

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

J. K. Rout, K. S. Sahoo, A. Dalmia, S. Bakshi, M. Bilal and H. Song, "Understanding Large-Scale Network Effects in Detecting Review Spammers," in IEEE Transactions on Computational Social Systems, doi: 10.1109/TCSS.2023.3243139.

<https://doi.org/10.1109/TCSS.2023.3243139>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Understanding Large-Scale Network Effects in Detecting Review Spammers

Jitendra Kumar Rout^{ID}, *Member, IEEE*, Kshira Sagar Sahoo^{ID}, *Member, IEEE*, Anmol Dalmia^{ID},
Sambit Bakshi, *Senior Member, IEEE*, Muhammad Bilal^{ID}, *Senior Member, IEEE*,
and Houbing Song^{ID}, *Fellow, IEEE*

Abstract—Opinion spam detection is a challenge for online review systems and social forum operators. Opinion spamming costs businesses and people money since it deceives customers as well as automated opinion mining and sentiment analysis systems by bestowing undeserved positive opinions on target firms and/or bestowing fake negative opinions on others. One popular detection approach is to model a review system as a network of users, products, and reviews, for example using review graph models. In this article, we study the effects of network scale on network-based review spammer detection models, specifically on the *trust* model and the *SpammerRank* model. We then evaluate both network models using two large publicly available review datasets, namely: the *Amazon dataset* (containing 6 million reviews by more than 2 million reviewers) and the *UCSD dataset* (containing over 82 million reviews by 21 million reviewers). It has been observed that *SpammerRank* model provides a better scaling time for applications requiring reviewer indicators and in case of *trust* model distributions are flattening out indicating variance of reviews with respect to spamming. Detailed observations on the scaling effects of these models are reported in the result section.

Index Terms—Online review spam, opinion spam, review graphs, spam detection, unlabeled review.

I. INTRODUCTION

ONLINE opinions are textual data left by users on online sites, such as review platforms. These opinions help other users to determine the quality and other characteristics of an object (e.g., movie, and consumer product), and also provide important data to owners and competitors to design future business plans. Hence, not surprisingly, this is a topic of interest for e-commerce organizations, as well as other review platforms, since the rating and reviewing of products

Manuscript received 1 September 2022; revised 10 December 2022; accepted 2 February 2023. (Corresponding author: Muhammad Bilal.)

Jitendra Kumar Rout is with the Department of Computer Science and Engineering, National Institute of Technology Raipur, Raipur 492010, India (e-mail: jitu2rout@gmail.com).

Kshira Sagar Sahoo is with the Department of Computing Science, Umeå University, 901 87 Umeå, Sweden (e-mail: ksahoo@cs.umu.se).

Anmol Dalmia is with the Senior Software Development Engineer, Seattle, WA 98101 USA (e-mail: dalmia.anmol@gmail.com).

Sambit Bakshi is with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, Odisha 769008, India (e-mail: bakshisambit@ieee.org).

Muhammad Bilal is with the Department of Computer Engineering, Hankuk University of Foreign Studies, Yongin-si 17035, South Korea (e-mail: m.bilal@ieee.org).

Houbing Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA (e-mail: h.song@ieee.org).

Digital Object Identifier 10.1109/TCSS.2023.3243139

2329-924X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

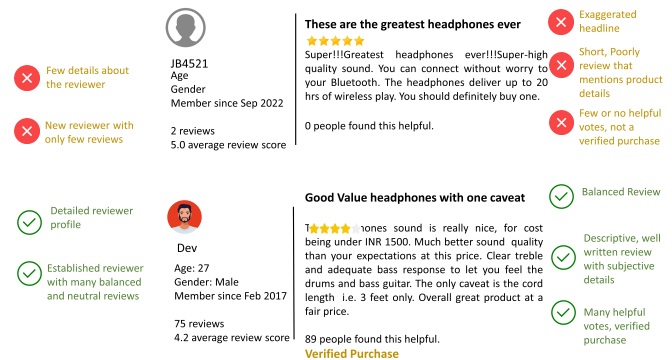


Fig. 1. Sample deceptive and genuine review.

and services have significant financial implications [1]. For example, Yelp is a system dedicated to hosting reviews for restaurants and other local businesses in various countries. Another similar system is Amazon, which hosts reviews about particular products sold on Amazon. Typical features of such systems include allowing users to rate the product or service, provide feedback, and like/dislike certain products or services.

Reviews can, however, be fabricated (also referred as spam reviews), in the sense that users (also referred to as review spammers) can be paid to provide glowing feedback or to post negative comments about a target product or service. These spam reviews might also be called as fake/deceptive/nongenuine/fraudulent reviews. Some of the hypothetical differences between deceptive reviews and genuine ones are: deceptive reviews contain more words (more quantity), a more average number of words per sentence (more complexity), frequent mention of brand names whereas truthful reviews contain more number of unique words (diversity), more average word length, and more number of digits. Apart from these truthful reviews may contain more of nouns, prepositions, adjectives, determiners, and coordinating conjunctions while imaginative writing have more pronouns, verbs, adverbs, and predetermines. In addition, imaginative writing has more number of connectors such as and/or/however and more self-referencing words i.e., immediacy). An example for a sample review is given in Fig. 1.

The review spammers may work individually or in groups. Rookie spammers (or bots) may use the same or similarly worded reviews for multiple sites, while the more sophisticated ones may customize the content to avoid detection. There are a number of challenges in identifying review spam and

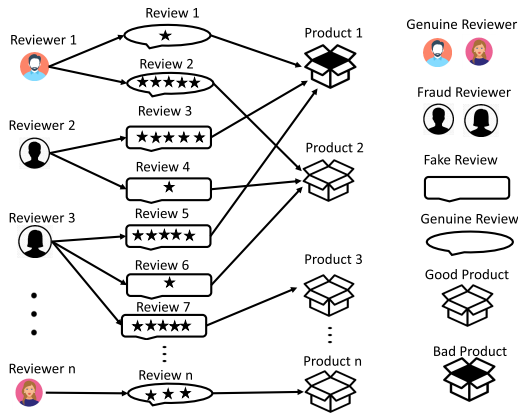


Fig. 2. Graphical representation of a product review system.

spammers, such as lack of ground truth (that can be used to train spam detection models), variance in writing and framing styles, and the possible interleaving of spammer and consumer behavior [2], [3].

There have been attempts to design solutions to detect either fake reviews or review spammers, for example using machine learning (ML). A popular approach is to visualize the reviewing system as a network of reviews and reviewers [4], [5], [6], [7], [8], [9]. Thus, this work investigates the behavior of graph-based models in detecting spam reviewers on large-scale reviewing systems. Here, *graph* and *network* are used synonymously. A graphical representation of the product review system is given in Fig. 2.

The major contributions of the article are as follows.

- 1) To study the feature- and structure-based review graph algorithms when applied to large datasets of reviews.
- 2) To study the effect of network scaling using various network-based review spammer detection models such as trust model ([3]) and the SpammerRank ([10])
- 3) To evaluate network models using two large publicly available review datasets such as the Amazon dataset [10], [11] (6 million reviews, 2 million reviewers) and the University of California San Diego (UCSD) dataset [12], [13] (82 million reviews, 21 million reviewers)
- 4) To compare the scalability of both models in terms of performance cost.

The rest of the article is organized as follows. We first review the related approaches in Section II, before introducing the review graph models to be studied in Section III. The datasets to be used in our evaluation are described in Section IV. Sections V and VI describe the experiment setup and findings, respectively. Finally, we conclude this article in Section VII.

