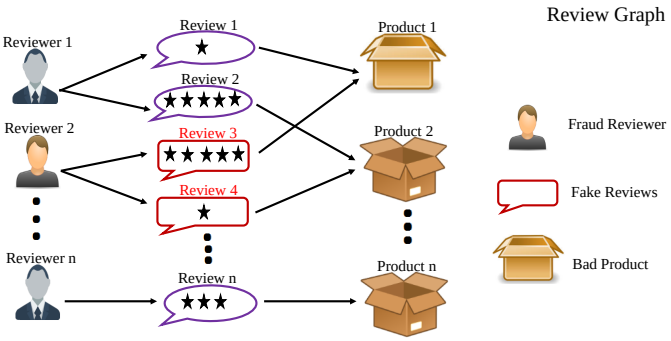
Detecting Product Review Spammers using Principles of Big Data

***Abstract*—Growing consumerism has led to the importance of online reviews on the Internet. Opinions voiced by these reviews are taken into consideration by many consumers for making financial decisions online. This has led to the development of opinion spamming for profitable motives or otherwise. Work has been done to tackle the challenge of identifying such spammers, but the scale of the real-world review systems demands this problem to be tackled as a Big Data challenge. So, an effort has been made to detect online review spammers using the principle of Big Data. In this work, a rating-based model has been studied under the light of large-scale datasets (more than 80 million reviews by 20 million reviewers) using the Hadoop and Spark frameworks. Scale effects have been identified and mitigated to provide better context to large review systems. An improved computational framework has been presented to compute the overall spamcity of reviewers using exponential smoothing. The value of the smoothing factor was set empirically. Finally, future directions have been discussed.**

***Index Terms*—Spam Reviews, Review Spammer Detection, Big Data, E-commerce.**

I. Introduction

User reviews are modes of voicing opinions regarding the qualities of a certain product or service by online entities. Online reviewing systems have grown to be one of the cornerstones of e-commerce websites. Growing popularity has even spawned numerous websites dedicated towards reviewing products, places, attractions, and so on, such as Yelp, Zomato, Dianping, etc. As online consumerism expands across geographical expanses, increasingly more consumers rely on such reviews for deciding upon buying things online. User reviews can be materialized into many forms. Product reviewing is a common practice on video streaming websites such as YouTube. Despite this sophistication, textual reviews remain the dominant mode of reviewing.



Issues	1: How to identify the reviews from the review graphs? 2: How to deal with large scale datasets?
Purpose	1: Using Rating Model to study large scale datasets with Hadoop and Spark. 2: Identification and Mitigation of scaling effects.

Figure 1: Overview of the proposed work

Though online reviews are quite popular and helpful, due to ease in posting of reviews and lack of moderation and filtering, most online reviewing systems face the challenge of *opinion spamming*. Review spammers may post reviews that do not voice legitimate opinions. This may be done to have personal gains or for professional reasons. Review spamming is usually aimed at derailing the prevalent opinion about a product or service. It may be used to unhealthily promote a subject or to demote a rival subject. Since the popularization of Web 2.0 as well as crowd-sourcing [1], review spam has been viewed as a serious threat to businesses and online marketing firms as they distort the consumer experience and harm the reputations of the respective firms. This has prompted researchers to actively tackle the problem of review spam detection with many industrial supporters as well [2].

Review spam analysis was introduced by Jindal and Liu [3] who identified different types of review spam. Since then, a multitude of approaches has been taken in literature to detect deceptive reviews. Textual analysis and investigative methods using associated data have been exploited to detect opinion spam. Apart from detecting such reviews, identifying spamming individuals and their possible associations have also been pursued. Machine learning and graph-based approaches have been explored too [4]–[6]. A major problem in this field is the lack of a standard dataset. Since collecting ground truth about online reviews is difficult for humans [7], studies have exploited synthetic review datasets. But most of these datasets are relatively smaller in size when compared to real-world reviewing systems which contain hundreds of millions of reviews. This has inspired the development of Big Data frameworks for study and deployment purposes. While there have been developments of such systems for applications to social systems in general [8]–[10], the principles of Big Data have not been directly employed to model reviewing systems and their processing. The theme of the proposed work based on principles of Big Data is shown in Figure 1.

This work presents a study of metadata based modeling on large-scale reviewing systems using Big Data. Figure 2 briefly describes the recent trend as well as contribution to the work. Major contributions of this work are listed as follows:

1) Analysis of simple metadata based modeling [11] over a large reviewing system.
2) Identification of scaling effects on Rating Models and its mitigation by applying exponential smoothing.
3) Proposed a computational framework to compute the overall spamicity of reviewers.
4) An effort has been made to use Big Data as an important tool to study real-world reviewing systems.
5) An attempt to improve a metadata based model by extending the existing work by Savage et al.(2015) [11]

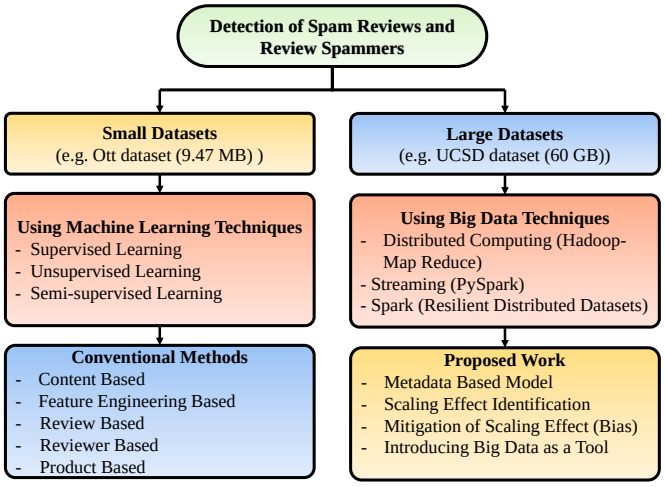


Figure 2: Recent trend for detection of review spammers

in the context of a large review corpus.

6) An inclusive study of literature has been done by emphasizing more on detection of review spammers (individual as well as group).

The rest of this paper has been organized as follows. In Section II, different approaches taken by previous research works have been enumerated. The model formulation, as well as the proposed Rating Model has been presented in Section III. Section IV emphasizes the Big Data infrastructure used for conducting the experiments and an in-depth discussion on the observations made. The paper has been concluded, and future scopes have been indicated in Section V.

II. Related Work

Spam detection has been traditionally treated for e-mail and websites [3], [12]–[14]. Drawing inspiration from these works, Jindal and Liu [15] the problem and importance of detecting online review spam formally. This was succeeded by Ott et al. [7], [16], [17] who presented a synthetic 'gold-standard' dataset of hotel reviews generated by anonymous online workers. Since then, various models have been developed to detect opinion spam. Broadly, this problem has been viewed from three different perspectives, (i) detecting spam reviews [18], (ii) detecting review spammers, and (iii) detecting collaborations of review spammers [4], [5].

Based on the information available for a given reviewing system, most of these models can be distinguished as *text-based* or *metadata-based* models. Machine learning forms a major class of approaches applied in text-based models using textual features [3], [19]–[26], n-gram features [3], [7], [27], linguistic features [28]–[31] and sentiment [32] as a feature. Figure 3 briefly describes existing research on review spam. Traditionally, supervised learning is applied to detect spam reviews. Owing to the lack of sufficiently large and annotated datasets representing ground truth, Semi-supervised learning methods [33]–[35] have been proposed to detect spam reviews [36]–[40]. In addition to these, unsupervised methods have also been developed that treat deceptive reviews as outliers in a composite feature space [19].

