Ingénierie des Systèmes d'Information
Vol. 27, No. 6, December, 2022, pp. 1019-1027
Journal homepage: http://iieta.org/journals/isi

# GrFrauder: A Novel Unsupervised Clustering Algorithm for Identification Group Spam Reviewers

Rathan Kumar Chenoori[1*], Radhika Kavuri[2]

[1] Department of Computer Science and Engineering, Osmania University, Telangana 500007, India
[2] Department of Information Technology, CBIT, Telangana 500075, India

Corresponding Author Email: rathanoucse@gmail.com

https://doi.org/10.18280/isi.270619

**Received:** 15 October 2022
**Accepted:** 20 November 2022

***Keywords:***
*opinion spamming, spammer groups, spam identification, spam indicators, group spam indicators*

**ABSTRACT**

As e-commerce has expanded, people's lives now include some aspect of online buying, because buyers frequently use online product reviews to make purchasing decisions. Merchants frequently collaborate with review spammers to write spam reviews that promote or demote selected items. Spammers who work in groups, in particular, are more dangerous than individual attacks. Previous studies provided various frequent item mining and graph-based techniques to detect such spammer groups. In this paper, we recommend a technique referred to as GrFrauder (Group Fraud detection) method to detect online spam reviewer groups with an unsupervised manner. Our technology identifies spammer candidate groups initially based on product - product review graph and collaboration among reviewers constructed with several behavioral patterns. It then embeds reviewers into an embedding space and calculates spam score for every group; with higher spam scores the model generates ranks for each group. Studies using four real-world datasets reveal that GrFrauder outperforms numerous state-of-the-art baselines in terms of precision and is able to identify more high-quality spammer groups.

## 1. INTRODUCTION

Online product opinions are critical in determining the products (items) quality before customers make purchase decisions. Few years back, machine learning approaches are used for identification of reviews as real or fake because in earlier days very few people might post fake reviews. With the increase of social media applications opinion spamming came into existence. The sentimental analysis has the power either to increase or decrease sales of the product with spam reviewers [1, 2]. In traditional approaches, initially a candidate group with frequent product purchases is formed then ranking of labels are performed. In order to identify the spammers group few patterns like the reviewers who have posted 5 star or 0 star rating and within which time frame they have posted their reviews are used. The model also needs to analyze the number of products viewed by the user. The model has to focus on the collaborative patterns of the different reviewers.

Analyzing the information, including the viewpoints, is crucial. Users are permitted to openly voice their opinions on any part of social media, including product reviews. Opinion mining is now happening. Text analysis or sentiment analysis are other names for opinion mining. The text's context is recognized and extracted using computational linguistics and natural language processing. The viewpoint may be favorable, adverse, or impartial. A person's likes and dislikes can be discovered through opinion mining, which is akin to reading someone's mind. The majority of the article recommendations rely on this opinion mining. Data of any structure can be processed as well. The models used for opinion mining can concentrate on any aspect of the opinion, including subjective thoughts, intentions, and Different types of opinion mining exist. They are presented are in Figure 1.

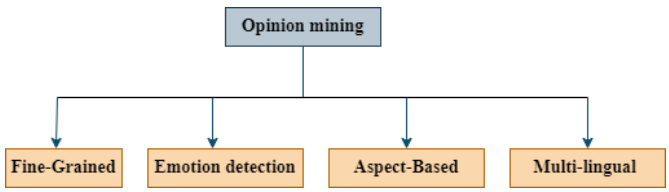


**Figure 1.** Classification of opinion mining techniques

Liu and Jindal originally introduced the difficulty of detecting fake reviews/reviewers in 2008 [1] from online reviews; their method identified individual review spammer from opinion (review) dataset. Researchers then suggested behavior-based [3, 4], probability-based [5, 6], rule-based [7], and graph-based [8, 9] algorithms to identify individual spammers to improve the performance of spam detection. Recently, there has been a movement aimed at detecting group spammers. Deceptive merchants may collaborate with some reviewers to create false reviews to elevate or demote specific items or a service, which is known as opinion spam (fraud). Such reviewers are referred to as spam reviewers, and the items that are targeted are referred to as target products [9-11]. Spam reviewers sometimes collaborate as a group to totally control the targeted items sentiment, split overall effort, and disguise themselves. Such spammers are called spammer groups, these are dangerous than single spam reviewers.