II. RELATED APPROACHES

Detection of deceptive opinions was formally introduced by Jindal and Liu [2]. Since text mining and text processing have been successfully applied to tackle multiple problems across various disciplines, most notably recommender systems and sentiment analysis and even to detect spam in social platforms [14], detection of fake reviews are usually centered at

text analysis [15]. Several models have been proposed to detect spam reviews based on text similarity and duplication, psycholinguistic features, sentimental content, graph-based [4], [16], and time-series based [17]. Most of the works are based on supervised learning [18], [19]. A major hurdle for these works is the availability of sufficient labeled data, which inspired advances based on semi-supervised learning [20], [21]. Apart from detecting fake opinions contained in spam reviews, approaches for detecting spammers have also been suggested. These are broad of two kinds: detecting individual spammers [22], [23], [24], [25] and detecting groups or clusters of cooperating spammers [26], [27], [28], [29], [30], [31], [32], [33], [34]. Various characteristics of reviews such as relevant metadata and content-based features are used to achieve this objective.

In metadata-based approaches, a popular one is the use of rating data to classify reviewers as spammers. There exist models based on deviation from the average rating for a given product by various reviewers [35], [36], [37], [38]. Apart from solely using rating deviations as spam indicators, additional complementary features have also been exploited by Lim et al. [22], Mukherjee et al. [26], [27], and Sharma and Lin [39]. These ideas have also inspired other applications of rating information such as the formulation of various reputation systems [40]. In addition to rating data, the timeliness of the reviews can also be exploited for classification purposes. Spammers usually post spam reviews during a specific time interval (burst) during which the number of reviews and other related events rise drastically. This assumption supported by observations across other similar systems such as social networks has been used by many researchers to detect spammers. Xie et al. [41] proposed a model that can be used to detect spamming activity of reviewers who have few reviews by analyzing the unusual temporal review patterns. Similarly, Fei et al. [42] focused on the burst patterns in reviews for a specific product using kernel density estimation techniques to detect burstiness intervals along with behavioral features. This model focused only on behavioral features and ignores textual features which might improve the accuracy for spammer detection. The use of other crucial features like purchase verification has also been suggested for better performance [41]. However, such features might not be applicable to all reviewing systems. Another weakness of such methods with sole focus on burst patterns is missing those spammers who have posted sporadically or have posted uniformly many reviews for target products over long periods of time.

The discussed attempts mainly make use of information related to *reviews* only. However, relationships between reviews, reviewers, and products may shed light on detecting spammers as well [43]. Such relationships have been taken into account by devising graph-based approaches [44], [45], [46]. Wang et al. [3], [47] proposed the first graph-based approach to identify review spammers using store review data from www.resellerrating.com. A heterogeneous review graph was used to capture the said relationships. The key parameters taken into consideration were: the trust of reviewers, the honesty of reviews, and the reliability of stores. The demerits

of the approach are that it did not incorporate review content information, and that it is not linearly scalable and hence applicable to only small graphs. A similar approach was proposed by Fayazbaksh and Sinha [48] which differs from Wang et al. [3] in the score make-up and initialization methods. Also, the proposed algorithm is non-iterative in nature.

To overcome the limitations of Wang et al. [3], [47] (i.e., missing content information and scalability), Akoglu et al. [49] proposed the generalized and network-based FRAUDEAGLE framework claimed to be effective, unsupervised, and scalable in nature. The approach exploits the network effects among reviewers and products for a given reviewing system. Liang et al. [10] also presented a novel multiedge graph model which integrates the reviewers' features and their relationships with reviewers' unreliability scores. The authors proposed an unsupervised iterative computation framework for detecting spammers who always work cooperatively. Combining ML and graph-based approaches, Lu et al. [50] proposed an algorithm in a supervised framework, which they claimed to be the first one of its kind to detect both fake reviews and review spammers at the same time. Features related to both reviews and reviewers were extracted and incorporated in a *review factor graph* to leverage belief propagation between reviews and reviewers. One of the issues with the work is that the indicators used to label reviews as spam are not adequate. Noekhah et al. [51] extended this work and proposed an iterative algorithm to detect fake reviews, review spammers, and group of spammers at the same time. The authors have proposed a graph that considers all the features and entity relationships between reviews and reviewers. The works related to spammer detection can be broadly visualized by Fig. 3.

As is evident, all the approaches discussed make use of different features extracted from instances of different datasets. This makes comparison of such models very difficult. Though such comparisons have been made in some works [35], they have been performed on reduced or condensed datasets in general. This has inspired the proposed work to study the effects of scale on select approaches when applied over full-scale datasets.

III. REVIEW GRAPH MODELS

In this work, review graph models have been considered to be classified into two wide groups:

1) *Structure Driven*: In these models, parameters associated with the nodes and/or edges are evaluated on the basis of the structure of the review network only.

2) *Feature Driven*: In these models, in addition to the structural properties of the network, features of the nodes and edges are also employed in evaluating the parameters associated with the nodes and/or edges.

We take one model from each of these groups and apply it to the Amazon product review datasets mentioned in Section IV. For the structure driven models, that presented by Wang et al. [3] has been chosen for evaluation, while among the feature-driven models, the approach defined by

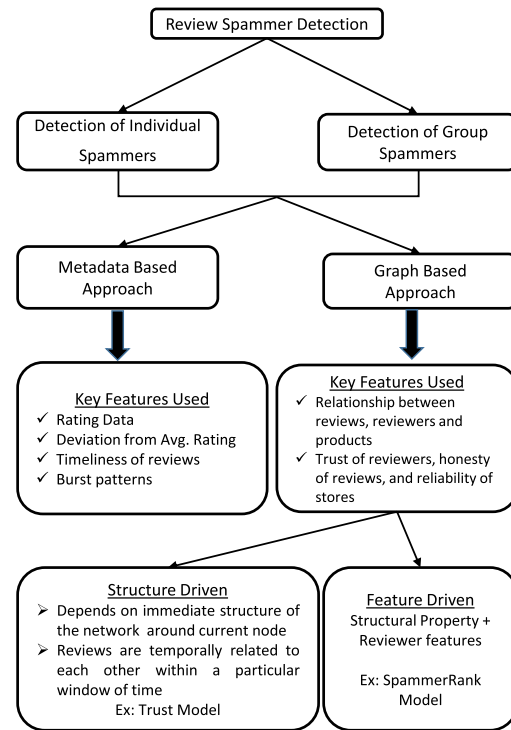


Fig. 3. Classification of related work done on review spammer detection.

Liang et al. [10] has been considered. For ease of reference to models in this work, the model described by Wang et al. is called the *trust model* whereas the one presented by Liang et al. has been called as the *SpammerRank model*. Each of these models has been explained in Sections III-A and III-B.

A. Trust Model

The trust model of review graphs was developed by Wang et al. [3]. The approach has been applied to analyze reviewing systems for online stores. The model has been adapted to work for product reviews in this work and has been discussed as follows.

In this approach, the complete reviewing system is regarded as a heterogeneous and directed graph. The reviewers, reviews, and products of the reviewing system form the nodes of the graph. Each node is associated with a parametric value that characterizes the *benignity* of the node in the system. These are called the *benignity indicators* for the nodes in this work. The benignity indicators for the reviewer, the review, and the product nodes are *reviewer trust*, *review honesty*, and *product reliability*, respectively. In [3], these values have been evaluated in the range $[-1, 1]$ but for this work, they have been linearly translated to the range $[0, 1]$ for ease of comparison and representation. The edges in the graph are unweighted and take their meanings as infrastructure-driven connect. Their significance has been described in Section III-A1.

In this model, the benignity feature-driven particular node is evaluated using those of its neighboring nodes. This calculation only depends on the immediate structure of the network around the current node and thus, makes this model structure driven.