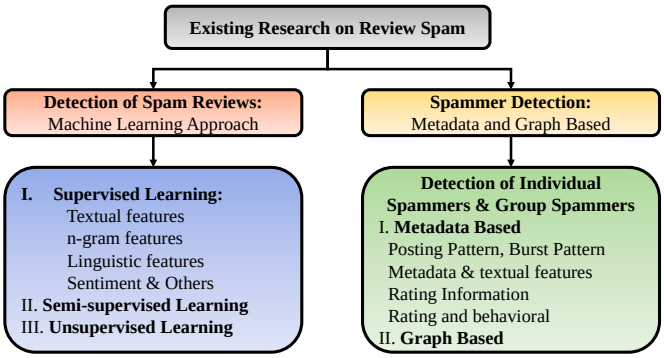


Figure 3: Overview of Related Work on Spam Reviews

Metadata-based models employ the associated information for the review such as reviewer identity, posting time, the pattern of posting [41], [42], etc. to detect review spam. This class of models also constitutes a majority of approaches that detect spam reviewers and their associations [5]. A popular metadata-based detection approach is to use the rating information associated with reviews for spammer detection. Various works [11], [43]–[45] have been proposed that exploit deviations and anomalies in rating patterns of reviewers across a system to segregate spam reviewers from benign reviewers. Apart from the timeliness of reviews, underlying relationships between reviews, reviewers, and products have also been mined to detect spam reviewers. These relationships have been formalized as *review graphs*, which are then processed to detect spam entities [46]–[53]. Akoglu et al. [54] combined the graph-based approach with rating modelling, called 'FraudEagle', to present benchmark results in spam reviewer detection. This model was then succeeded by Savage et al. [11] who proposed a statistical model of reviewers and ratings and claimed better performance than FraudEagle. These models are based on the assumption that ratings are visible indicators of a user's opinions, and thus, the general consensus for a product depends on the ratings it has. Spammers try to manipulate the ratings in order to distort their opinions, which can be detected by studying the rating patterns. Apart from these, there exist specific models [55]–[60] as well which combine rating information with other behavioral footprints to detect spam reviewers [61]–[65]. Recently, many works have been reported in literature for detection of individual [66]–[68] as well as spammer groups using unsupervised learning [69]–[77], semi-supervised learning [56], [59], supervised learning [78]–[82] as well as hybrid learning [83]. Escalante et al. [84] have emphasized on early detection of deception using profile based representations. But the drawback of profile based systems is validation and authentication of profiles as well as privacy concerns of users. Sharing of information across multiple sites regarding users may improve the detection process of spammers which in the current scenario seems to be challenging because of privacy and legal issues. A similar kind of work [85] has also been reported in literature that considers rating deviation, content based factors, and activeness of reviewers for spamicity detection. Even though the work is an effective one, it needs consideration of several

other review centric as well reviewer centric features for further improvement. The effectiveness of the system to deal with scalability is still doubtful. After a detailed literature work, it has been observed that only a few works have been reported in the literature that deals with a huge amount of data using Big Data or Deep Learning [86], [87].

Though the discussion in related work makes it clear that a plausible volume of research has been done to detect online review spammers, a vital problem that persists is the lack of a true standard dataset. Despite the datasets provided by Ott et al. [7], [16] are considered to be the 'gold-standard' datasets, they are partially synthetic in nature and have been analysed by researchers for their effective utility [88]. This has impeded the development of an important aspect of the problem— its scale. Yelp, a commercial reviewing website for local attractions and facilities in the US, claims the existence of more than 121 million reviews on their site by the end of 2016 [89]. Since Yelp filters the reviews it receives, the actual estimate of the number of reviews submitted by reviewers would be much larger. The same is the case for e-commerce websites such as Amazon [90]. Given these facts, it is clear that a sustainable spam detection system must be scalable. This has inspired this work to study reviewing systems in the context of Big Data. Though generalized precursory work has been done to detect deceptive opinions in social systems [8], the proposed work is a direct attempt to apply simple detection models to large scale reviewing systems. Similarly, Hussain et al. [82] have tried different spammer's behavioral features to calculate the review spam score to identify spammers and spam reviews. Even though experimental evaluations were conducted on a real-world Amazon review dataset with 26.7 million reviews and 15.4 million reviewers, the authors have not clearly mentioned how they have processed such a huge amount of data. Even there is no mention regarding how the detection of spammers was done.

III. Proposed Rating Model

The models for tackling challenges in Big Data are usually desirably simple and relatively computationally 'cheaper' in nature. Keeping in mind this reasoning, a simple rating based spam reviewer detection model [11] is chosen for evaluation and improvisation. It is based on the deviations of the ratings associated with the reviews to classify a given reviewer as a spammer or otherwise. Since this model uses only the rating value for classification purposes, it is a metadata-based model. This model has been shown to outperform benchmark graph-based systems [54]. The major reason for using this model in this work is its versatility and simplicity, making it an ideal model to be used as complement to content based models [5].

A. Notations Used

In this model, various symbols have been used to describe the reviewing system and its components. A review is represented by v, a product by p, and a reviewer by r for the rating model. The derived notations for the rest of the expressions used in this work have been enumerated in Table I.

Table I: Symbols used for the Rating Model

Symbol	Meaning
Δ	Threshold difference between honesty values
ϕ	Probability of posting disagreeing review
N	Total number of reviews in the system
N_d	Total number of disagreeing reviews in the system
n_r	Total number of reviews written by r
$\sigma_{r,p}$	Rating given by r to p for a given review
$\overline{\sigma}_{p,i}$	Weighted mean rating for p as determined for the ith iteration
$k_{r,i}$	Number of disagreeing reviews written by r as determined for the ith iteration
$u_{r,i}$	Honesty of r for the ith iteration
$\psi(r)$	Probability of a random reviewer to act like r purely by chance
$s(r)$	Spamcity value for r
$\Gamma(r)$	The set of reviews written by r
$\tau(p)$	The set of reviewers who reviewed p

B. The Base Rating Model

In this work, the model proposed by Savage et al. [11] has been used and referred to as the *Base Rating Model*. The model describes the formulation and calculation of the spammer-ness measure of each reviewer called the *Spamcity* value. This value is calculated using the deviations of the ratings given by a particular reviewer with respect to the prevalent opinions for the respective products.

Savage et al. [11] argue that since most reviewing systems implement rating systems for their reviews and that most of these ratings are visible to the public, the general consensus about the product/service is indicated by these ratings. They claim that using these ratings for spammer detection is better than using review text because many reviewing systems do not have review texts at all but implement a rating system in one form or the other. The model is developed further using the following two axioms:

(1) The majority of the reviews in the given system are composed of honest reviewers. This must be true for any credible review system as the negation means that the review system in question is completely broken.
(2) The mean rating represents the prevalent opinion for the product as honest reviewers tend to converge in their perception of the product/service quality via the ratings. This is demonstrated in [11].

Based on these premises, the authors model review spammers as entities who try to 'sway' the mean rating towards the extreme ends of the rating scale for a given product. This corresponds to the promotion or demotion of the product. Therefore, spam reviewers attempt to shift the prevalent opinion for a product by posting ratings which *disagree* with the mean rating, and thus can be detected by measuring the degree of such attempts. That is, if a reviewer disagrees with the prevalent opinion about the products (s)he reviews 'too frequently', (s)he has a higher probability of being a spammer. Here, *disagreement* refers to posting of reviews that lie on the other half of the rating spectrum as that of the mean rating.

For example, on a rating scale of 1 to 5, the rating spectrum is halved at the median rating value of 3. A review with a rating of 2 is said to disagree with the mean rating of 4 for a product as these values lie in different halves of the rating scale about the median. These rating conventions have been assumed in this work.

