*, *product tightness*, *average time window*, *rating variance*, *product review ratio*, *early review*, *group size*} overflowed its corresponding threshold, the group was labeled as spam. When comments between reviewers were centered, Choo et al (Choo, Yu, & Chi, 2015) constructed a graph by exploring community mining and sentiment sensing techniques, and they concluded that strong positive communities are more likely to be collaborative spam. Attributes that are employed to arrive at a spammer group include *content similarity*, *rating abused item ratio*, *maximum one day review ratio*, *review burstiness*, *first review ratio* and *deviated rating ratio*. Do et al (Do, Hussain, & Nguyen, 2017) modeled the review system as a weighted tri-partite graph (reviewers, reviews and products) and further updated each node's suspicion score via an epidemic propagation model. Initially, the suspicion score of any node was calculated through commonly used features, e.g., *duplication*, *deviation*, *one-time reviewer*, *burstiness*. Additionally, they verified that the K-means model outperformed FIM and others. After modeling the problem space as a bipartite graph, Akoglu et al (Akoglu, 2013) modeled the grouped spam problem as a graph classification problem with belief propagation through promoting or defaming word-of-mouth. Considering some ground-truth reviews, they classified all nodes and found top fraudsters. Finally, they grouped reviewers using the cross association clustering algorithm (Chakrabarti, Papadimitriou, Modha, & Faloutsos, 2004). Based on prior knowledge about some human-labeled groups, Zhang et al (Zhang et al., 2018) proposed a semisupervised framework based on positive unlabeled (PU) learning.

Researchers have also centered review content to prepare for clustering. In addition to directly evaluating review similarity, more studies attempted to compare aspect-oriented sentiments before grouping. Li et al (Jiwei Li, 2013) introduced the Latent Dirichlet Allocation (LDA) algorithm and verified that topic modeling is effective in exploring the difference between genuine and untruthful reviews (accuracy 95%). Peng et al (Peng & Zhong, 2014) computed a review's sentiment score based on its polarity and strength; however, the unexpected rules (e.g. sentiment inconsistency between the rating and review texts) based on which they calculated the spamicity score are highly in doubt and cannot generalize to most real-life opinion systems. Aiming to identify multiple opinions fabricated by the same spammer, Sandulescu et al (Vlad Sandulescu, 2015) adopted LDA for semantic similarity comparison. The Jensen-Shannon measure was utilized to quantize the distance between reviews. Based on LDA, Lee et al (Lee, Han, & Myaeng, 2016) introduced five topics based on POS (part of speech) and sentiments to distinguish sincere and deceptive reviews. The proposed topics are category-independent and can reflect a reviewer's writing patterns. Interestingly, Wang et al (Wang, Gu, & Xu, 2018) adapted LDA to a "product-cluster-reviewer" setting and successfully found top spammer groups. First, they selected abnormal reviewers from each cluster, and then modeled inter-review distances via Jaccard similarity fed by *review time* and *rating deviation*. Furthermore, they applied the SCAN method (X. Xu, Yuruk, Feng, & Schweiger, 2007) to cluster customers based on the indicator set introduced earlier (Wang et al., 2017).

Among these state-of-the-art solutions in detecting spam groups, approaches that can take advantage of aspect-oriented sentiments have great prospect in exposing camouflaged duplications, which can be labeled as grouped spam. The reasons are twofold: 1) contrast to lexical or syntactic indicators, through fine-grained sentiments we can uncover deep-buried similarity between reviews; and 2) these models are unsupervised and require no annotations. So far, the direction is still open, and there are many challenges to address, at least including: 1) semantic topics are difficult to visualize and interpret; 2) officially provided sentiment aspects are overlooked; and 3) most are time-consuming. In this paper, after examining the review practices of JD.com and TMALL.com, the top two B2C websites in China, we observe that reviewers are greatly privileged in selecting experience topics from a predefined list simply by finger-touching or mouse-clicking; what is left is only to append their feelings correspondingly. But state-of-the-art approaches are completely unsupervised and incapable of taking these suggested aspects to improve their performance.

Supported by these nominated topics, we propose a novel approach to identify aspect-oriented sentiments (explicitly and inexplicitly), and we then borrow the K-means algorithm to cluster reviews, i.e., a Grouped Spam Detection approach based on Nominated Topics (GSDNT); finally, we adopt content duplication and time burstiness to label highly suspicious groups as spam. The contribution of this paper is fourfold: 1) this is the first study using platform-offered sentiment topics to improve the performance of spam detection; 2) this is also the first study to address Chinese reviews for aspect-oriented fraud labeling; 3) cross-platform reviews are integrated for spam group clustering; and 4) we achieve high performance in spam clustering.

The remainder of this paper is organized as follows. In Section 2, we describe the dataset crawled from JD.com and TMALL.com. We present our solution GSDNT in detail in Section 3. In Section 4, we conduct some experiments and discuss them. We conclude this study in Section 5.

2. Datasets

In this paper, we attempt to unveil latent groups of review spam in Chinese language. From dominance consideration, we decide to crawl data from JD.com (referred as JD in the rest of this paper) and TMALL.com (referred as TMALL in the rest), following their corresponding data policies. According to a report published by iiMedia Research, 83.8% of the B2C market was shared by JD and TMALL during the first half of 2018 (Li et al., 2020). Another reason why we choose these companies is because of their advanced regulations with respect to spam.

Overall, 14,821 recent reviews were collected from JD, and 29,986 posts were collected from TMALL (crawled on April 16, 2020). Related product categories are clothes, purses, shoes, cell phones, laptops, solid-state disks, canteens and makeup (Table 1). Within these review texts, 20.67% are written in (topic, sentiment) format, whereas 97.8% of the topics are drawn from the officially nominated topic list that corresponds to the target item.