1) *Relationships in the Graph*: Broadly, there are three types of relationships in this review graph model.

a) *Reviewer–Review Edges*: These edges signify the writing of a review by a reviewer. An edge exists from a reviewer node to a review node in the event of the reviewer writing the particular review.

b) *Review–Product Edges*: These edges signify the subjection of a product by a review. An edge exists from a review node to a product node in the event of the review being written for the particular product.

c) *Review–Review Edges*: These edges relate to two review nodes on the basis of their agreement and closeness in the time scale of the reviewing system. Such edges are of two types: *agreeing* and *disagreeing*.

While the edges between reviewer nodes and review nodes, and review nodes and product nodes are intuitive in meaning, those between reviews carry a special meaning and are computed based on the characteristic of the reviews. For a given product, a review can be thought of as an event in the timeline of the product that brings about a change in the prevalent opinion of the product. Now, it is easy to mark that not all reviews *influence* each other. For example, a review written 2 years ago might not be as influential to a recent review as one which was written 1 month back. Thus, reviews are inspired by other reviews which are placed within a particular time frame relative to each other. In this work, such a relationship between reviews is called as the *surroundedness* of a particular review. In a broader term, it can be said that some reviews are *temporally related* to each other within a particular window of time.

Temporal relatedness may again be differentiated by the difference in the opinions' sense the respective reviews represent. Each review can have a certain *polarity* about the product/service described. For example, some reviews may voice a negative opinion about a product while, at the same time, another review presents a positive idea about the product. Some reviews may *agree* with each other while others may *disagree* among them. Although this degree of agreement can be mined using natural language processing (NLP) techniques over the reviews, Wang et al. [3] suggest making use of the ratings of the reviews as the direct indicators of such polarities. It can be assumed that if the ratings of two temporally related reviews differ by a high margin, they disagree with each other. On the other hand, if they have similar ratings, they tend to agree with each other. Thus, the review-review relationships can be further refined by saying that some reviews agree with each other and some do not. This is inferred by surroundedness and rating differences for a particular review set.

2) *Notations Used*: Section III-A1 described the various nodes and edge properties in the graph. In this model, a review is represented by v , a product by s , and a reviewer by r . The *trust* of reviewer r , *honesty* of review v and *reliability* of product s are represented as $T(r)$, $H(v)$, and $R(s)$, respectively. The rest of the symbols and their meanings have been organized under Table I. Wang et al. [3] propose an exponential normalization function to describe the nature of the benignity indicators. We denote it by $\text{norm}(x)$ as defined in

TABLE I
SYMBOLS USED FOR THE TRUST MODEL

Symbol	Meaning
v	a review
s	a product
r	a reviewer
$T(r)$	Trust of reviewer r
$H(v)$	Honesty of review v
$R(s)$	Reliability of product s
$A(v)$	net agreement for the review
$A_n(v)$	normalized $A(v)$
$R(s)$	Reliability of s
α_r^s	i^{th} review by r
β_r^s	Set of r 's reviews on s
n_r	Number of reviews written by r
H_r	Sum of honesty values of all the reviews written by r
κ_v	Reviewer who wrote v
Γ_v	Product for which v was written about
Ψ_v	Rating borne by v
S_v	Set of reviews temporally related to v
$S_{v,a}$	Set of agreeing reviews temporally related to v
$S_{v,d}$	Set of disagreeing reviews temporally related to v
t_v	Timestamp borne by v
U_s	Set of reviews written for s

the following equation:

$$\text{norm}(x) = \frac{2}{1 + \exp(-x)} - 1. \quad (1)$$

The following section presents the mathematical treatment of the trust review graph model.

3) *Computational Description*: The formal treatment of the trust model has been illustrated in [3] which has been concisely presented as follows.

The trust for a reviewer r is given by the following equation:

$$T(r) = \text{norm}(H_r). \quad (2)$$

To calculate a review v 's honesty, the surrounding set of temporally related reviews is calculated as per the following equation:

$$S_v = \{i : \Gamma_i = \Gamma_v, |t_i - t_j| \leq \Delta t\} \quad (3)$$

where Δ is the threshold temporal difference. For the threshold rating agreement difference δ , the set S_v is partitioned into the disjoint sets of agreeing and disagreeing reviews as illustrated in the following equation:

$$S_{v,a} = \{i : |\Psi_i - \Psi_j| \leq \delta\} \quad (4)$$

$$S_{v,d} = S_v - S_{v,a}. \quad (5)$$

The net agreement for the review is then calculated and normalized as in (6) and (7); and finally, the honesty is calculated as in the following equation:

$$A(v) = \sum_{i \in S_{v,a}} T(\kappa_i) - \sum_{j \in S_{v,d}} T(\kappa_j) \quad (6)$$

$$A_n(v) = \text{norm}(A(v)) \quad (7)$$

$$H(v) = |R(\Gamma_v)| \times A_n(v). \quad (8)$$

Reliability of s is then calculated using (9). In the expression, μ represents the median of the rating system used for the reviews.

$$R(s) = \text{norm} \left(\sum_{v: v \in U_s, T(\kappa_v) > 0} T(\kappa_v) \times (\Psi_v - \mu) \right). \quad (9)$$

Finally, all the benignity indicators have been translated to the range $[0, 1]$ using the min-max normalization technique.

Because of a circular dependency of the benignity indicators among each other, Wang et al. [3] proposed an iterative algorithm to calculate the said parameters. In this work, an optimized framework for the same has been formulated by clustering similar operations together to improve the locality of reference in the computations. The framework has been represented in Algorithm 1. The complexity of the algorithm has been determined to be $\mathcal{O}(\text{maxIterations} \times (n_{\text{reviews}} + n_{\text{reviewers}} + n_{\text{products}}))$, where maxIterations is the final value of roundCounter in Algorithm 1, n_{reviews} is the number of reviews, $n_{\text{reviewers}}$ is the number of reviewers, and n_{products} is the number of products.

Algorithm 1 Trust Model-Modified Framework

INPUT: The set of Reviewers, \mathcal{R} , the set of reviews \mathcal{V} , the set of products \mathcal{S}

OUTPUT: Reviewers' Trust T , Reviews' Honesty H , Products' Reliability R

```

1: Set  $T(r) = 1.0 \forall r \in \mathcal{R}$ 
2: Set  $R(s) = 1.0 \forall s \in \mathcal{S}$ 
3:  $\text{roundCounter} = 0$ ;
4: while  $\text{roundCounter} < \text{rounds}$  do
5:   for each  $v \in \mathcal{V}$  do
6:     Compute  $H(v)$  using Equation 7;
7:   end for
8:   for each  $r \in \mathcal{R}$  do
9:     Compute  $T(r)$  using Equation 2;
10:  end for
11:  for each  $s \in \mathcal{S}$  do
12:    Compute  $R(s)$  using Equation 9;
13:  end for
14:   $\text{roundCounter}++$ ;
15: end while
16: Apply min-max normalization to  $T$ ,  $H$ , and  $R$ 

```

B. SpammerRank Model

Similar to the trust model, the SpammerRank model describes the reviewing system as a review graph. This model was developed by Liang et al. [10] and uses a link-ratio-based metric called *SpammerRank* value for each reviewer as her/his spammer-like indicator. The SpammerRank metric is inspired by the PageRank algorithm [52] used to rank nodes in a network.

In this modeling approach, the review system is modeled as a heterogeneous and directed multigraph. The initial graph consists of reviewers, reviews, and products. An edge exists between a reviewer and a review if the reviewer has written the review. Similarly, an edge exists between a review and a product if it is written about that product. Each of the review nodes is tagged with the rating the review bears. This graph is tripartite in nature. After this graph is constructed, it is reduced to another graph consisting of only reviewer nodes.