To decide if (s)he disagrees with the mean ratings disproportionately, the authors estimate the probability of a random honest reviewer to have posted as many disagreeing reviews as the given reviewer using a *binomial distribution*. This probability derives the *Spamcity* value for the reviewer. The formulations and mathematical expressions for the calculations in the model have been presented in Section III-C.

The binomial hypothesis testing function represents the binomial distribution represented by $P()$. For a given number of trials n out of which k trials are favorable, for the random variable X and the probability of success as p, the one-tailed binomial testing function is described by Equation 1.

$$P(X \geq k; n, p) = \ 1 - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i} \tag{1}$$

C. Model Formulation

In this Section, the formal treatment of the rating model is presented. As described in Section III-B, for a given reviewer, the probability of a random honest reviewer to have disagreed as many times is computed. For a random reviewer, the probability of writing a disagreeing review for any given product is described by Equation 2.

$$\phi = \frac{N_d}{N} \tag{2}$$

In binomial distributions, any experiment has two outcomes— *success* or *failure*. In this context, the posting of a disagreeing review by a random reviewer is considered as a successful outcome. Thus, for a given reviewer r who has posted n_r reviews, of which k_r reviews disagree with the mean ratings of their respective products, the probability that a *random honest reviewer* would post the same number of disagreeing reviews is given by using the binomial testing function as in Equation 3.

$$\psi(r) = P(X \geq k_r; n_r, \phi) \tag{3}$$

This probability indicates the likelihood of r posting k_r or more disagreeing reviews out of n_r reviews by random chance. Thus, the *spamcity* for r can be derived as the complement of this value with respect to unity, defined in Equation 4.

$$s(r) = 1 - \psi(r) \tag{4}$$

Since the calculation of the value of $\psi(r)$ depends on the value of k_r, which in turn depends on the mean ratings for the products, the value will be inaccurate as the calculation of these mean values also takes into account those reviews which are posted by the spammers. To tackle this challenge of properly estimating the prevalent opinion, an iterative method has been described by the authors of the model. In this method, the prevalent consensus for any product p is calculated as a weighted mean of the ratings of the reviews for p. The weights for calculating this mean are derived as the honesty values of the reviewers who have reviewed p. For an iteration i, the weighted mean rating is calculated using Equation 5.

$$\overline{\sigma}_{p,i} = \sum_{r \in \tau(p)} \frac{\sigma_{r,p} \times u_{r,i}}{|\tau(p)|} \tag{5}$$

Based on the weighted mean ratings, new honesty values are calculated for all reviewers. First, the new count of all disagreeing reviews for the current iteration is computed. Using this, the honesty for a reviewer is defined as the probability of the reviewer to post agreeing reviews on any product as is defined by Equations 6 and 7.

$$k_{r,i} = count(\forall \sigma_{r,p} \in \Gamma(r) : (\sigma_{r,p} < 3 \wedge \overline{\sigma}_{p,i} \geq 3) \vee (\sigma_{r,p} \geq 3 \wedge \overline{\sigma}_{p,i} < 3)) \tag{6}$$

$$u_{r,i} = 1 - \frac{k_{r,i}}{n_r} \tag{7}$$

A check is then made to see if the honesty values have started converging. This is done by setting a threshold value for the difference denoted by Δ. If the difference between the old and new honesty values for all authors is less than the threshold difference, the iterations stop. Finally, the spamcity values for all reviewers are then calculated using Equation 4. Note that it is assumed that all reviewers are honest in the beginning, that is, $u_{r,0} = 1$ for all r. The process is formally depicted using Algorithm 1. Further elaboration can be found in [11].

The complexity of the algorithm has been determined to be $\mathcal{O}(maxIterations \times (n_{reviews} + n_{reviewers}) + n_{reviewers})$ [11] where $maxIterations$ is the final value of $roundCounter$ in Algorithm 1.

D. Drawbacks of the Base Rating Model

Though the model used is demonstrated to be effective for reviewing systems [11], it suffers from a major drawback when applied to large scale reviewing systems.

Consider Equation 2 to calculate the probability of a random reviewer to post a disagreeing review. This probability is considered to be fixed in Algorithm 1 (it is only being used in the last iteration). Though the weighted mean ratings for the products are computed in each iteration to smooth the honesty values of the reviewers, these values should also be taken into account to update the number of total disagreeing reviews in the system. Unfortunately, this is not being done in Algorithm 1, and this information is being lost. The preservation of this information has been pointed out by the authors themselves [11]. As a modification, the proposed work suggests calculating the global disagreement probability, ϕ, in each iteration as ϕ_i.

The second significant drawback also relates to the calculation of ϕ. Considering Equation 2 again, it can be noted that for large reviewing systems, N will be a large number. For

Algorithm 1 Computational Framework to compute spamcity of reviewers

INPUT: The set of Reviewers, $\mathcal{R}$, the set of reviews $\mathcal{V}$, the set of products $\mathcal{P}$, Δ, $rounds$
OUTPUT: Spamcity values, s, for all reviewers

```
Set u_{r,0} = 1.0 ∀r ∈ R
roundCounter = 1;
while roundCounter ≤ rounds do
    for each p ∈ P do
        Compute σ̄_{p,i} using Equation 5;
    end for
    for each r ∈ R do
        Compute k_{r,i} using Equation 6;
        Compute u_{r,i} using Equation 7;
    end for
    if |u_{r,i-1} − u_{r,i}| < Δ ∀r ∈ R then
        break;
    end if
    roundCounter + +;
end while
for each r ∈ R do
    Compute k_{r,roundCounter} using Equation 6;
    Compute ψ(r) using Equation 3;
    Compute s(r) using Equation 4;
end for
```

growing values of N, it is clear that for a robust and utilitarian reviewing system, the rate of growth of N_d will be much less than that of N, that is:

$$\frac{dN_d}{dN} \ll 1 \tag{8}$$

Therefore, ϕ as defined in Equation 2 will decrease inherently for large reviewing systems.

$$\lim_{N \to +\infty} \phi = 0 \tag{9}$$

As the probability of posting disagreeing values decreases, $\psi(r)$ will start decreasing, and so, $s(r)$ will increase. This indicates that the model inherently inflates the spamcity of the reviewers when the number of reviews in the system grows in a healthy fashion. This implies that for large reviewing systems, this model will judge more number of reviewers as spammers because of this bias of the number of reviews.

E. Improved Computational Framework

In order to tackle these drawbacks in a manner that does not incur large overheads, a collective effort is necessary. An *exponential smoothing* process has been proposed in this work as the required effort. This approach takes into account the history of the spamcity values as well as currently computed value to generate the overall spamcity value, $S(r)$. The spamcity values are computed for each iteration instead of only at the last, and are treated as data points spaced across the values of ϕ. In this way, all the values of ϕ are accounted for throughout the iterations. The smoothing factor, denoted by α, is defined as the weight for the exponential averaging process. Since this factor exponentially averages over the different spamcity values, it can be treated as a *control parameter* to oppose the taxing of spamcity values. The smoothing process can be depicted by Equation 10.

$$S(r,i) = \alpha \times s(r,i) + (1-\alpha) \times S(r,i-1) \tag{10}$$

where $s(r,i)$ and $S(r,i)$ are the original and overall spamcity values of reviewer r for iteration i, respectively.

Again, note that the initial values of the overall spamcity for each reviewer will be set to 0 as it is assumed initially that all reviewers are honest. The modified framework has been presented in Algorithm 2. The runtime for the algorithm changes to $\mathcal{O}(maxIterations \times (n_{reviews} + n_{reviewers}))$.