Based on these observations, we argue that nominated-topic-oriented sentiment mining is more appropriate than aimless clustering. The reason is that suggested topics are top attractive aspects related to the target product; therefore, they have a high probability of being selected, regardless of whether it is in a clear format. For other topics posted by reviewers, regardless of whether the post is explicit or

Table 1
Products adopted in JD and TMALL.

Brands	Series
Coach	Purse F27583, F27591
Apple	iPhone 11, XS Max
Nike	Lebron Witness IV
Lenovo	ThinkPad E490, E41
Kingston	SSD SATA3.0 2.5-inch A400
L'Oreal	Eye cream
Nivea	Lip Balm
Yanghe	Blue Classic Sea Blue 52 degrees
Under Armour	Men's training sports tights 1,257,468
Belle	Derby leather shoes 21,621
Tiger	Portable water bottle MBR-S06G-RR
Vancl	T-shirt 1,093,605

implicit, since they are secondary, we choose to ignore them in this paper. Thus, before clustering, two challenges must be addressed, i.e., how to detect whether a nominated topic is discussed by a reviewer and how to ascertain its polarity.

3. Methodology

Based on the officially nominated aspects, we propose a new method GSDNT to label highly suspect spam groups. Fig. 1 illustrates the Event-driven Process Chain (EPC) chart, in which hexagons denote events, rectangles represent resources and rounded rectangles stand for procedures. Typically, there are three phases: preprocessing, sentiment comprehension and clustering (denoted by rectangles to hexagons).

3.1. Preprocessing

Aiming to incorporate related reviews crawled from JD and TMALL together, we introduce several definitions.

Definition 1:. *Given two products, x and y; if they are highly correlated, i. e., their compound intersection $CI_{x,y} \neq \emptyset$, then they are equivalent. Therein, $\|CI_{x,y}\| = \prod_{I=B,C,T}(\|x_I \cap y_I\|)$, where $B = \{brands\}$, $C = \{categories\}$ and $T = \{nominated\ topics\}$.*

In another words, *x* and *y* are *equivalent* only if they satisfy the following conditions concurrently: 1) they are made by the same manufacturer, 2) they belong to the same category and 3) they have at least one nominated topic in common.

Definition 2:. *Given a product set S ($S \neq \emptyset$), if we cannot find a pair of products (x, y) that is not equivalent, then S is an equivalent group.*

Theorem 1:. *If S is an equivalent group, then any nonempty subset of S is equivalent.*

Theorem 2:. *Assume that S is an equivalent group; its nominated topic set is $T_S = \bigcup f(t)$, where t denotes any topic nominated by any member product of S, and $f(x)$ is a filtering function. A simple example of f could be $f(x) = \{x|count_x > \theta\}$, where θ is the threshold for topic appearance across S.*

Definition 3:. *Assume a review pair (r_x, r_y); if × and y belong to the same equivalent group, then r_x and r_y are compatible reviews.*

Through these definitions, we clarify the problem space and obtain groups of compatible reviews. Specifically, in our datasets, we find that products duplicating on categories and manufacturers always have a large proportion of nominated topics in common. For instance, sentiment aspects including *speed*, *appearance* and *standby* are nominated along most products manufactured by Apple Inc., just as *comfort*, *permeability*, *elasticity* along items made by Nike Inc. Thus, we take two steps to generate *equivalent groups*: 1) cluster items into groups based on their categories and brands; and 2) grow *equivalent groups* from equivalent pairs. For the latter, we first match individual items into item pairs according to Definition1, where an item can be replicated to different pairs. Next, we iteratively merge these pairs into larger item sets until no sets could be combined. Any sets left are *equivalent groups* and reviews targeting them are *compatible reviews*.

Subsequently, given a group of *compatible reviews*, our objective is to analyze their aspect-oriented sentiments and cluster them into candidate groups of spam, which can be further labeled as spam groups based on homogeny and/or synchronization.

3.2. Sentiment comprehension

Based on predefined formatting, we first cut compatible reviews into two subcategories, i.e., explicit-topic-oriented reviews and latent-topic-oriented reviews, and then mine their corresponding sentiment lexically.