This is the final review graph for this model and is used to compute the SpammerRank metric of spammerness for each node. Another feature of this model is its distinction between products and reviewers. Those products in the graph which have been reviewed by only one reviewer are called as *outlier products*. Also, those reviewers who have reviewed only some outlier products are distinguished as *outlier reviewers*. The model taxes the outlier reviewers and their associations for being more suspicious than normal reviewers. Also, in addition to taking the structural properties of the review graph into account while computing the SpammerRank values, an aggregation of features extracted from the review system for each reviewer is also considered. This makes the SpammerRank model a feature-driven one.

Section III-B1 presents a brief discussion about reducing the initial graph to the final review graph while the notations used for this model have been explained in Section III-B2. The formal methods of computing the SpammerRank measure have been elucidated in Section III-B3.

1) *Graph Reduction:* As introduced, the SpammerRank model constructs an initial tripartite graph from the given review system's information. This initial graph is then reduced to a directed weighted multigraph as follows.

First, different polarities of the review nodes are decided by the values of the ratings they bear. The ratings 1 and 2 are considered as bad, 3 is considered to be average, and 4 and 5 are considered to be good ratings. But to make this distinction more complete and to conform to the numerous reviewing systems that support intermediate rating values, like 3.5, we consider the distinctions to be spread over the continuous values rather than just considering the integral values. In other words, in the proposed work, the ratings in the range $[1, 2]$ are considered to be *bad*, and those in the range $(2, 4)$ are considered to be *average*, and those in the range $[4, 5]$ are assumed to be *good* ratings. Based on these rating classes, edges are drawn between reviewer nodes of the graph. An edge is drawn between two reviewers only if they have mutually reviewed a common product, that is if a common product node is reachable from both the reviewer nodes. This edge may be either *conflicting* or *supportive* in nature. This nature is decided as follows: the edge is supportive if both the reviewers have rated the product as good or bad, or the edge is conflicting if one of the reviewers has rated the product good whereas the other has rated it as bad. Since there may exist several such common products between any pair of reviewer nodes, an edge is drawn between the two nodes for each such product. Such edges are reciprocated by both the nodes, that is, if a reviewer-reviewer edge exists from reviewer a to reviewer b , then its reciprocation from b to a is also drawn. Finally, when all such edges have been drawn, all the review and product nodes along with the associated edges are dropped from the graph. The resultant graph is the final review graph that is processed. In this graph, all those reviewer nodes which are disconnected are the outlier reviewers and are processed separately. This process renders multiple edges drawn among all the reviewer nodes. For a given product, edges are drawn between all pairs of reviewers who have reviewed it, with multiple edges being drawn for each review

TABLE II
SYMBOLS USED FOR THE SPAMMERRANK MODEL

Symbol	Meaning
r	reviewer r
p	product p
$SDN(r)$	Deviation of r 's review count from the global average review count
$RF(r)$	Ratio of count of first reviews to that of total reviews by r
$RO(r)$	Fraction of products reviewed by r which were outlier products
$ASDR(r)$	Average difference between $AVRO(r, p)$ and $AVRP(r, p)$
$SDRG(r)$	Deviation of r 's $AVRR$ value from the global average $AVRR$ value
$ANH(r)$	Average number of unhelpful feedbacks on all products reviewed by r
$ARH(r)$	Average fraction of unhelpful feedbacks on all products reviewed by r
$BF(r)$	Binary feature indicating for if r has given ratings that all belong to a single rating class
$\beta_{sup}(r)$	represents the support for the given reviewer by fellow reviewers
$\beta_{con}(r)$	denotes the impedance for the given reviewer by fellow reviewers
$score_{features}(r)$	static score for the reviewer r
\mathcal{R}	The set of all reviewers
$R(r)$	Set of reviews written by r
$P(r)$	Set of products reviewed by r
$FR(r)$	Set of all first reviews written by r on various products
$OP(r)$	Set of outlier products reviewed by r
$SP(r)$	SpammerRank measure for the reviewer r
$E_{sup}(r)$	Set of supporting edges that relate to r
$E_{con}(r)$	Set of conflicting edges that relate to r
$AVRR(r)$	Average rating over all reviews written by r
$AVRO(r, p)$	Average rating given by reviewers other than r on p
$AVRP(r, p)$	Average rating given by r on p
$HF(r, p)$	Number of helpful feedback counts received by r on p
$NF(r, p)$	Number of unhelpful feedback counts received by r on p

if a reviewer has reviewed the said product multiple times. This makes the resultant review graph a dense multigraph. A more detailed account can be found in [10].

2) *Notations Used:* The parameters in the review graph in this model are the various aggregate measures of all the reviewers and the final SpammerRank metric. Initially, aggregate statistics are calculated for all the reviewers—measures which are inferred from the global view of the reviewing system. Using the aggregation factors, a *static score*, $score_{features}$, for all reviewers is calculated. For a reviewer, r and a product p , the notations used are defined in Table II.

3) *Computational Description:* After deriving the review graph, the SpammerRank values are computed. First, the features for a reviewer r and a product p are calculated using (10) through (17). These expressions have been modified for better approximations as compared to those presented by Liang et al. [10].

- 1) Deviation of r 's review count from the global average review count

$$SDN(r) = \left| |R(r)| - \frac{\sum_{r' \in \mathcal{R}} |R(r')|}{|\mathcal{R}|} \right|. \quad (10)$$

- 2) Ratio of count of first reviews to that of total reviews by r

$$RF(r) = \frac{|FR(r)|}{|R(r)|}. \quad (11)$$

- 3) Fraction of products reviewed by r which were outlier products

$$RO(r) = \frac{|OP(r)|}{|P(r)|}. \quad (12)$$

- 4) Average difference between $AVRO(r, p)$ and $AVRP(r, p)$

$$ASDR(r) = \frac{\sum_{p \in P(r)} |AVRO(r, p) - AVRP(r, p)|}{|P(r)|}. \quad (13)$$

- 5) Deviation of r 's $AVRR$ value from the global average $AVRR$ value

$$SDRG(r) = \left| AVRR(r) - \frac{\sum_{r' \in \mathcal{R}} AVRR(r')}{|\mathcal{R}|} \right|. \quad (14)$$

- 6) Average number of unhelpful feedbacks on all products reviewed by r

$$ANH(r) = \frac{\sum_{p \in P(r)} |NF(r, p)|}{|P(r)|}. \quad (15)$$

- 7) Average fraction of unhelpful feedbacks on all products reviewed by r

$$ARH(r) = \frac{\sum_{p \in P(r)} \left(\frac{NF(r, p)}{HF(r, p) + NF(r, p)} \right)}{|P(r)|}. \quad (16)$$

- 8) Binary feature—an indicator for if r has given ratings that all belong to a single rating class

$$BF(r) = \begin{cases} 1, & \text{if the reviewer gave only one type of rating} \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

After calculation of these features, the static score for the reviewer r is calculated with the following equation by aggregating the computed feature scores:

$$score_{features}(r) = \frac{1}{8} (SDN(r) + RF(r) + RO(r) + ASDR(r) + SDRG(r) + ANH(r) + ARH(r) + BF(r)). \quad (18)$$

Finally, the SpammerRank value is calculated by exponential smoothing with smoothing factor α , using (19) through (21)

$$\beta_{sup}(r) = \left[\sum_{r_j: (r, r_j) \in E_{sup}(r)} \frac{SP(r_j)}{|E_{sup}(r_j)|} \right] \quad (19)$$

$$\beta_{con}(r) = \left[\sum_{r_j: (r, r_j) \in E_{con}(r)} \frac{SP(r_j)}{|E_{con}(r_j)|} \right] \quad (20)$$

$$SP(r) = \alpha \cdot \left[\frac{\beta_{sup}(r)}{\exp(\beta_{con}(r))} \right] + (1 - \alpha) \cdot score_{features}(r). \quad (21)$$

In this work, the factors $\beta_{sup}(r)$ and $\beta_{con}(r)$ have been separated from the computation of the SpammerRank value in contrast to that presented in [10]. This is done to perform parallel computations for the model and improve performance. $\beta_{sup}(r)$ represents the support for the given reviewer by fellow reviewers while $\beta_{con}(r)$ denotes the impedance for the given reviewer by fellow reviewers.