Group spamming detection receives less attention than individual spammer detection. Previous research in group spam detection used the frequent item set mining (FIM) method to create potential spammer clusters then develop

models to detect them.

However, there are many drawbacks for using FIM to generate candidate groups. (i) Due to combinatorial explosion [4], for a large dataset with a huge number of reviewers and products, the minimum support count used in FIM cannot be <3, which implies that only a group working on at least three products can be detected. (ii) Group spammers often have to finish their tasks in a prescribed time limit, whereas the FIM-based method does not take into account the time window when generating candidate spammer groups. (iii) In each FIM candidate group, every reviewer must have reviewed all the common products reviewed by each member of the group. Wang et al. [12] discovered tight spammer groups where everybody in the group must go over all of the items that have been targeted.

Group spammers, on the other hand, frequently work in a looser approach; for example, under certain group spam activities, reviewers really aren't required to assess each target product. In other approaches graph based algorithms were used to construct reviewer graph based on graph potential and clustering algorithms. However, the problem of group spammer was never totally solved owing to a lack of ground-truth data sets.

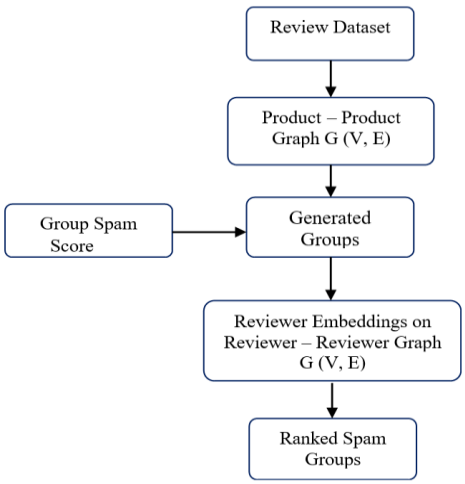


**Figure 2.** Workflow of GrFrauder

GrFrauder, a new structure for detecting spam reviewer groups, is introduced in this paper. As represented the structure of GrFrauder in Figure 2, it contributes (a) detection of fraudulent activity groups using the integration of various coherent reviewers' behavioral signals and (b) A unique ranking approach, by ranking groups depending on their degree of spamicity.

The reminder sections of the paper are organized as follows. Section - II talks over the related work, Section - III provides group spam indicators, which are utilized as features on spammer's groups. Section IV details our GrFrauder method of candidate spammer group's detection and Ranking spam groups. Section V describes the experimental evaluation. Finally, Section VI, summarise this paper.

## 2. RELATED WORK

Byun et al. [13] introduced SC-Com, an enhanced framework for collusive community detection. The approach builds reviewer graphs based on behavior collusion and separates a network into communities based on mutual suspicion. The framework only takes into account main data, such as ratings and time data. The method has a 91% F1 score in a live scenario, making it a trustworthy and solid solution. The spammers' concealment is prohibited under the model. For the purpose of evaluating the framework's effectiveness, the publishers compared 7 various models. The framework helps identify two-way abnormalities. The users will be divided up into smaller communities by the model, and supervised categorization will take place. The work illustrates the importance of collusiveness given spam detection. Only 4 of the dataset's 6 attributes were taken into consideration by the authors. Better outcomes might be obtained using more intricate strategies like deep neural networks. This concept can be applied to issues like cyber bullying, social media, disputes within online communities, etc.

Paul and Nikolaev [14] published a paper. The study is a SLR (Systematic Literature Review) that emphasizes both technical approaches and FRD modeling methodologies. The report also analyses the strategies and laws now in place that have not been successful in eradicating the negative impacts of fake review activity in daily life. Decentralized information systems with user empowerment were emphasized by the writers. Only the user reviews that have been synthesized can be found in reliable sources. The key findings are that expert attackers are well-equipped to mimic the behavior of honest reviews, feature-based approaches require training datasets and cannot function in real-time, training dataset construction is time- and money-consuming, and algorithms that use textual, rating, or graph-based features frequently run slowly and are not scalable. All of the approaches listed fail to distinguish between malicious reviews and honest reviews. This can result in the harassment of trustworthy reviewers.