3.2.1. Recognizing topics

For the current dataset, the default formatting of explicit-topic-

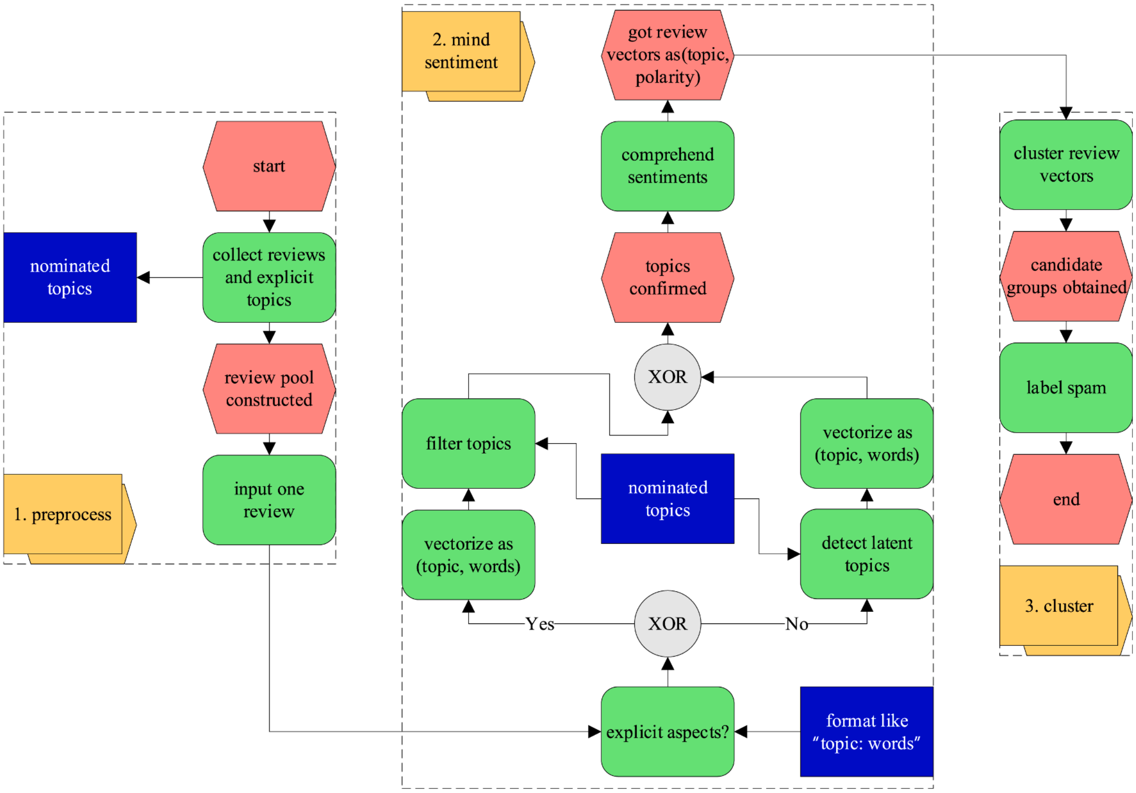

Fig. 1. The EPC chart of the framework.

oriented sentiment is "<topic>: <sentences>;" and we can easily construct $<$ topic, phrases $>$ vectors by delimiter recognition and word cutting. Assume that S is an equivalent group; then, according to Theorem 2 we can generate the nominated topics of S as T_S.

Algorithm I: detecting_latent_topics

```
input: review, nominated_topics, stop_words
01 Initiate review_words, topic_synonyms, review_vector, topic_indices
02 Cut review into terms
03 for term in terms if term not in stop_words and is noun:
04     Append term to review_words
05     for topic, synonym in nominated_topics and topic_synonyms if term = topic or term in synonym:
06         Insert term as a key to review_vector
07         Append review_words.index(term) to topic_indices
08 for topic in review_vector.keys():
09     review_vector[topic] = review_words[topic_indices[topic] + 1: topic_indices[topic + 1]]
10 return review_vector
```

When there are no explicit topics, we unveil any latent topics upon text mining. The pseudo code is shown in Algorithm I. First, we cut sentences into phrases (line 2). Given a phrase, if it is a noun and not in the list of stop words (line 3), leave it in the review content for following processing (line 4); further, if it matches any topic in T_S or one of its equivalent expressions (line 5), deem it to be the corresponding topic, reserve a place in the review vector (line 6) and memorize its index in the review content (line 7). Any discovered topic out of T_S is discarded for it is not widely adopted by the majority of reviewers. Terms that follow a matched aspect and precede the next aspect are deemed to be the sentiment of the front aspect (line 8–9). Any new terms found in topic matching or sentiment understanding are comprehended only once. Finally, we collect these aspects and sentiment into a vector for further processing (line 10).

3.2.2. *Mining sentiment*

Since most reviewers opt to recommend (other than defaming) the item to others, we choose to inspect sentiment words by applying a positive filter first (line 1–2) and a negative filter next (line 3–4) in Algorithm II. If nothing is matched, we label the sentiment as neutral (line 5). A library of words together with their polarities is established by adapting existing dictionaries (Chung-Chi Chen, 2018; Meng et al., 2011). It is also worth noting that sentiment terms are understood only once.

Algorithm II: sensing_sentiment_polarity

```
input: review_words, positive_words, negative_words
01 for word in review_words if word in positive_words:
02     return 1
03 for word in review_words if word in negative_words:
04     return −1
05 return 0
```

3.3. *Clustering and labeling*

After vectoring these reviews, we adopt the K-means algorithm for clustering (Algorithm III), which is recommended by previous study (Dominic & Lijo, 2017). There are four steps after initiation (line 1). First, we randomly set cluster centers (line 3). For a specific review vector (line 4), we compute its distances to different cluster centers (line 5); based on these distances, we place the review into the nearest cluster (line 6). The next run starts from recalculating cluster centers (line 7). When there are no updates (line 2), we collect the clustering results (line 8).

Therein, the cluster number k is determined by the elbow method, i.e., the break point of Sum of the Squared Errors (SSE). $SSE = \sum_{j=1}^{r}\sum_{j=1}^{n_i}(X_{ij} - \overline{X_i})^2$, where $\overline{X_i} = \sum_{j=1}^{n_i}X_{ij}/n_i$ and $i = 1, 2, \ldots, r$.

Algorithm III: clustering_using_kmeans

```
input: review_vectors
01 Initiate k, review_labels
02 while there is no update:
03     Randomize k cluster centers
04     for review in review_vectors:
05         Calculate distances to k cluster centers
06         Cluster review to the nearest one
07     Recalculate all cluster centers
08 return review_labels
```

When a candidate group of reviews is ready, we choose two pervasively accepted indicators, *duplication* and *burstiness* (Dewang & Singh, 2018; Heydari et al., 2015; Li et al., 2020) to evaluate its spamicity. $SI_1 = countif(r_i = r_j)/n_i$, where r_i denotes the content of review i, n_i represents the total count of reviews in cluster C and $i < j$; or $SI_2 = \sum_{i,j\in C}\|Y_{ij} - \overline{Y}\|^2$, where Y_{ij} denotes the time difference between r_i and r_j, and $\overline{Y}$ represents the mean. Considering the efficiency of *duplication* and *burstiness* in identifying spam across many platforms, we apply SI_1 (duplication level) in the first place and SI_2 (time burstiness) is used as a complement.