Algorithm 2 SpammerRank Model- Modified Framework

INPUT: The set of Reviewers, \mathcal{R} , the set of reviews \mathcal{V} , the set of products \mathcal{S}

OUTPUT: SpammerRank, SP

```

1: Construct review graph,  $G$ , as described in Section III-B1
2: Identify outlier reviewers set,  $\mathcal{R}_{outliers}$ 
3: Set  $\mathcal{R}' = \mathcal{R} - \mathcal{R}_{outliers}$ 
4: Compute  $score_{features}(r) \forall r \in \mathcal{R}'$  using Equation 18
5: Set  $SP(r) = BF(r) \forall r \in \mathcal{R}_{outliers}$ 
6: Set  $SP(r) = score_{features}(r) \forall r \in \mathcal{R}'$ 
7:  $roundCounter = 0$ ;
8: while  $roundCounter < rounds$  do
9:   for each  $r \in \mathcal{R}'$  do
10:    Update  $SP(r)$  using Equation 21;
11:   end for
12:    $roundCounter++$ ;
13: end while

```

The computational framework for calculating SpammerRank values is iterative as well and is illustrated by Algorithm 2. The framework has been modified by processing outlier reviewers separately to avoid unnecessary calculations otherwise incurred. The complexity of the algorithm has been determined to be $\mathcal{O}(\maxIterations \times (n'_{reviewers}))$, where \maxIterations is the final value of $roundCounter$ in Algorithm 2, and $n'_{reviewers}$ is the number of reviewers excluding outliers.

IV. DATASETS

To study the effects of scale of the subject review systems for the algorithms in this work, two of the largest available datasets for research in this field were chosen.

The Amazon review dataset was prepared by Jindal and Liu [2]. The dataset contains reviews about numerous products from amazon.com from categories such as music, books, DVDs, and so on. In addition to the review content, the dataset also contains metadata such as posting timestamp, rating, reviewer ID, product ID, and so on. It consists of nearly 6 million reviews by more than 2 million reviewers. This dataset has been referred to as the *Amazon dataset* in this article, as has been done by Liang et al. [10]. The second dataset used for the proposed work is that curated by McAuley et al. [12], [13]. This dataset has been referred to as the *UCSD dataset* in this article. As of this writing, this dataset is the largest available collection of reviews. Similar to the Amazon dataset, the UCSD dataset also contains relevant metadata for processing purposes. The original dataset contains over 140 million reviews from Amazon but the authors suggest heavy presence of duplicates. For this purpose, they have released a cleansed version of the dataset which has been obtained after the de-duplication of subject products. This de-duplicated dataset contains over 82 million reviews by 21 million reviewers from amazon.com.

To study the effects of scaling in review graphs for the algorithms exploited, these datasets were chosen as they are the largest available openly. Moreover, the Amazon dataset

TABLE III
SUMMARY OF VARIOUS DATASETS USED

Dataset	Amazon	UCSD (De-duplicated)
Attributes	Reviews with metadata about various products on amazon.com	
Dataset Size	6.31 GB	58.93 GB
No. of Reviews	5,838,049	82,677,091
No. of Reviewers	2,146,064	21,176,519
No. of Products	1,230,908	9,874,208
Rating Scale Used	{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5}	{1, 2, 3, 4, 5}

was introduced in 2008 while the UCSD dataset contains reviews from the period 1996–2014. Thus, both datasets are snapshots of the same reviewing system across time and constitute an excellent platform for studying the network effects in growth. A comparative summary of both datasets has been presented in Table III.

This work is an attempt to standardize the use of the exploited datasets in the field of fake opinion detection. The greatest challenge in this field is the lack of annotated datasets. The “gold-standard” dataset available was prepared by Ott et al. [53] and is synthetic in nature. Also, this dataset does not contain any metadata, thus making it irrelevant to be used in metadata based models. As has been demonstrated in Section II, most works make use of different datasets. This makes comparing works difficult if not impossible. The datasets used in this work contain a multitude of metadata which make them suitable to be used in metadata based models. Therefore, this work is a step toward standardizing the use of the exploited datasets for developing and comparing detection models for the sake of comparison of performance.

V. EXPERIMENTAL SETUP

The models discussed in Section III were evaluated on the datasets mentioned in Section IV. This composed of a total of 4 experiments across the models and datasets. The experiments were all carried out on a 12-core Intel Xeon E2620 processor-based machine with 64 GB memory in Linux environment. Python 3.5 was used to implement the experimental systems and the principal database used to store and manipulate the mined data was MySQL.

Since both the datasets used contained reviews based on the Amazon rating systems, the median rating value, μ , for the trust model was set at 3 for the 5-star rating system. Also, as suggested by Wang et al. [3], the threshold rating difference, δ , was set to unity while the threshold period of surrounding reviews, Δ , was set to 1 month. Since the processing involved use of UNIX timestamps, this is equivalent to 2 592 000 UNIX seconds. The number of rounds was set to 10 iterations which were determined experimentally. Similarly, for the SpammerRank model, the number of rounds was set to 3 iterations and the value of the smoothing factor α was empirically determined to be 0.3.

The datasets were first preprocessed to obtain and store the necessary information. The review graphs were then generated for the experiments. The macroscopic statistics for the various review graphs generated are enumerated in Table IV. In contrast, the original trust model was applied to over 408 470 reviews by 343 603 reviewers whereas the SpammerRank model was applied on 226 764 reviews and

TABLE IV
COMPARISON OF GRAPHICAL CHARACTERISTICS
ACROSS MODELS AND DATASETS

Observations	Trust Model		SpammerRank Model	
Dataset	Amazon	UCSD	Amazon	UCSD
Number of Nodes	9,215,021	113,727,818	9,215,021	113,727,818
Number of Edges	96,594,190	7,071,355,186	201,138,879	18,242,353,149
Graph Density ($\times 10^{-6}$)	1.1375	0.5467	2.3687	1.4104

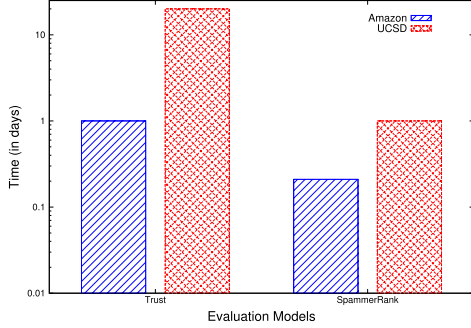


Fig. 4. Comparing runtimes of Trust and SpammerRank algorithms across datasets used.

144 401 reviewers. While the graphical information for the trust model is absent from [3], Liang et al. [10] reported the creation of 13 267 416 edges for the SpammerRank model.

VI. FINDINGS AND DISCUSSION

The models were evaluated on the review graphs generated from the experiments described in Section V. To study the behavior of the models, the metrics obtained were plotted and compared on several grounds of difference. The following sections present the said comparisons and inferences that can be drawn from them.

A. Runtimes

Both models take considerable time for computing the respective spam indicator values. The trust model differs from the SpammerRank model by an order of magnitude in runtime as illustrated by Fig. 4. This excess time is traded off with additional information being generated by the algorithm. The trust model not only produces the trust value for all the reviewers but also generates the respective indicators for the products and the reviews in the system. Since the number of reviews varies greatly for even a small number of reviewers and products, this difference in the scale of runtimes is observed. Thus, for applications requiring only the indicators for reviewers, the SpammerRank model can provide a better scaling time.