Algorithm 2 Modified Computational Framework to compute overall spamcity of reviewers

INPUT: The set of Reviewers, $\mathcal{R}$, the set of reviews $\mathcal{V}$, the set of products $\mathcal{S}$, Δ, $rounds$, α
OUTPUT: Overall Spamcity values, S, for all reviewers

```
Set u_{r,0} = 1.0 ∀r ∈ R
Set S_r = 0.0 ∀r ∈ R
roundCounter = 1;
while roundCounter ≤ rounds do
    for each p ∈ P do
        Compute σ̄_{p,i} using Equation 5;
    end for
    Compute φ_i using Equation 2;
    for each r ∈ R do
        Compute k_{r,i} using Equation 6;
        Compute u_{r,i} using Equation 7;
        Compute ψ(r) using Equation 3;
        Compute s(r,i) using Equation 4;
        Update S(r) using Equation 10;
    end for
    if |u_{r,i-1} − u_{r,i}| < Δ ∀r ∈ R then
        break;
    end if
    roundCounter + +;
end while
```

IV. Experiments and Results

The rating model was implemented and applied over the Amazon and the UCSD datasets. Since the datasets used are large in size, the *Hadoop* framework was used to store and process effectively. In Section IV-A, the various job structures for the distributed computation of the spamcity values have been described. The Datasets were described in Section IV-B. The results and their discussions have been elucidated in Section IV-C.

A. Experimental Setup

For the computations in the model, a multinode Hadoop distributed computing cluster was set up. This cluster was

heterogeneous in nature and consisted of 3 Linux based and 1 MacOS based systems. The Linux based systems ran over a 12-core Intel Xeon E2620 processor, while the macOS based system ran on Intel i5 5th Generation processor. The total memory for the cluster was allocated to be 64 GB with configured 4 TB of storage space. The redundancy factor was set at 3 for all data points in the system. In addition to the Hadoop framework, the Spark framework was also exploited on the said cluster. All the MapReduce jobs were carried out using Python streaming, while for Spark, the PySpark shell was used. For implementing the model described in Section III, the following three major tasks were identified and carried out.

a) Pre-processing Phase: In this phase, the datasets were processed to extract the various dictionaries that would be needed for further processing. To dump these dictionaries to their respective HDFS files, the *join* concept was used. MapReduce on Hadoop Yarn was used to perform this task using Python3 streaming.

b) Processing Phase: In this phase, the main iterative framework was executed. In order to do this, MapReduce proved to be inefficient. This is because Hadoop jobs are *Shared Nothing* jobs, i.e., they do not share any data in between them. Since all the jobs needed to have the dictionaries for processing, loading them all to each job was found to be inefficient. For this reason, the process was shifted to the Spark framework where the *Resilient Distributed Datasets* were used to process the files in the PySpark shell. A single program was written and iterated over the shell to achieve the computation. The threshold difference value was set to be $\Delta = 10^{-4}$ for all the computations. At this threshold, interestingly, all the experiments were completed by the 5th round.

c) Post-processing Phase: In this phase, the final computation of the values took place. This was a simple MapReduce task to dump the results and the histograms for the data. Data was then processed to dump the required histogram values as CSV files.

B. Datasets Used

The model described in Section III has been implemented and studied under the light of large scale datasets using Big Data processing systems. The datasets chosen represents one of the largest publicly available data collection in this field of research.

Two datasets have been used in this work. The *Amazon dataset* [3] contains over 5.8 million reviews written by more than 2 million reviewers. The reviews were collected by crawling the Amazon website [90] from a variety of genres and categories. Each review is a tab separated tuple in the dataset, which consists of review information such as reviewer and product identifiers, rating values, feedback information, review title and text, etc. This dataset has also been used by Savage et al. [11].

The second dataset used was collected by McAuley et al. [91], [92] and has been called as the *UCSD dataset* in the proposed work. This dataset contains more than 140 million reviews, also collected from Amazon. Due to the merging of similar products by Amazon, this dataset consists of numerous duplicates. So, a *de-duplicated* version of the dataset has been used in this work. This data collection contains more than 82 million reviews for 10 million products and is the largest available dataset for research in this field. This dataset also contains review metadata in addition to the review text. Each review is present as a JSON dictionary in the dataset. The Amazon dataset is 6.31 GB in size, while the UCSD dataset is about 60 GB in size. Both contain rating information on a scale of 1 to 5.

Both the datasets are based on the Amazon store and have been collected at different points of time. The Amazon dataset pre-dates the UCSD dataset. For this reason, studying the rating model over these datasets can provide better insights not only for how the model behaves with the scale but also for the evolution of the given reviewing system across time.

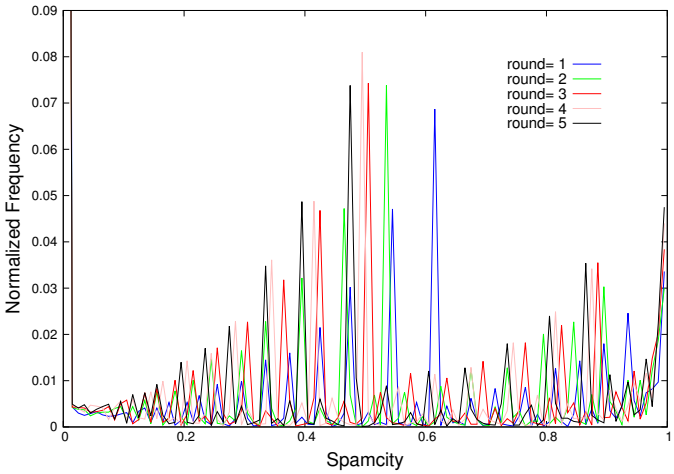
C. Analysis and Inferences

The results were collected from the cluster in the form of histogram information and were processed on a single machine. Histogram values have been plotted into graphs and compared to visualize the model's behavior on a global scale. Observations made and inferences drawn from these graphs have been presented in the sections that follow.

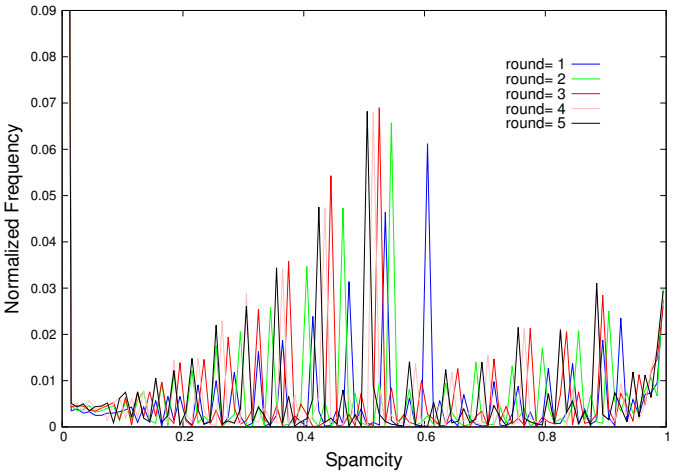
1) Behaviour of the Original Model: The original rating model, described by Algorithm 1, dictates the calculation of the Spamcity values of the reviewers after the final iteration takes place. In order to illustrate the intermediate states of processing of the model, Spamcity values were also calculated at the end of each iteration instead of only at the final step. The distributions for these intermediate Spamcity values were analysed to comment on the nature of the original rating model [11].