3.4. *Time complexity*

In preprocessing, assume we have N_{item} items and each has at most N_{topic} nominated topics, the clustering over categories and manufacturers requires $O(N_{item})$. For the second step, the key transaction is item comparison, which needs $O(N_{topic}^2)$. The worst case is that we arrive at only one *equivalent group* at last and find only one match in the end of every iteration. In this case, the count of comparisons is $C_N^2 + C_{N-1}^2 + \cdots + C_2^2$ where $N = N_{item}$, i.e.,$O(N_{item}^2)$. Therefore, the total time required for item comparison is $O(N_{topic}^2 N_{item}^2)$. It seems inefficient, but data show that items duplicated on categories and manufacturers always have a large proportion of nominated topics in common, if not the same. Thus, the preprocessing complexity is near $O(N_{item})$.

For latent topic mining among reviews, according to Algorithm I the time complexity can be evaluated by $O(N_{review}N_{term}N_{topic})$, where N_{review} denotes the number of total reviews in an *equivalent group* and N_{term} represents the maximum count of effective terms per review. Besides, the complexity can be immensely reduced because a new term is only identified once; so, the time can be evaluated by $O(N_{new\_term}N_{topic})$, where N_{new_term} is the total number of new terms encountered.

As respect to sentiment comprehending, the case when all sentiment words are labeled as neutral is the most time-consuming one; the time complexity is $O(N_{new\_sentiment\_term}N_{positive}N_{negative})$, where $N_{new\_sentiment\_term}$ denotes the total number of new sentiment terms, $N_{positive}$ stands for the count of positive words and $N_{negative}$ represents the count of negative words. Furthermore, because $N_{positive}$ and $N_{negative}$ are constants, the complexity for sentiment mining is $O(N_{new\_sentiment\_term})$.

To sum up the vectoring, the step of recognizing latent topics requires the most time, i.e., $O(N_{new\_term}N_{topic})$, just as the complexity of Latent Semantic Analysis (LSA) ($O(N_{review}N_{topic}^2)$) or LDA ($O(N_{new\_term}N_{topic}T)$). However, considering the share of latent-topic-based reviews ($<$80%), GSDNT is more efficient.

After embedding, we cluster theses review vectors based on K-means. The complexity of K-means is $O(N_{review}KT)$, where K is the size of group number that can be determined beforehand and T denotes the iteration count which can also be estimated in advance (e.g., 1,000); in this manner, the time complexity is $O(N_{review})$. For spam group labeling, it requires $O(N_{review}^2)$. Therefore, GSDNT surpasses FIM, $O(N_{item}^3)$ with the minimum support = 3. Thus, the proposed method is preferable in time complexity comparing to FIM, LSA or LDA. Note that the step of clustering and labeling is not referred in comparing to LSA or LDA, because they take effect merely in review embedding and all requires this step to obtain spammed groups.

4. Experiments and discussion

We perform several experiments to verify GSDNT, including making comparisons with state-of-the-art methods. The testbed for this study is a laptop computer (Lenovo ThinkPad T490) with an Intel Core i7-8565 (1.8 GHz) CPU, Kingston 16 GB 2400 MHz DDR RAM, Lenovo SATA3 1.0 TB SSD and Intel Gigabit NIC. All the algorithms are implemented using Python v3.6.4. Third-party packages that we have employed include *os*, *re*, *numpy*, *pandas*, *matlibplot*, *jieba*, *scipy* and *sklearn*. We have uploaded the entire project to GitHub (Li, 2020).

According Definition 1-3 and Theorems 1 and 2, we first group product reviews and then generate nominated topics. The threshold for topic recognition is set to $\theta = 3$ to authorize more topics. In another words, any topic proposed by more than three items are accepted as a group topic. For sentiment comprehension, we apply *jieba* to cut Chinese sentences into terms. Our stop words are adapted from previous studies, such as those from Harbin Institute of Technology, Sichuan University and Baidu.com. Equivalent expressions to nominated topics are constructed manually. The term polarity is evaluated based on three studies performed by Tsinghua University, Hownet and National Taiwan University. After vectoring, we apply *sklearn.cluster.KMeans* to cluster them. The elbow method is employed to determine *k*. For instance, considering the distortion chart of an equivalent group in Fig. 2, we initiate $k = 10$, where distortion is significantly alleviated afterwards.

4.1. Comparison of aspect-oriented vectorizing

```
Algorithm IV: topic_mining_using_LDA
input: reviews, probability_threshold, max_topic_number
Initiate topic_number
for review in reviews:
    Cut review into terms
    Collect all terms as corpus
Vectorize corpus using TF-IDF as tfidf_matrix
review_topic_matrix ← Call LDA (n_components = topic_number) to fit tfidf_matrix
Filter review_topic_matrix with probability_threshold and max_topic_number
return review_topic_matrix
```

For the comparison, we choose LSA and LDA (Blei, Ng, & Jordan, 2003) because of their dominant adoption in topic modeling. LSA takes effect by Singular Value Decomposition (SVD), and the pseudocode of LDA is shown in Algorithm IV. Core APIs are sourced from *sklearn.decomposition*, such as *TruncatedSVD* and *LatentDirichletAllocation*. Note that the topic number (n_components) is set according to the corresponding value of the equivalent group. When the *review_topic_matrix* (rows: reviews; columns: probability to cover this topic) is ready, we use two thresholds (*probability_threshold* and *max_topic_number*) to discover any latent topics. First, any probability that is equal to or less than the mean of the current review vector is directly set to zero; second, if there are more nonzero values than *max_topic_number*, we aggressively clear the tailing values (to zeros). The threshold of *max_topic_number* is set according to the maximum number found in the nominated-topic-based method. Last, we adopt the K-means algorithm once more to cluster the vectors.