B. Value Distributions

In this work, all metrics have been normalized to the range [0, 1]. Figs. 5–7 illustrates the distributions of benignity indicators from the trust model whereas Fig. 8 depicts the distributions of the calculated SpammerRank metric, both across datasets.

The coincidence in the relative positions of the peaks and troughs in the distributions indicates the correlation of the values across the datasets. This observation is brought about by the fact that the datasets represent the growth in the same

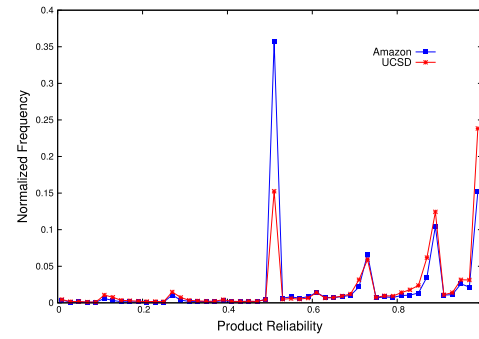


Fig. 5. Distributions of product reliability across datasets.

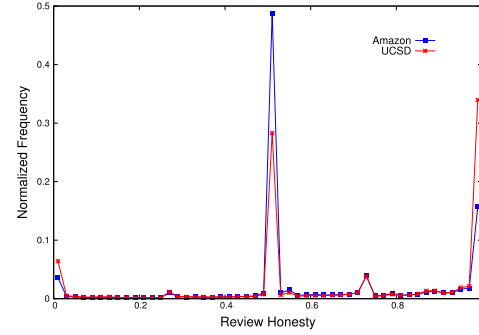


Fig. 6. Distributions of review honesty across datasets.

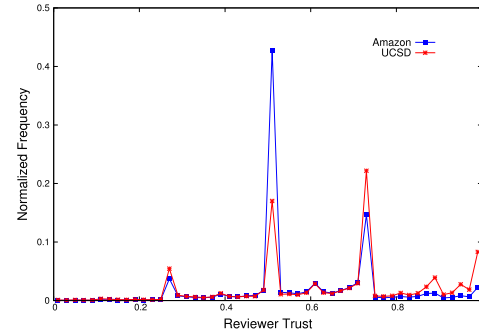


Fig. 7. Distributions of reviewer trust across datasets.

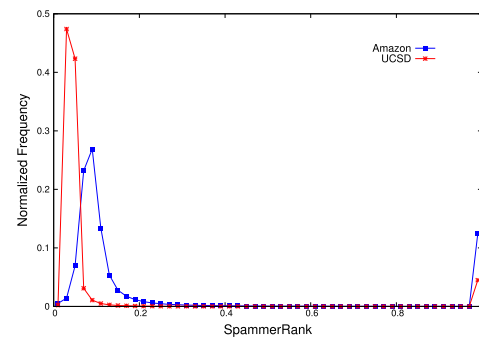


Fig. 8. Distributions of SpammerRank across datasets.

reviewing system as explained in Section IV. An important observation is the “flattening out” of the distributions for the trust model while the SpammerRank values are coalescing. For the trust model, this spreading out of values indicates the increase in the variance of reviews with respect to spam. In other words, where most reviewers tended to be “neutral,” the trend has changed to one where the trustworthiness of the

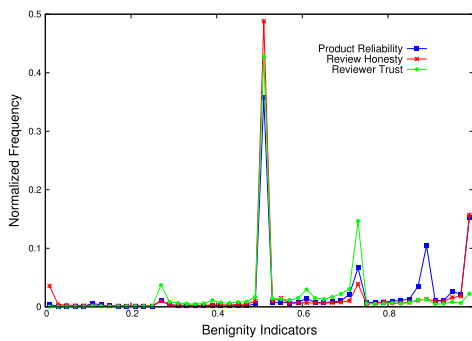


Fig. 9. Distributions of benignity indicators for Amazon Dataset.

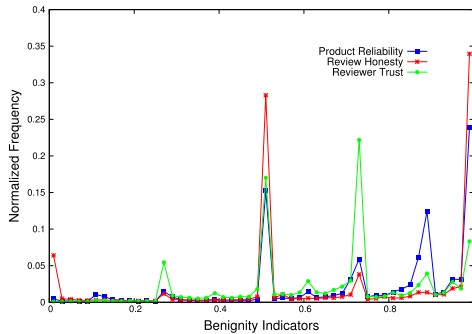


Fig. 10. Distributions of benignity indicators for UCSD Dataset.

TABLE V
CORRELATION COEFFICIENTS FOR DISTRIBUTIONS OF
BENIGNITY INDICATORS FOR EACH DATASET

Correlating Parameters	Amazon	UCSD
Honesty and Reliability	0.9578	0.8812
Reliability and Trust	0.8901	0.6227
Trust and Honesty	0.9228	0.5876

reviewers is increasing as indicated by the rise in peaks near the value of unity. A similar explanation can be given for the rise in peaks for SpammerRank values near the value of 0. This suggests that, over time, the relative number of spam reviewers has decreased.

For the trust model, Figs. 9 and 10 depicts the distributions of the benignity indicators for each dataset. The coincidence in the shape of the curves suggests a high correlation between the benignity indicators.

The correlation values have been tabulated in Table V. For the trust model, the indicators seem to be almost directly correlated. From the values calculated for the UCSD dataset, though, it can be noted that the correlation values start to diminish as the size of the system grows. Nevertheless, the correlation is still higher than anticipated for different benignity indicators, which is a possible disadvantage for the trust model.

C. Values for Top Reviewers

In this section, the trust and SpammerRank metrics have been compared for the top reviewers of the networks created. For this purpose, 20 most active reviewers were identified from both the datasets based on the number of reviews written. For each dataset, the values of the trust and the SpammerRank

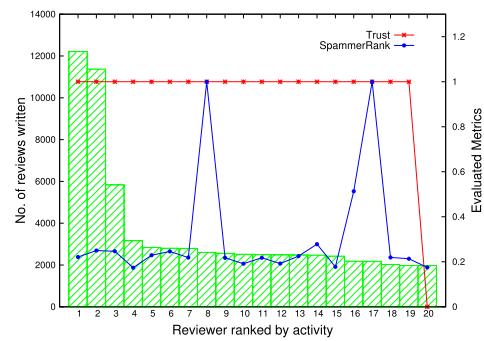


Fig. 11. Trust and SpammerRank values for 20 most active reviewers for Amazon dataset.

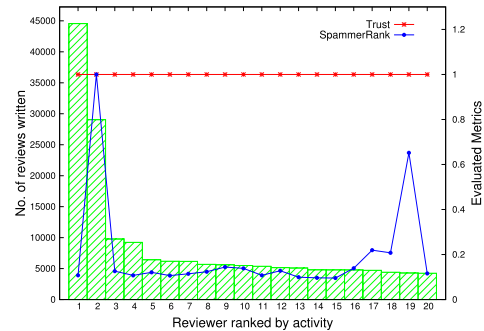


Fig. 12. Trust and SpammerRank values for 20 most active reviewers for UCSD dataset.

TABLE VI
CORRELATION COEFFICIENTS FOR EVALUATED METRICS FOR AND
NUMBER OF REVIEWS BY EACH AUTHOR IN THE DATASETS

Metrics	Amazon	UCSD
Trust	0.1335	0.1543
SpammerRank	-0.0021	0.0150

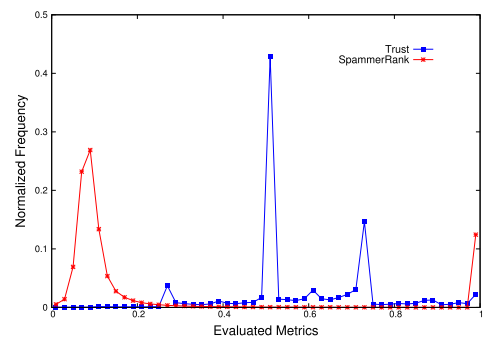


Fig. 13. Comparison of evaluated metrics for Amazon dataset.

metrics have been plotted against the number of reviews by these reviewers in Figs. 11 and 12.