Figure 4(a) depicts the distributions of the intermediate Spamcity values for the Amazon dataset. As can be observed, the values *decrease* as the number of rounds increase. This is a consequence of the gradual correction of the mean rating values. Since the honesty values for all reviewers are initially set to 1, the mean rating computed for any given product is at its maximum value. This value gradually decreases as the honesty values for all reviewers drop and start converging to their respective 'real' values. This brings in better approximations of the agreement of the reviewers as most of them are assumed honest (see Axiom (1) in Section III-B). Therefore, the Spamcity values shift to the left end of the spectrum on a macroscopic scale. A similar trend is observed in the distributions for the UCSD model illustrated in Figure 4(b).

An important observation that can be made about the model is the way *agreement* is viewed. As Table II illustrates, reviewers usually provide disproportionately high ratings to all products. This causes the mean ratings to have higher values too, and thus, agreements are established more on the higher half of the rating spectrum than the lower half. In other words, a reviewer is more likely to have agreed with the mean rating falling above the median rating of 3 than below it.



(a) Distributions of intermediate Spamcity values for Amazon dataset



(b) Distributions of intermediate Spamcity values for UCSD dataset

Figure 4: Distributions of intermediate Spamcity values

Table II: Frequencies (in number of reviews) observed for various ratings

Rating Given	Amazon	UCSD
1.0-1.5	482,826	6,712,115
2.0-2.5	316,958	4,265,229
3.0-3.5	507,462	7,049,296
4.0-4.5	1,170,374	15,480,804
5.0	3,360,429	49,169,647

2) Existence of Bias in Spamcity: In Section III-D, the existence of an inherent bias in the calculation of the Spamcity values with the scale of the review system has been discussed and demonstrated formally. To validate the arguments in this work, the distributions of the Spamcity values calculated as per the original model [11] have been compared. The graphs for the distributions have been illustrated in Figure 5. Note that these values were calculated at the end of the iterations as per Algorithm 1. It can be observed that there has been a consistent shift of Spamcity values towards unity across the datasets. The similarity in the relative positioning of the peaks is due to the fact that both the datasets depict the same reviewing system across time (See Section IV-B). This shift of values means that the Spamcity values have increased in a regular fashion for all reviewers to the extent that it can be visually discerned. If there were no bias, this increase would only be observed by accepting the argument that all reviewers have somehow managed to become more 'spammer-like' in a consistent manner. Since a global collaboration at such scale is not possible, this cannot be the ground truth, and thus, this demonstrates the existence of bias in Spamcity calculation as described in Section III-D.

3) Behavior of the Modified Model: In order to mitigate the bias in calculating Spamcity values, an exponential smoothing process was introduced in Section III-D. The modified model, represented by Algorithm 2, was implemented. The goal of the modifications was to oppose the increase in Spamcity values because of scale. Therefore, it can be said that the aim of the modifications was to make the Spamcity distribution more uniform, but at the same time maintain consistency by preserving the relative structure of the distribution.

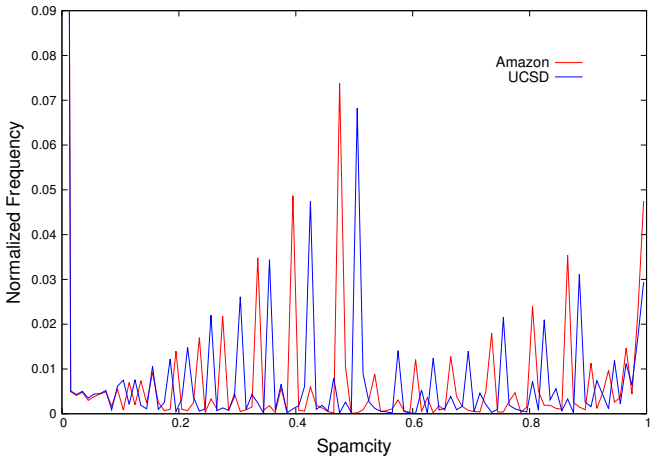
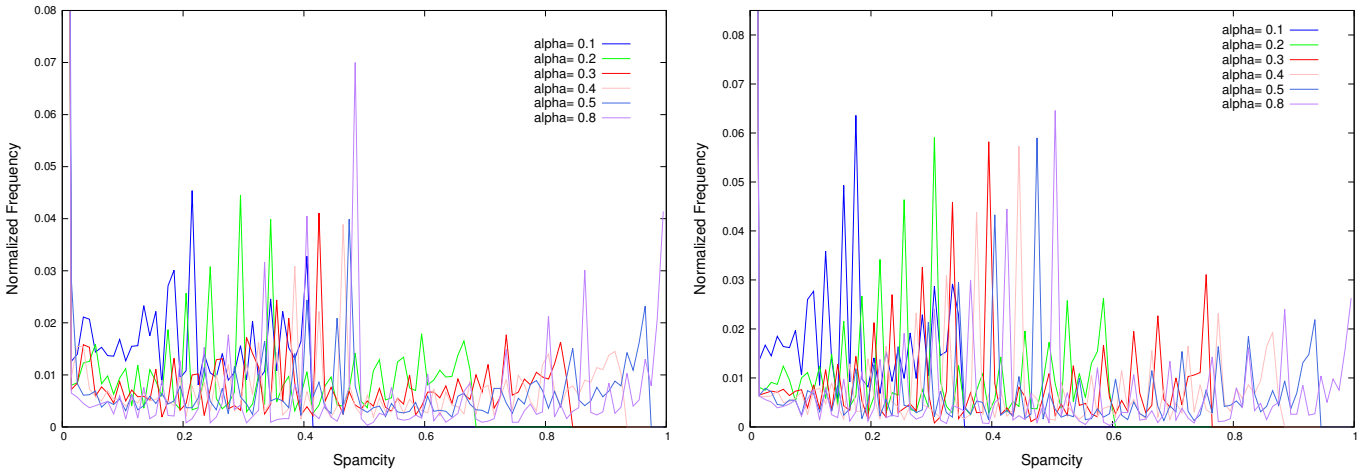


Figure 5: Distributions of Spamcity values across datasets

Figures 6 (a) and (b) illustrate the distributions of the modified Spamcity values calculated as per Algorithm 2 on the Amazon and the UCSD datasets, respectively. The distributions have been plotted against various values of the smoothing factor α. It may be noted that at $\alpha = 0$ in Equation 10, the Spamcity values $S(r)$ will not be updated and would fall back to their initialized values which is 0. On the other hand, as $\alpha \to 1$, the values of $S(r)$ will converge with the values of the intermediate Spamcity values $s(r)$ in Equation 10. Thus, as α varies, the distribution of the modified Spamcity values $S(r)$ should shift towards unity, which can be observed in Figures 6 (a) and (b).

4) Setting the parameter α: As described in Section III-D, the smoothing factor α can be used to control the degree of



(a) Distributions of Modified Spamcity values for Amazon dataset

(b) Distributions of Modified Spamcity values for UCSD dataset

Figure 6: Distributions of Modified Spamcity values

opposition to the bias in the original rating model. Thus, α can be treated as a control parameter for the process. This also means that for different value distributions, the value of α may change to best suit the model. Empirically, the parameter should be set such that the value distribution may become more uniform, with the Spamcity values shifting to the left to accommodate the biasing. At the same time, the distribution should not lose correlation with the original distribution.

The setting of α is illustrated in Figures 7 (a) and (b). The values have been chosen so as to approximate the original values. As can be discerned graphically, the modified Spamcity values are distributed more uniformly. For the Amazon dataset, the resultant uniformity is clearly distinguishable by comparing peak strengths with the original Spamcity values, while it is not so for the UCSD dataset. In both cases, the distributions have shifted slightly to the left. Note that the shift is lesser for the Amazon dataset than for the UCSD dataset due to differences in scale. With further setting using experimentations and trials, the final values of α were set to be 0.4 and 0.62 for the Amazon and UCSD datasets, respectively.