Comparisons are conducted between the clusters vectorized by LSA, LDA and GSDNT. Metrics include the CH (Calinski-Harabasz) index, the SC (silhouette coefficient) and SSE, which are highlighted in the evaluation phase. CH is the ratio of the inter-cluster distance to the inner-cluster distance, and $CH_K = [tr_B/(K-1)]/[tr_W/(N-K)]$, where K is a cluster out of N clusters, tr_B denotes the trace of the distance difference matrix between clusters and tr_W denotes the trace of the distance difference matrix within cluster K. Furthermore, $tr_B = \sum_{j=i}^{k}\|z_j - z\|^2$ and $tr_W = \sum_{j=1}^{k}\sum\|x_i - z_j\|^2$, where z_j is the mean distance of cluster c_j, z is the overall mean of all distances and x_i is the inner-cluster distance between two samples. We can easily conclude that a larger CH represents better performance. For every individual review, its SC can be drawn from two aspects: 1) the averaged distance to any other samples within the same cluster; 2) the averaged distance to any samples of the nearest cluster. Specifically, $SC_K = (b-a)/\max(a,b)$ and the mean value of all the reviews can be concluded to be the coefficient of the entire corpus. Similarly, a larger value of SC implies better performance. Specifically, −1 means incorrect, 0 stands for overlapping and 1 denotes perfect. We also apply SSE to evaluate the condensation level. $SSE = \sum_{j=1}^{r}\sum_{j=1}^{n_i}(X_{ij} - \overline{X_i})^2$, where $\overline{X_i} = \sum_{j=1}^{n_i}X_{ij}/n_i$ and $i = 1, 2, \ldots, r$. For a performance comparison, a smaller SSE is better. After running on twelve equivalent groups, a comparison of vectoring is collected in Table 2.

From Table 2, we can clearly observe that GSDNT outperforms LSA and LDA on CH and SC, but not on SSE. Suppose the CH level of LSA is higher than that of our approach, i.e., $\mu_1 \geq \mu_2$; through *t*-test, we reject this hypothesis because P (4.80e-13) $< \alpha$ (0.05). We also arrive at the knowledge that GSDNT has a greater level of SC for P (1.39e-18) $< \alpha$ (0.05). Comparison between LDA and GSDNT has the same result ($P =$ 1.02e-10 for CH and $P =$ 6.92e-27 for SC). Both CH and SC cover inter-cluster distances and intra-cluster distances, where CH highlights the ratio of the inter-cluster distances to the intra-cluster distances, and SC measures the difference between the intra-cluster distance and the distance to the nearest cluster. Therefore, the superiority of these two comprehensive metrics verifies that nominated-aspect-based review vectoring is more efficient in supporting review clustering with sparser distributed clusters and denser reviews inside.

Unlike CH or SC, the measure of SSE focuses only on an intra-cluster feature, i.e., the error of review distances. From these statistical means in Table 2, we can see that clusters derived from our method are more

Table 2
Clustering comparison (one-sided *t*-test over the hypothesis that $\mu_1 \geq \mu_2$ with $\alpha = 0.05$; *p* values $< \alpha$ are in bold).

Metric	LSA ($\bar{x} \pm S$)	LDA ($\bar{x} \pm S$)	GSDNT ($\bar{x} \pm S$)	*P* (LSA VS. GSDNT)	*P* (LDA VS. GSDNT)
CH	314.89 ± 4.48	715.66 ± 23.82	894.91 ± 32.57	**4.80e-13**	**1.02e-10**
SC	0.13 ± 0.01	0.41 ± 0.01	0.87 ± 0.00	**1.39e-18**	**6.92e-27**
SSE	2.40e + 05 ± 9.75e + 03	1.89e + 04 ± 4.39e + 04	1.45e + 05 ± 4.47e + 05	0.26	0.20

Note: For CH and SC, μ_1 represents the overall mean of LSA or LDA, and μ_2 denotes that of GSDNT; for SSE, μ_1 represents the overall mean of our approach and μ_2 denotes that of LSA or LDA.

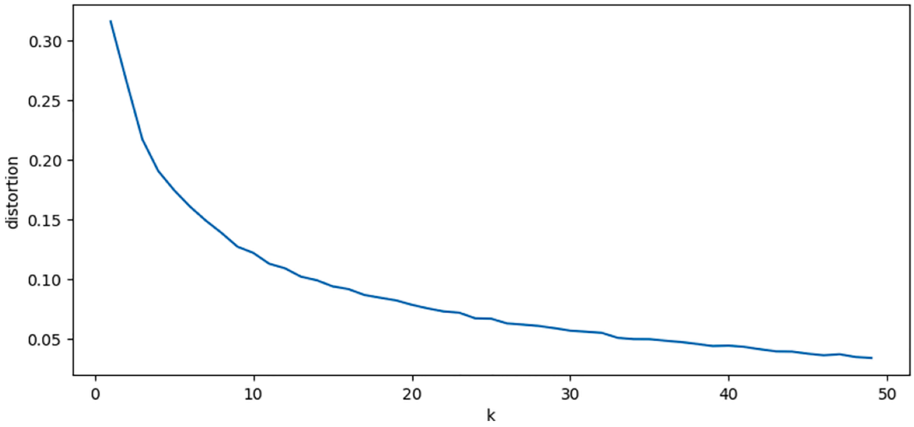

Fig. 2. An example of distortion.

condensed than those generated by LSA, but less than those contributed by LDA. For LSA, since there is no probability bias, terms/topics turn out to be evenly distributed among topics/reviews; thus, distinct sentiments adopting some commonly used words have a fairly higher possibility to be clustered together, which results in a greater SSE. In GSDNT, sentiments not only talking about the same set of topics, but also resembling in corresponding polarities can be clustered together, so the ratio of shared words is higher than LSA, rendering its SSE to a lower level. We also perform a simple test to explore the margin of LDA. The experiment simply taking an entire equivalent group as one cluster shows that the averaged distance among review vectors generated by LDA is smaller than that of the nominated-topic-based method. The reason behind this phenomenon is that LDA discovers only one topic for most reviews (59.81% vs. 11.17% of GSDNT); in most cases it results in duplicate review vectors in which the distances are small.