As can be observed, both the metrics disagree largely for these reviewers. For the Amazon dataset, only 1 out of 20 reviewers was set to be completely untrustworthy whereas two reviewers have high SpammerRank values. Trust values are 1 for the rest of the reviewers in top 20 whereas SpammerRank values are relatively uniform. A similar observation can be made for the UCSD dataset where all of the top 20 reviewers were marked as fully trustworthy by the trust model but varied SpammerRank values were observed for all. Although these observations suggest that the trust model

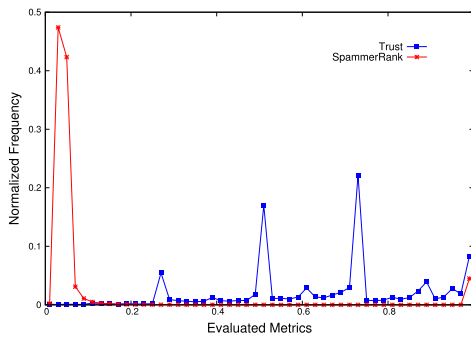


Fig. 14. Comparison of evaluated metrics for UCSD dataset.

overestimates the trust value for those reviewers who have written more number of reviews, the correlation values for the number of reviews and the evaluated metrics, as tabulated in Table VI, do not agree. Due to lack of ground truth, the only viable justification for this discrepancy is introduction of error in estimating the correct metrics by the models as the scale of the network grows.

D. Overall Comparison

The distributions of the trust and the SpammerRank metric for each dataset have been illustrated in Figs. 13 and 14. Given that trust and SpammerRank metrics have opposite meanings—trust value indicates the trustworthiness of the reviewer while SpammerRank value is a measure of being spammer-like for the reviewer—the distributions accurately depict complementary behaviors. The peaks for SpammerRank are observed where the trust value is nearly 0 and vice versa. The models depart from this behavior at unity value. Though inconsistencies and disagreeing values for the models were observed for individual reviewers in Section VI-C, this macroscopic view indicates consistency among the models on the global scale. This is the most important observation for such models when applied to large reviewing systems. Another important observation is the highly irregular distribution of SpammerRank values. Most observed values lie near the value of 0. A possible reason for this observation is the exponential decay of the SpammerRank values for given disagreements. This results in fast decay of the values during computation and makes the SpammerRank metric a poor one when applied to large systems while the trust model appears to be robust in this perspective.

VII. CONCLUSION AND FUTURE WORK

In this article, we empirically evaluated the performance of the trust model of Wang et al. [3] and the SpammerRank model of Liang et al. [10] using the Amazon review dataset [10], [11] and the UCSD dataset [12], [13]. Adaptations and modifications were made to the models wherever necessary to make the process more efficient when applied to large-scale datasets. Computed values were compared and analyzed to determine the underlying effects of large-scale review networks on the metrics. As a result, various departures from expected behaviors were observed along with the strengths and weaknesses of the applied models. Despite of using two

largest available datasets the key scope for improvement lies with the experimentation with recent datasets and the possible exploitation of a few more features.

CONFLICT OF 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 article.

REFERENCES

- [1] F. Gillani, E. Al-Shaer, and B. Assadhan, "Economic metric to improve spam detectors," *J. Netw. Comput. Appl.*, vol. 65, pp. 131–143, Apr. 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S108480451630008X>
- [2] N. Jindal and B. Liu, "Analyzing and detecting review spam," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Oct. 2007, pp. 547–552, doi: [10.1109/ICDM.2007.68](https://doi.org/10.1109/ICDM.2007.68).
- [3] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review graph based online store review spammer detection," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 1242–1247, doi: [10.1109/ICDM.2011.124](https://doi.org/10.1109/ICDM.2011.124).
- [4] S. Noekhah, N. B. Salim, and N. H. Zakaria, "Opinion spam detection: Using multi-iterative graph-based model," *Inf. Process. Manage.*, vol. 57, no. 1, Jan. 2020, Art. no. 102140, doi: [10.1016/j.ipm.2019.102140](https://doi.org/10.1016/j.ipm.2019.102140).
- [5] Y. Zhang, S. Hao, and H. Wang, "Detecting incentivized review groups with co-review graph," *High-Confidence Comput.*, vol. 1, no. 1, Jun. 2021, Art. no. 100006, doi: [10.1016/j.hcc.2021.100006](https://doi.org/10.1016/j.hcc.2021.100006).
- [6] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "NetSpam: A network-based spam detection framework for reviews in online social media," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1585–1595, Jul. 2017, doi: [10.1109/TIFS.2017.2675361](https://doi.org/10.1109/TIFS.2017.2675361).
- [7] Z. Wang, R. Hu, Q. Chen, P. Gao, and X. Xu, "ColluEagle: Collusive review spammer detection using Markov random fields," *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1621–1641, Nov. 2020, doi: [10.1007/s10618-020-00693-w](https://doi.org/10.1007/s10618-020-00693-w).
- [8] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Exp. Syst. Appl.*, vol. 42, no. 7, pp. 3634–3642, May 2015, doi: [10.1016/j.eswa.2014.12.029](https://doi.org/10.1016/j.eswa.2014.12.029).
- [9] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *J. Big Data*, vol. 2, no. 1, pp. 1–24, Dec. 2015, doi: [10.1186/s40537-015-0029-9](https://doi.org/10.1186/s40537-015-0029-9).
- [10] D. Liang, X. Liu, and H. Shen, "Detecting spam reviewers by combing reviewer feature and relationship," in *Proc. Int. Conf. Informative Cybern. Comput. Social Syst. (ICCSS)*, Oct. 2014, pp. 102–107, doi: [10.1109/ICCSS.2014.6961824](https://doi.org/10.1109/ICCSS.2014.6961824).
- [11] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. Int. Conf. Web Search Web Data Mining (WSDM)*, 2008, pp. 219–230, doi: [10.1145/1341531.1341560](https://doi.org/10.1145/1341531.1341560).
- [12] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 43–52, doi: [10.1145/2766462.2767755](https://doi.org/10.1145/2766462.2767755).
- [13] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 785–794, doi: [10.1145/2783258.2783381](https://doi.org/10.1145/2783258.2783381).
- [14] J. Cao, R. Xia, Y. Guo, and Z. Ma, "Collusion-aware detection of review spammers in location based social networks," *World Wide Web*, vol. 22, no. 6, pp. 2921–2951, Nov. 2019, doi: [10.1007/s11280-018-0614-x](https://doi.org/10.1007/s11280-018-0614-x).
- [15] R. Kaur, S. Singh, and H. Kumar, "Rise of spam and compromised accounts in online social networks: A state-of-the-art review of different combating approaches," *J. Netw. Comput. Appl.*, vol. 112, pp. 53–88, Jun. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804518300961>
- [16] Y. Liu, B. Pang, and X. Wang, "Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph," *Neurocomputing*, vol. 366, pp. 276–283, Nov. 2019, doi: [10.1016/j.neucom.2019.08.013](https://doi.org/10.1016/j.neucom.2019.08.013).
- [17] A. Heydari, M. Tavakoli, and N. Salim, "Detection of fake opinions using time series," *Exp. Syst. Appl.*, vol. 58, pp. 83–92, Oct. 2016, doi: [10.1016/j.eswa.2016.03.020](https://doi.org/10.1016/j.eswa.2016.03.020).