We have extended the work done in [11] to detect spurious reviewer factions in reviewing systems using the strengths of Big Data. Apart from the dataset used by the base model, a larger UCSD dataset is used for the experiment. Table III describes a comparison between the base rating model and the proposed model based on different parameters. Moreover, [11] has a subjective result interpretation based on tabular metrics of subjective spamcity indicators. In contrast, we have used more objective and numerical performance indicators to validate our model in addition to subjective reasoning.

V. Conclusion and Future Work

In this work, a Big Data approach was applied for the detection of review spammers. A metadata-based rating model for detecting review spammers was implemented on large datasets to study the effect of scale on such models. The discrepancies were identified, and mitigations for the same in the form of exponential smoothing were proposed. A distributed computing platform was set up and heterogeneous methods were applied to compute the various spam indicators. After the experiments were conducted, the results were analysed, and the modifications were validated and justified graphically. The findings of this study can be utilized in business (e-commerce platforms) and research domains to detect review spammers and carry out further research.

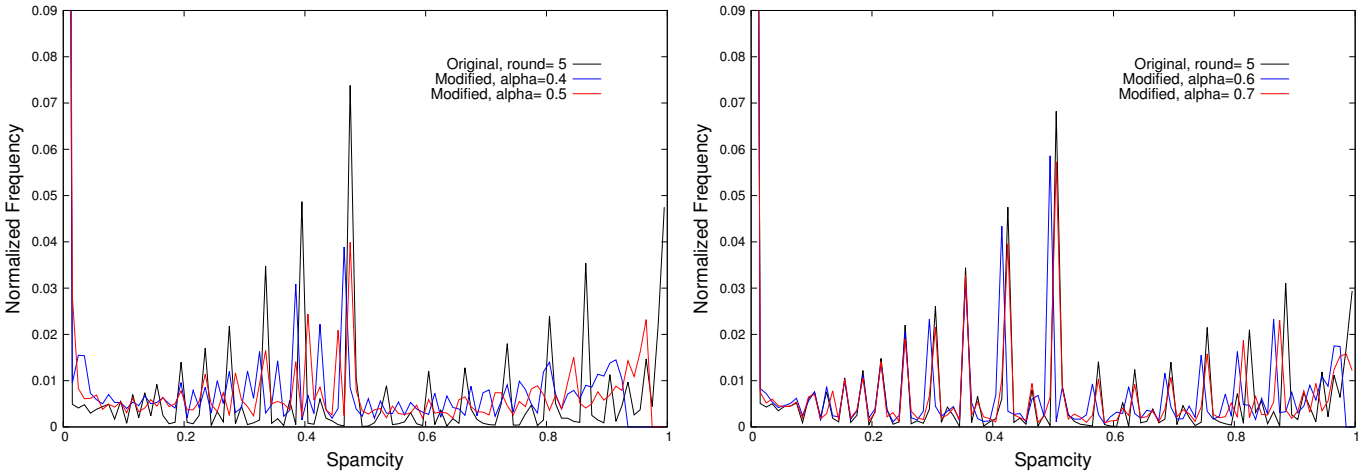
In the future, further scope of improvements may be exploited to develop metadata based models. For tackling Big Data problems, simple processes are desirable to process large volumes of data. Better representatives of prevalent opinion, such as those supplemented by other associated metadata like feedback and textual features, may be chosen to improve the model. Other approaches such as Machine Learning and Graphical Analysis may also be applied using Big Data frameworks for accurately modeling real-world review systems. In-memory Big Data Management systems [93] is also scope for future development. Finally, in our future work, we will focus more on the early detection of spam reviews as well as spammers and will compare with all existing state-of-the-art techniques using the data sets developed in recent times

Acknowledgments

We are very much thankful towards the authors of the base rating model "Detection of opinion spam based on anomalous rating deviation" by Savage et al.(2015). Their work not only inspires us but also motivated us to extend it further for the current findings. Apart from this we are very much thankful to all the reviewers for their advice and feedback, without which the work may be an incomplete one.

References

[1] Q. J. Y. Wang, W. Dai and J. Ma, “Bcinet: A biased contest-based crowdsourcing incentive mechanism through exploiting social networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2926–2937, 2020, DOI: 10.1109/TSMC.2018.2837165.



(a) Setting the value of α for Amazon dataset

(b) Setting the value of α for UCSD dataset

Figure 7: Setting the value of α

Table III: Comparison of proposed model with the base rating model

	Base Rating Mode	Proposed Model
Dataset Size	Amazon Dataset [3]: 6.31 GB 5.8 million reviews by 2.1 million reviewers for 1.2 million products.	Amazon Dataset [3] and UCSD Dataset: 60 GB [91], [92]: 140 million reviews by 237 million reviewers for 10 million products
Time complexity to compute spamicity of reviewers	$\mathcal{O}(maxIterations \times (n_{reviews}+n_{reviewers})+n_{reviewers})$	$\mathcal{O}(maxIterations \times (n_{reviews}+n_{reviewers}))$
Scalability	Not suitable for large scale reviewing system	Can deal with large scale review corpus using Big Data techniques(Hadoop, Spark)
Existence of Bias in spamicity calculation	Bias is there which leads to information loss	Exponential smoothing is used to deal with bias

[2] "Google updates spam detection for reviews, warns seos," http://www.webpronews.com/google-updates-spam-detection-for-reviews-warns-seos-2013-02/, accessed: 2016-07-18.

[3] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 2008, pp. 219–230, DOI: 10.1145/1341531.1341560.

[4] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, pp. 1–24, 2015, DOI: 10.1186/s40537-015-0029-9.

[5] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Systems with Applications*, pp. 3634 – 3642, 2015, DOI: 10.1016/j.eswa.2014.12.029.

[6] R. Mohawesh, S. Xu, S. N. Tran, R. Ollington, M. Springer, Y. Jararweh, and S. Maqsood, "Fake reviews detection: A survey," *IEEE Access*, vol. 9, pp. 65 771–65 802, 2021, DOI:10.1109/ACCESS.2021.3075573.

[7] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 2011, pp. 309–319.

[8] W. Zhang, R. Lau, and C. Li, "Adaptive big data analytics for deceptive review detection in online social media," in *Proceedings of 2014 International Conference on Information Systems*, 2014, pp. 1–19.

[9] D. Zhang, D. Wang, N. Vance, Y. Zhang, and S. Mike, "On scalable and robust truth discovery in big data social media sensing applications," *IEEE Trans. Big Data*, 2018, DOI:10.1109/TBDATA.2018.2824812.

[10] M. Cheung, J. She, and N. Wang, "Characterizing user connections in social media through user shared image," *IEEE Trans. Big Data*, 2017, DOI:10.1109/TBDATA.2017.2762719.

[11] D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang, "Detection of opinion spam based on anomalous rating deviation," *Expert Systems with Applications*, pp. 8650 – 8657, 2015, DOI:10.1016/j.eswa.2015.07.019.

[12] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, 1999, DOI:10.1109/72.788645.

[13] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in *Proceedings of the 15th International Conference on World Wide Web*, 2006, pp. 83–92, DOI:10.1145/1135777.1135794.

[14] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, 2004, pp. 576–587.

[15] N. Jindal and B. Liu, "Analyzing and detecting review spam," in *Proceedings of the Seventh IEEE International Conference on Data Mining*, 2007, pp. 547–552, DOI: 10.1109/ICDM.2007.68.

[16] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 497–501.

[17] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 201–210, DOI: 10.1145/2187836.2187864.