Detailed distribution of topic numbers found by LSA, LDA and GSDNT is depicted in Fig. 3. In this figure, topics are almost evenly labeled by LSA, reflecting its deficiency in aspect modeling. This figure also shows that the ratio of topic-less cases is quite trivial (1.12%) using LDA, which reveals the fact that LDA tends to label a review with at least one topic, even when there is no explicit sign. With the help of the nominated topics, there is a high proportion (81.59%) of vectors with no topics in our method. This phenomenon echoes the fact that the majority of reviewers tend to post quick experiences out of the profit motive (Dellarocas & Narayan, 2006; Li et al., 2020), leaving their review generic, shallow and short. From this comparison, we can conclude that topic mining guiding by officially provided topics performs better, even though LDA can discover more topics most of the time.

4.2. *Comparison of spamicity*

Once candidate groups of reviews are ready, we label them based on SI_1 and SI_2. Comparisons among LSA-contributed clusters, LDA-generated clusters and nominated-topic-oriented clusters are shown in Fig. 4 and Table 3, where the top-*n* clusters on content duplication and time burstiness are evaluated. Note that the comparison is based on the same setting, including the topic number and the cluster count. Since FIM is widely adopted in behavior-based (Mukherjee et al., 2012) or graph-oriented (Allahbakhsh et al., 2013) grouped spam labeling, we also use it validate our model with the minimum support = 3 according to previous studies; however, no groups with 2 or more reviewers are found in our dataset. The primary cause can be drawn from the fact that spammers keep registering new accounts to manipulate. This case echoes the finding that spammers are more difficult to expose than their opinions.

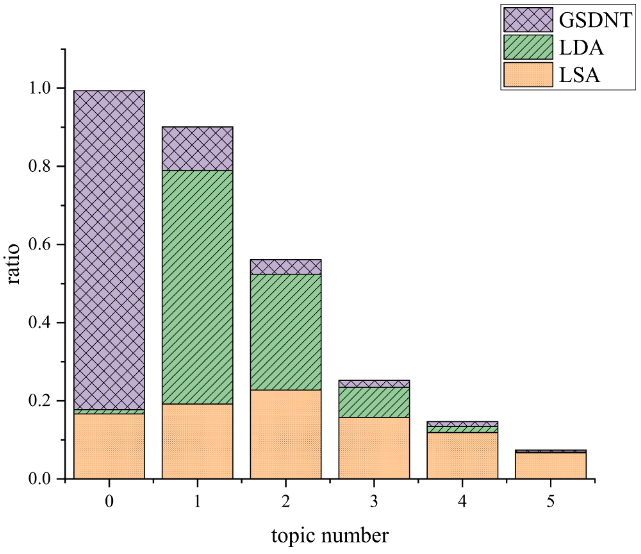

Fig. 3. The distribution of topics found.

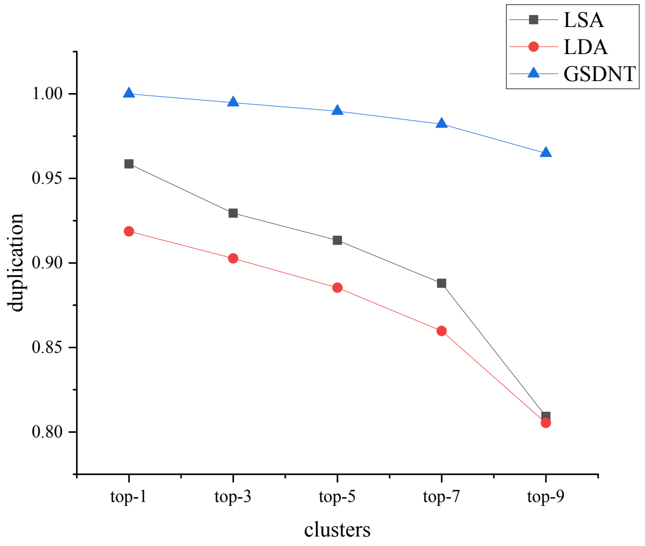

Fig. 4. Duplication comparison.

Table 3
Burstiness comparison.

Clusters	LSA	LDA	GSDNT	Margin (GSDNT over LSA)	Margin (GSDNT over LDA)
top-1	6.88e + 07	9.65e + 08	4.21e + 07	38.77%	95.63%
top-3	1.05e + 08	1.71e + 09	7.33e + 07	30.25%	95.71%
top-5	1.36e + 08	2.40e + 09	8.70e + 07	36.19%	96.37%
top-7	2.57e + 08	2.83e + 09	1.01e + 08	60.43%	96.42%
top-9	7.69e + 09	3.49e + 09	3.12e + 08	95.94%	91.07%

From Fig. 4 and Table 4, we can conclude that GSDNT surpasses LSA and LDA on the metric of duplication. Given these *P* values as 0.004 and 0.006 (<0.05), the hypothesis that LSA or LDA might have a greater duplication level than GSDNT ($\mu_1 \geq \mu_2$) cannot be trusted, so we argue that clusters generated by our method are more likely to be spammed groups. Without the basis of probability, LSA's convergence is dependent of a larger corpus together with a richer term dictionary. Without them, regular terms can be falsely taken as a special one. In that case, it opts to believe that different reviews are addressing the same topic; whereas they cover diverse sentiment aspects with only a few regular

Table 4
Spamicity comparison (one-sided *t*-test over the hypothesis that $\mu_1 \geq \mu_2$ with $\alpha = 0.05$; *p* values $< \alpha$ are in bold).