- [18] M. Z. Asghar, A. Ullah, S. Ahmad, and A. Khan, "Opinion spam detection framework using hybrid classification scheme," *Soft Comput.*, vol. 24, no. 5, pp. 3475–3498, Mar. 2020, doi: [10.1007/s00500-019-04107-y](#).
- [19] T.-K.-H. Le, Y.-Z. Li, and S.-T. Li, "Do reviewers' words and behaviors help detect fake online reviews and spammers? Evidence from a hierarchical model," *IEEE Access*, vol. 10, pp. 42181–42197, 2022, doi: [10.1109/ACCESS.2022.3167511](#).
- [20] D. Hernández Fusilier, M. Montes-y-Gómez, P. Rosso, and R. Guzmán Cabrera, "Detecting positive and negative deceptive opinions using PU-learning," *Inf. Process. Manage.*, vol. 51, no. 4, pp. 433–443, Jul. 2015, doi: [10.1016/j.ipm.2014.11.001](#).
- [21] Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang, and J. Wu, "HPSD: A hybrid PU-learning-based spammer detection model for product reviews," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1595–1606, Apr. 2020, doi: [10.1109/TCYB.2018.2877161](#).
- [22] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 939–948, doi: [10.1145/1871437.1871557](#).
- [23] Z. Guo et al., "Robust spammer detection using collaborative neural network in Internet-of-Things applications," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9549–9558, Jun. 2021, doi: [10.1109/JIOT.2020.3003802](#).
- [24] J. Yin, Q. Li, S. Liu, Z. Wu, and G. Xu, "Leveraging multi-level dependency of relational sequences for social spammer detection," *Neurocomputing*, vol. 428, pp. 130–141, Mar. 2021, doi: [10.1016/j.neucom.2020.10.070](#).
- [25] Y. Liu and B. Pang, "A unified framework for detecting author spamicity by modeling review deviation," *Exp. Syst. Appl.*, vol. 112, pp. 148–155, Dec. 2018, doi: [10.1016/j.eswa.2018.06.028](#).
- [26] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal, "Detecting group review spam," in *Proc. 20th Int. Conf. Companion World Wide Web*, Mar. 2011, pp. 93–94, doi: [10.1145/1963192.1963240](#).
- [27] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proc. 21st Int. Conf. World Wide Web*, Apr. 2012, pp. 191–200, doi: [10.1145/2187836.2187863](#).
- [28] Z. Wang, S. Gu, X. Zhao, and X. Xu, "Graph-based review spammer group detection," *Knowl. Inf. Syst.*, vol. 55, no. 3, pp. 571–597, 2018, doi: [10.1007/s10115-017-1068-7](#).
- [29] Z. Wang, S. Gu, and X. Xu, "GSLDA: LDA-based group spamming detection in product reviews," *Appl. Intell.*, vol. 48, no. 9, pp. 3094–3107, 2018, doi: [10.1007/s10489-018-1142-1](#).
- [30] L.-C. Cheng, H.-W. Hu, and C.-C. Wu, "Spammer group detection using machine learning technology for observation of new spammer behavioral features," *J. Global Inf. Manage.*, vol. 29, no. 2, pp. 61–76, Mar. 2021, doi: [10.4018/JGIM.2021030104](#).
- [31] F. Zhang, X. Hao, J. Chao, and S. Yuan, "Label propagation-based approach for detecting review spammer groups on e-commerce websites," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105520, doi: [10.1016/j.knsys.2020.105520](#).
- [32] H. Byun, S. Jeong, and C.-K. Kim, "SC-Com: Spotting collusive community in opinion spam detection," *Inf. Process. Manage.*, vol. 58, no. 4, 2021, Art. no. 102593, doi: [10.1016/j.ipm.2021.102593](#).
- [33] S.-J. Ji et al., "A burst-based unsupervised method for detecting review spammer groups," *Inf. Sci.*, vol. 536, pp. 454–469, Oct. 2020, doi: [10.1016/j.ins.2020.05.084](#).
- [34] P. Rathore, J. Soni, N. Prabakar, M. Palaniswami, and P. Santi, "Identifying groups of fake reviewers using a semisupervised approach," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1369–1378, Dec. 2021, doi: [10.1109/TCSS.2021.3085406](#).
- [35] D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, "Detection of opinion spam based on anomalous rating deviation," *Exp. Syst. Appl.*, vol. 42, no. 22, pp. 8650–8657, Dec. 2015, doi: [10.1016/j.eswa.2015.07.019](#).
- [36] C. M. Aye and K. M. Oo, "Review spammer detection by using behaviors based scoring methods," in *Proc. Int. Conf. Adv. Eng. Technol.*, 2014, pp. 350–355, doi: [10.15242/iee.e0314158](#).
- [37] N. Jindal, B. Liu, and E.-P. Lim, "Finding unusual review patterns using unexpected rules," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 1549–1552, doi: [10.1145/1871437.1871669](#).
- [38] H. Xue, F. Li, H. Seo, and R. Pluretti, "Trust-aware review spam detection," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 726–733, doi: [10.1109/Trustcom.2015.440](#).
- [39] K. Sharma and K.-I. Lin, "Review spam detector with rating consistency check," in *Proc. 51st ACM Southeast Conf.*, Apr. 2013, p. 34, doi: [10.1145/2498328.2500083](#).
- [40] Y. Mao and H. Shen, "Web of credit: Adaptive personalized trust network inference from online rating data," *IEEE Trans. Computat. Social Syst.*, vol. 3, no. 4, pp. 176–189, Dec. 2016, doi: [10.1109/TCSS.2016.2639016](#).
- [41] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 823–831, doi: [10.1145/2339530.2339662](#).
- [42] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting burstiness in reviews for review spammer detection," in *Proc. ICWSM*, vol. 13, 2013, pp. 175–184.
- [43] R. K. Dewang and A. K. Singh, "State-of-art approaches for review spammer detection: A survey," *J. Intell. Inf. Syst.*, vol. 50, no. 2, pp. 231–264, Apr. 2018, doi: [10.1007/s10844-017-0454-7](#).
- [44] B. Manaskasemsak, J. Tantisuwankul, and A. Rungsawang, "Fake review and reviewer detection through behavioral graph partitioning integrating deep neural network," *Neural Comput. Appl.*, vol. 35, pp. 1–14, Apr. 2021, doi: [10.1007/s00521-021-05948-1](#).
- [45] Z. Wang, T. Hou, D. Song, Z. Li, and T. Kong, "Detecting review spammer groups via bipartite graph projection," *Comput. J.*, vol. 59, no. 6, pp. 861–874, 2016, doi: [10.1093/comjnl/bxv068](#).
- [46] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "Spatio-temporal graph representation learning for fraudster group detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 25, 2022, doi: [10.1109/TNNLS.2022.3212001](#).
- [47] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Identify online store review spammers via social review graph," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, p. 61, 2012, doi: [10.1145/2337542.2337546](#).
- [48] S. K. Fayazbakhsh and J. Sinha, "Review spam detection: A network-based approach," Dept. Comput. Sci., Stony Brook Univ., Stony Brook, NY, USA, Final Project Rep., CSE 590, Nov. 2012.
- [49] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. ICWSM*, vol. 13, 2013, pp. 2–11.
- [50] Y. Lu, L. Zhang, Y. Xiao, and Y. Li, "Simultaneously detecting fake reviews and review spammers using factor graph model," in *Proc. 5th Annu. ACM Web Sci. Conf.*, May 2013, pp. 225–233, doi: [10.1145/2464464.2464470](#).
- [51] S. Noekha, E. Fouladfar, N. Salim, S. H. Ghorashi, and A. A. Hozhabri, "A novel approach for opinion spam detection in e-commerce," in *Proc. 8th Int. Conf. E-Commerce Focus E-Trust*, 2014, pp. 1–8.
- [52] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, 1999.
- [53] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Portland, OR, USA, 2011, pp. 309–319.