[18] Y. Ren and D. Ji, "Learning to detect deceptive opinion spam: A survey," *IEEE Access*, vol. 7, pp. 42 934–42 945, 2019, DOI: 10.1109/ACCESS.2019.2908495.

[19] J. K. Rout, S. Singh, S. K. Jena, and S. Bakshi, "Deceptive review detection using labeled and unlabeled data," *Multimedia Tools and Applications*, pp. 1–25, 2016, DOI:10.1007/s11042-016-3819-y.

[20] S. Shojaee, M. A. A. Murad, A. Bin Azman, N. M. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *Proceedings of 13th International Conference on Intelligent Systems Design and Applications*, 2013, pp. 53–58, DOI:10.1109/ISDA.2013.6920707.

[21] N. H. Long, P. H. T. Nghia, and N. M. Vuong, "Opinion spam recognition method for online reviews using ontological features," *Tap chi KHOA HoC DHSP TPHCM*, no. 61, p. 44, 2014.

[22] J. Li, M. Ott, C. Cardie, and E. H. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Lingustics*, 2014, pp. 1566–1576.

[23] C. Lai, K. Xu, R. Y. Lau, Y. Li, and L. Jing, "Toward a language modeling approach for consumer review spam detection," in *Proceedings of IEEE 7th International Conference on e-Business Engineering*, 2010, pp. 1–8, DOI: 10.1109/ICEBE.2010.47.

[24] Y. Lin, T. Zhu, X. Wang, J. Zhang, and A. Zhou, "Towards online review spam detection," in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 341–342, DOI:10.1145/2567948.2577293.

[25] Y. Lin, T. Zhu, H. Wu, J. Zhang, X. Wang, and A. Zhou, "Towards online anti-opinion spam: Spotting fake reviews from the review sequence," in *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 2014, pp. 261–264, DOI:10.1109/ASONAM.2014.6921594.

[26] S. P. Algur, A. P. Patil, P. Hiremath, and S. Shivashan, "Conceptual level similarity measure based review spam detection," in *International Conference on Signal and Image Processing*, 2010, pp. 416–423, DOI: 10.1109/ICSIP.2010.5697509.

[27] D. H. Fusilier, M. Montes-y Gómez, P. Rosso, and R. G. Cabrera, "Detection of opinion spam with character n-grams," in *Proceedings of the 16th International Conference on Computational Linguistics and Intelligent Text Processing*, 2015, pp. 285–294, DOI:10.1007/978-3-319-18117-2_21.

[28] S. Banerjee and A. Y. K. Chua, "Applauses in hotel reviews: Genuine or deceptive?" in *2014 Science and Information Conference*, 2014, pp. 938–942, DOI:10.1109/SAI.2014.6918299.

[29] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010, DOI:10.1177/0261927X09351676.

[30] P. Rayson, A. Wilson, and G. Leech, "Grammatical word class variation within the british national corpus sampler," *Language and Computers*, vol. 36, no. 1, pp. 295–306, 2001.

[31] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, 2012, pp. 171–175.

[32] S. Singh, J. K. Rout, and S. K. Jena, "Construct-based sentiment analysis model," in *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, 2016, pp. 171–178, DOI:10.1007/978-81-322-3589-7_18.

[33] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[34] Z. Zhang, L. Jia, M. Zhao, G. Liu, M. Wang, and S. Yan, "Kernel-induced label propagation by mapping for semi-supervised classification," *IEEE Trans. Big Data*, 2018, DOI:10.1109/TBDATA.2018.2797977.

[35] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 542–542, 2009.

[36] J. K. Rout, A. Dalmia, K.-K. R. Choo, S. Bakshi, and S. K. Jena, "Revisiting semi-supervised learning for online deceptive review detection," *IEEE Access*, pp. 1–1, 2017, DOI:10.1109/ACCESS.2017.2655032.

[37] D. Hernández, R. Guzmán, M. Móntes y Gomez, and P. Rosso, "Using PU-learning to detect deceptive opinion spam," in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2013, pp. 38–45.

[38] D. H. Fusilier, M. Montes-y Gómez, P. Rosso, and R. G. Cabrera, "Detecting positive and negative deceptive opinions using PU-learning," *Information Processing & Management*, vol. 51, no. 4, pp. 433–443, 2015, DOI:10.1016/j.ipm.2014.11.001.

[39] I. Ahmed, R. Ali, D. Guan, Y.-K. Lee, S. Lee, and T. Chung, "Semi-supervised learning using frequent itemset and ensemble learning for sms classification," *Expert Systems with Applications*, pp. 1065–1073, 2015, DOI:10.1016/j.eswa.2014.08.054.

[40] F. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence(IJAI)*, vol. 22, no. 3, 2011, pp. 2488–2493.

[41] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12, 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 *Proceedings of the 2013 International AAAI Conference on Web and Social Media*, vol. 13, 2013, pp. 175–184.

[43] C. M. Aye and K. M. Oo, "Review spammer detection by using behaviors based scoring methods," in *Proceedings of international conference on advances in engineering and technology*, 2014.

[44] N. Jindal, B. Liu, and E.-P. Lim, "Finding unusual review patterns using unexpected rules," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 1549–1552, DOI:10.1145/1871437.1871669.

[45] H. Xue, F. Li, H. Seo, and R. Pluretti, "Trust-aware review spam detection," in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 726–733, DOI:10.1109/Trustcom.2015.440.

[46] G. Wang, S. Xie, B. Liu, and S. Y. Philip, "Review graph based online store review spammer detection," in *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011, pp. 1242–1247, DOI:10.1109/ICDM.2011.124.

[47] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Identify online store review spammers via social review graph," *ACM Transactions on Intelligent Systems and Technology*, pp. 61:1–61:21, 2012, DOI:10.1145/2337542.2337546.

[48] D. Liang, X. Liu, and H. Shen, "Detecting spam reviewers by combing reviewer feature and relationship," in *Proceedings 2014 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*, 2014, pp. 102–107, DOI:10.1109/ICCSS.2014.6961824.

[49] S. K. Fayazbakhsh and J. Sinha, "Review spam detection: A network-based approach," *Final Project Report: CSE*, vol. 590, 2012.

[50] Y. Lu, L. Zhang, Y. Xiao, and Y. Li, "Simultaneously detecting fake reviews and review spammers using factor graph model," in *Proceedings of the 5th Annual ACM Web Science Conference*, 2013, pp. 225–233, DOI:10.1145/2464464.2464470.

[51] S. Noekhah, E. Fouladfar, N. Salim, S. H. Ghorashi, and A. A. Hozhabri, "A novel approach for opinion spam detection in e-commerce," in *Proceedings of the 8th International Conference on E-commerce with focus on e-Trust*, 2014, pp. 1–8.

[52] Z. Wang, S. Gu, X. Zhao, and X. Xu, "Graph-based review spammer group detection," *Knowledge and Information Systems*, vol. 55, no. 3, pp. 571–597, 2018, DOI:10.1007/s10115-017-1068-7.

[53] S. Noekhah, N. binti Salim, and N. H. Zakaria, "Opinion spam detection: Using multi-iterative graph-based model," *Information Processing & Management*, vol. 57, no. 1, p. 102140, 2020, DOI:10.1016/j.ipm.2019.102140.

[54] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proceedings of the 2013 International AAAI Conference on Web and Social Media*, vol. 13, 2013, pp. 2–11.

[55] F. B. T. Rodrigues, A. Veloso, J. Almeida, M. Goncalves, and V. Almeida, "Practical detection of spammers and content promoters in online video sharing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 688–701, 2012, DOI:10.1109/TSMCB.2011.2173799.