Metric	LSA $(\bar{x} \pm S)$	LDA $(\bar{x} \pm S)$	GSDNT $(\bar{x} \pm S)$	*P* (LSA vs. GSDNT)	*P* (LDA vs. GSDNT)
Duplication	37.40%± 26.98%	55.90%± 20.97%	78.30%± 13.32%	**0.004**	**0.006**
Burstiness	7.69e + 09 s ± 1.9e + 10 s	9.35e + 09 s ± 1.86e + 10 s	6.78e + 08 s ± 1.23e + 09 s	0.139	0.088

Note: For content duplication, μ_1 represents the overall mean of LSA or LDA, and μ_2 denotes that of GSDNT; for time burstiness, μ_1 represents the overall mean of our approach and μ_2 denotes that of LSA or LDA.

words in common. This tendency explains LSA's ineffectiveness on the metric of review duplication. The key reason for LDA's deficit on duplication can be drawn from the fact that it nominates topics aggressively. Given a cluster full of duplicated content, LDA appends extra reviews with different content; therefore, the duplicated level of this cluster decreases. This deficit is also the consequence of blinded topic mining. Leveraged by the nominated topics and their synonyms, GSDNT can examine a review if it is talking about a specific aspect with high accuracy; thus, the probability of over-labeling a review with extra topics is low. Although manually generated equivalent expressions are insufficient to cover arbitrary reviewers' writing habits, for spam labeling, at least it is preferable than overfitting.

For the current study, we focus only on review content to detect spam, and there is no supervision over review time in our approach. However, time concentration within a cluster generated by GSDNT is also superior to that observed by LSA or LDA, although not significant enough ($P = 0.139$ and $P = 0.088$). To analyze the reason behind, we think it results from duplication. According to previous work, reviewers (especially crowdsourcing spammers who adhere to spam efficiency) tend to copy recent experiences, which are exhibited at the top of the recommendation page. In this manner, duplicated reviews are closely posted in the time dimension, which leads to a cluster with highly duplicated reviews having a relatively low sum of errors of time differences.

Since duplication and burstiness are two efficient markers in spam labeling, we argue that our method GSDNT is more appropriate than state-of-the-art spam clustering solutions. The key ingredient for our success is the nominated topics, as they make our approach reliable and robust for sentiment understanding and review vectorizing. Clustering via the commonly used clustering method, K-means, also verifies its effectiveness on spam recognition. We also use reviewer duplication as an indicator in grouped spam detection. However, all methods harvest little (<4.4%), which validates the fact that spamming with varying accounts is now common.

From these results we can say that, GSDNT has responded to the three open challenges discussed in the Introduction section: 1) sentiment topics are direct and clear in our model; 2) because of officially nominated aspects, GSDNT achieves attractive performance on content duplicate and time burstiness; and 3) our solution is also less complicated according to the discussion in the Methodology section (time complexity subsection).

5. Conclusions

With the help of pervasively existing nominated review topics across the top two B2C websites, JD.com and TMALL.com, this paper endeavors to improve the performance of aspect-oriented sentiment mining and grouped spam detection. Our approach GSDNT has three phases: 1) preprocessing to define equivalent groups, their target topics and compatible reviews across platforms; 2) mining recommendation topics and sentiment polarities; and 3) clustering similar reviews. Experimental comparisons with FIM, LSA and LDA show that GSDNT performs better on mining suggested topics and clusters derived by our method have higher spamicity. By this study, managers of opinion sharing communities or e-business websites can unveil collaborative spam groups with more confidence; meanwhile, considering their severe damage in real world, product reputations are expected to be corrected soon. Moreover, with recommendations getting more trustable, online shoppers will hesitate less and arrive at a deal in shorter time, and the e-business market will boom once again.

A limitation of GSDNT is that it depends on the representativeness of the officially predefined topics. Although these topics are expected to be adopted by most reviewers, in fact customers can nominate any sentiment aspects. Future work can scan all reviews of an item and build a customized list of top-n aspects. It is worth noting that, n should be carefully set because of the sparseness of vectorizing. Besides, it appears to be unfair to compare GSDNT to unsupervised ones; however, currently we cannot find any supervised solutions that is more applicable than the LSA and LDA for our study. For example, labeled LDA (Ramage, Hall, Nallapati, & Manning, 2009) assumes that each document has a definite topic set; whereas in our scenario, reviewers can arbitrarily pick any topic to express their feelings. Another variant called SSHLDA (Mao et al., 2012) aims to discover more topics based on pre-observed topics; whereas in our study, we only consider sentiments on nominated topics, and there is no need to explore extra topics.

We also plan to explore the underlining differences between the pattern shaping a sincere review and the pattern fabricating a spam with Chinese text highlighted in the future. Compared to detecting English spam, one of the key challenges here lies in the fact that Chinese is less strict syntactically; therefore, lexical features are expected to be less efficient in indicating Chinese spam. Nevertheless, contrast to exclusively using permutation as glyph in English, Chinese characters are much more variable, let alone its complicated correlation with pronunciation. This fact enlightens us that, besides term permutation, shilling attacks in Chinese could also be caught from glyph footprints. Furthermore, like the gold-standard datasets contributed by Ott 2011 (Ott, Choi, Cardie, & Hancock, 2011), we plan to build a Chinese annotated review sets by recruiting volunteers.

CRediT authorship contribution statement

Jiandun Li: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Pin Lv:** Supervision, Writing - review & editing, . **Wei Xiao:** Data curation, Writing - review & editing, . **Liu Yang:** Data curation, Funding acquisition, Writing - review & editing, . **Pengpeng Zhang:** Resources, Funding acquisition, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the National Nature Science Foundation of China, under grant 61801285 and 61802247. We thank anonymous reviewers for their constructive comments.

References

Akoglu, L. (2013). Opinion fraud detection in online reviews by network effects. *Paper presented at the Seventh International AAAI Conference on Weblogs and Social Media*.

Allahbakhsh, M., Ignjatovi, A., Benatallah, B., Beheshti, S.-M.-R., Bertino, E., & Foo, N. (2013). Collusion detection in online rating systems. *Paper presented at the APWeb*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(null), 993–1022.