[56] 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 Transactions on Cybernetics*, vol. 50, no. 4, pp. 1595–1606, 2020, DOI:10.1109/TCYB.2018.2877161.

[57] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "Netspam: A network-based spam detection framework for reviews in online social media," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1585–1595, 2017, DOI:10.1109/TIFS.2017.2675361.

[58] F. Masood, G. Ammad, A. Almogren, A. Abbas, H. A. Khattak, I. U. Din, M. Guizani, and M. Zuair, "Spammer detection and fake user identification on social networks," *IEEE Access*, vol. 7, pp. 68 140–68 152, 2019, DOI:10.1109/ACCESS.2019.2918196.

[59] L. Zhang, Z. Wu, and J. Cao, "Detecting spammer groups from product reviews: A partially supervised learning model," *IEEE Access*, vol. 6, pp. 2559–2568, 2018, DOI:10.1109/ACCESS.2017.2784370.

[60] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, and K. Yu, "Robust spammer detection using collaborative neural network in internet of thing applications," *IEEE Internet of Things Journal*, pp. 1–1, 2020, DOI:10.1109/JIOT.2020.3003802.

[61] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal, "Detecting group review spam," in *Proceedings of the 20th international conference companion on World wide web*, 2011, pp. 93–94, DOI: 10.1145/1963192.1963240.

[62] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 191–200, DOI: 10.1145/2187836.2187863.

[63] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 632–640, DOI: 10.1145/2487575.2487580.

[64] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 939–948, DOI:10.1145/1871437.1871557.

[65] K. Sharma and K.-I. Lin, "Review spam detector with rating consistency check," in *Proceedings of the 51st ACM Southeast Conference*, 2013, pp. 34:1–34:6, DOI: 10.1145/2498328.2500083.

[66] D. Kumar, Y. Shaalan, X. Zhang, and J. Chan, "Identifying singleton spammers via spammer group detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018, pp. 656–667, DOI:10.1007/978-3-319-93034-3_52.

[67] 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, 2019, DOI:10.1016/j.neucom.2019.08.013.

[68] 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, 2021, DOI:10.1016/j.neucom.2020.10.070.

[69] 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, 2019, DOI:10.1007/s11280-018-0614-x.

[70] Z. Wang, S. Gu, and X. Xu, "Gslda: Lda-based group spamming detection in product reviews," *Applied Intelligence*, vol. 48, no. 9, pp. 3094–3107, 2018, DOI:10.1007/s10489-018-1142-1.

[71] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, and K. Yu, "Robust spammer detection using collaborative neural network in internet-of-things applications," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9549–9558, 2021, DOI:10.1109/JIOT.2020.3003802.

[72] Z. Wang, R. Hu, Q. Chen, P. Gao, and X. Xu, "Collueagle: collusive review spammer detection using markov random fields," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1621–1641, 2020, DOI:10.1007/s10618-020-00693-w.

[73] Y. Zhang, S. Hao, and H. Wang, "Detecting incentivized review groups with co-review graph," *High-Confidence Computing*, p. 100006, 2021, DOI:10.1016/j.hcc.2021.100006.

[74] J. Li, P. Lv, W. Xiao, L. Yang, and P. Zhang, "Exploring groups of opinion spam using sentiment analysis guided by nominated topics," *Expert Systems with Applications*, vol. 171, p. 114585, 2021, DOI:10.1016/j.eswa.2021.114585.

[75] F. Zhang, X. Hao, J. Chao, and S. Yuan, "Label propagation-based approach for detecting review spammer groups on e-commerce websites," *Knowledge-Based Systems*, vol. 193, p. 105520, 2020, DOI:10.1016/j.knosys.2020.105520.

[76] S.-j. Ji, Q. Zhang, J. Li, D. K. Chiu, S. Xu, L. Yi, and M. Gong, "A burst-based unsupervised method for detecting review spammer groups," *Information Sciences*, vol. 536, pp. 454–469, 2020, DOI:10.1016/j.ins.2020.05.084.

[77] Y. Liu and B. Pang, "A unified framework for detecting author spamicity by modeling review deviation," *Expert Systems with Applications*, vol. 112, pp. 148–155, 2018, DOI:10.1016/j.eswa.2018.06.028.

[78] M. A. Azad and R. Morla, "Rapid detection of spammers through collaborative information sharing across multiple service providers," *Future Generation Computer Systems*, vol. 95, pp. 841–854, 2019, DOI:10.1016/j.future.2017.12.026.

[79] L.-C. Cheng, H.-W. Hu, and C.-C. Wu, "Spammer group detection using machine learning technology for observation of new spammer behavioral features," *Journal of Global Information Management (JGIM)*, vol. 29, no. 2, pp. 61–76, 2021, DOI:10.4018/JGIM.2021030104.

[80] V. Gupta, A. Aggarwal, and T. Chakraborty, "Detecting and characterizing extremist reviewer groups in online product reviews," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 741–750, 2020, DOI:10.1109/TCSS.2020.2988098.

[81] H. Byun, S. Jeong, and C.-k. Kim, "Sc-com: Spotting collusive community in opinion spam detection," *Information Processing & Management*, vol. 58, no. 4, p. 102593, 2021, DOI:10.1016/j.ipm.2021.102593.

[82] N. Hussain, H. T. Mirza, I. Hussain, F. Iqbal, and I. Memon, "Spam review detection using the linguistic and spammer behavioral methods," *IEEE Access*, vol. 8, pp. 53 801–53 816, 2020, DOI:10.1109/ACCESS.2020.2979226.

[83] M. Z. Asghar, A. Ullah, S. Ahmad, and A. Khan, "Opinion spam detection framework using hybrid classification scheme," *Soft computing*, vol. 24, no. 5, pp. 3475–3498, 2020, DOI:10.1007/s00500-019-04107-y.

[84] H. J. Escalante, E. Villatoro-Tello, S. E. Garza, A. P. López-Monroy, M. Montes-y Gómez, and L. Villaseñor-Pineda, "Early detection of deception and aggressiveness using profile-based representations," *Expert Systems with Applications*, vol. 89, pp. 99–111, 2017, DOI:10.1016/j.eswa.2017.07.040.

[85] A. Heydari, M. Tavakoli, and N. Salim, "Detection of fake opinions using time series," *Expert Systems with Applications*, vol. 58, pp. 83–92, 2016, DOI:10.1016/j.eswa.2016.03.020.

[86] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, "Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace," *Future Generation Computer Systems*, vol. 117, pp. 205–218, 2021, DOI:10.1016/j.future.2020.11.028.

[87] N. Hussain, H. T. Mirza, A. Ali, F. Iqbal, I. Hussain, and M. Kaleem, "Spammer group detection and diversification of customers' reviews," *PeerJ Computer Science*, vol. 7, p. e472, 2021, DOI:10.7717/peerj-cs.472.

[88] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Proceedings of the 2013 International AAAI Conference on Web and Social Media*, 2013.

[89] "Yelp factsheet: To connect people with great local businesses," https://www.yelp.com/factsheet, accessed: 2016-07-18.

[90] "Amazon," https://www.amazon.com/, accessed: 2016-07-18.

[91] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 43–52, DOI:10.1145/2766462.2767755.

[92] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 785–794, DOI:10.1145/2783258.2783381.

[93] Q. Cai, H. Zhang, W. Guo, G. Chen, B. C. Ooi, K. L. Tan, and W. F. Wong, "Memepic: Towards a unified in-memory big data management system," *IEEE Trans. Big Data*, 2018, DOI:10.1109/TBDATA.2017.2789286.