Chakrabarti, D., Papadimitriou, S., Modha, D., & Faloutsos, C. (2004). Fully automatic cross-associations. *KDD-2004*. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/1014052.1014064

Choo, E., Yu, T., & Chi, M. (2015). Detecting opinion spammer groups through community discovery and sentiment analysis. Paper presented at the 29th IFIP Annual Conference on Data and Applications Security and Privacy (DBSEC).

Chung-Chi Chen, H.-H. H. a. H.-H. C. (2018). NTUSD-Fin: A market sentiment dictionary for financial social media data applications. Paper presented at the the 1st Financial Narrative Processing Workshop, Miyazaki, Japan.

Dellarocas, C., & Narayan, R. (2006). What motivates consumers to review a product online? A study of the product-specific antecedents of online movie reviews. Paper presented at the WISE.

Dewang, R. K., & Singh, A. K. (2018). State-of-art approaches for review spammer detection: A survey. *Journal of Intelligent Information Systems, 50*(2), 231–264. https://doi.org/10.1007/s10844-017-0454-7

Do, Q. N. T., Hussain, F. K., & Nguyen, B. T. (2017). A Fuzzy approach to detect spammer groups. *2017 IEEE International Conference on Fuzzy Systems (Fuzz-Ieee)*. Retrieved from <Go to ISI>://WOS:000426449100100.

Dominic, D., & Lijo, V. P. (2017). Opinion spam detection using review and reviewer centric features. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (Icpcsi), 1154-1159. Retrieved from <Go to ISI>://WOS:000464679901041.

Hennig-Thurau, T., Gwinner, K. P., Walsh, G., & Gremler, D. D. (2004). Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the internet? *Journal of Interactive Marketing, 18*(1), 38–52.

Heydari, A., Tavakoli, M.a., Salim, N., & Heydari, Z. (2015). Detection of review spam: A survey. *Expert Systems with Applications, 42*(7), 3634–3642. https://doi.org/10.1016/j.eswa.2014.12.029

Hussain, N., Turab Mirza, H., Rasool, G., Hussain, I., & Kaleem, M. (2019). Spam review detection techniques: A systematic literature review. *Applied Sciences, 9*(5), 987. https://doi.org/10.3390/app9050987

Jiwei Li, C. C., Li, S. (2013). Topicspam: a topic-model based approach for spam detection. Paper presented at the the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013), Sofia, Bulgaria.

Junting Ye, L. A. (2015). Discovering opinion spammer groups by network footprints. *Paper presented at the 15 PKDD*.

Lee, K. D., Han, K., & Myaeng, S.-H. (2016). Capturing word choice patterns with LDA for fake review detection in sentiment analysis. Paper presented at the 6th International Conference on Web Intelligence, Mining and Semantics (WIMS 2016), Nimes, France.

Li, J. (2020). Code: Detecting gouped spam reviews based on officially nominated topics. Github. Retrieved from https://github.com/smellydog521/spam_group_detection.

Li, J., Wang, X., Yang, L., Zhang, P., & Yang, D. (2020). Identifying ground truth in opinion spam: An empirical survey based on review psychology. *Applied Intelligence*. https://doi.org/10.1007/s10489-020-01764-7

Mao, X.-L., Ming, Z.-Y., Chua, T.-S., Li, S., Yan, H., & Li, X. (2012). SSHLDA: A semi-supervised hierarchical topic model. Paper presented at the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea.

Mayzlin, D. (2006). Promotional Chat on the Internet. *Marketing Science, 25*(2), 9. https://doi.org/10.1287/mksc.1050.0137

Meng, J. L., Hongfei, & Yu, Y. (2011). A two-stage feature selection method for text categorization. *Computers & Mathematics with Applications, 62*(7), 2793–2800.

Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. *Paper presented at the Proceedings of the 21st international conference on World Wide Web*.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Paper presented at the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*.

Peng, Q., & Zhong, M. (2014). Detecting spam review through sentiment analysis. *Journal of Software, 9*(8). https://doi.org/10.4304/jsw.9.8.2065-2072

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Paper presented at the 2009 Conference on Empirical Methods in Natural Language Processing*.

Vidanagama, D. U., Silva, T. P., & Karunananda, A. S. (2019). Deceptive consumer review detection: A survey. *Artificial Intelligence Review, 1–30*. https://doi.org/10.1007/s10462-019-09697-5

Vlad Sandulescu, M. E. (2015). Detecting singleton review spammers using semantic similarity. Paper presented at the the 24th International Conference onWorld WideWeb Companion (WWW 2015), Florence, Italy.

Wang, Z., Gu, S., & Xu, X. (2018). GSLDA: LDA-based group spamming detection in product reviews. *Applied Intelligence, 48*(9), 3094–3107. https://doi.org/10.1007/s10489-018-1142-1

Wang, Z., Gu, S., Zhao, X., & Xu, X. (2018). Graph-based review spammer group detection. *Knowledge and Information Systems, 55*(3), 571–597. https://doi.org/10.1007/s10115-017-1068-7

Wang, Z., Hou, T., Song, D., Li, Z., & Kong, T. (2017). Detecting review spammer groups via bipartite graph projection. *The Computer Journal, 59*(6), 861–874. https://doi.org/10.1112/comjnl/bxv068

Xu, C., & Zhang, J. (2015). Towards collusive fraud detection in online reviews. *IEEE International Conference on Data Mining (Icdm), 2015*, 1051–1056. https://doi.org/10.1109/Icdm.2015.62

Xu, X., Yuruk, N., Feng, Z., & Schweiger, T. A. J. (2007). SCAN: a structural clustering algorithm for networks. Paper presented at the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, California, USA. https://doi.org/10.1145/1281192.1281280.

Zhang, L., Wu, Z., & Cao, J. (2018). Detecting spammer groups from product reviews: A partially supervised learning model. *IEEE Access, 6*, 2559–2568. https://doi.org/10.1109/access.2017.2784370