Hussain et al. [15], the DSR (Diversified Set of Reviews) technique, which chooses a diversified set of top-k evaluations with negative, positive, and unbiased reviews, was proposed by Hussainet al. English and Roman Urdu real-world datasets are used for evaluation. The study suggests using Spammer Group Detection (SGD) identifies spam groups that may be questionable. To identify potential spam groups, the model applies the SCAN (Structural Clustering Algorithm for Networks) algorithm. CNN (Convolutional Neural Network), LSTM, GRU, and BERT are the classifiers that were used. The DSM technique will save reviewers time by allowing them to make decisions about the goods and services without taking into account all of the reviews. The K-fold Cross-validation approach is applied to assess suggested method's performance. For training and testing, the datasets Yelp and Daraz are split 80:20. Dataset can be enhanced with other parameters like email id, spammer's IP address, and location to increase accuracy.

Bhuvaneshwari et al. [16], The Self Attention-based CNN BI-LSTM model, a novel framework built on deep learning, was proposed by Bhuvaneshwari et al. The model determines the relative importance of each word in the phrase and looks for any indicators of spamming in the text. The weighting of each word is determined by the self-attention mechanism. The BI-LSTM layer gains knowledge of the document's structure and extracts contextual data. A non-linear layer with 10 neurons using ReLU as an activation function is provided vectors as input, and a dropout of 0.25 is used. With a decay of 10-6 and a learning rate of 0.0001, RMSprop is employed as an optimizer. For training and testing, the balanced dataset was 70% and 30%, respectively. It is necessary to accomplish

the detection based on aspect level and rating deviance. The use of spam detection algorithms can improve analytical capabilities in several industries, including healthcare, marketing, and law.

Danilchenko et al. [17] described a novel approach that combines ML with a message-passing algorithm to categorize reviewers as spammers or benign. The CRSD-net (Clique Reviewer Spammer Detection Network) model combines the strengths of traditional graphical models like Brief propagation with machine learning capabilities. The model was built using raw data. A careful edge sparsification approach is used to solve computational problems. Belief Propagation, a label propagation algorithm, is utilized for the first time. The edge and node potentials were calculated using the random forest machine learning algorithm. Combining domain experience and ML can improve the performance of the model. This model was only applied to one platform by the authors. Simply user categorization was put into the model. The model can be improved to classify reviews. The Wolfram Language, which has good support for graph structures, was employed by the authors.

Wang et al. [12] developed a novel technique called GSBP to find the loose spam reviewer's group. They initially constructed a reviewer bi-connected graph through relations among the reviewers in which they gave similar rating over a period of time. The dataset is converted in to a bipartite graph as a connection between reviewers and products they reviewed. Then used divide and conquer algorithm to generate loose spammer groups from bipartite graph. On each identified group calculated group spam indicator score and the groups which are above the threshold are treated as spammer groups. They evaluated their model on unlabeled Amazon dataset. Authors identified their previous method GSBP generating reviewer graph revealing the behavioral similarity between two reviewers which is not applicable for all reviewers. To overcome the drawback authors developed a new model called GGSpam [18] to detect spammer groups based on co-review collusiveness. GGSpam initially converts review data as a graph with nodes of reviewers and collusiveness among reviewers as edges. Then based on divide and conquer method graph reduced to smaller groups and with spamity score of each group authors identified the spam groups from dataset.

HIN-RNN was utilized by Shehnepoor et al. [19] to model the co-review relationships of the reviewers in a group over a set period—of 28 days. To forecast the spatiotemporal relationships of the group's reviewers, RNN is also applied to geographical interactions. Thirdly, the vector representations of the reviewers are improved via a GCN (Graph Convolution Network). K-means clustering is used to identify outliers after refining. The authors used a GCN with two layers. The suggested approach consists of three steps: Cluttering to weed out anomalous reviewers, SoWE group level representation, and a fully connected layer. On the available datasets, Continuous Bag of Words (CBoW) is trained. It is not believed that the stochastic modeling of the relationships between the reviewers would offer a more accurate behavioral depiction of the reviewers. All the reviewers remain in the group if there are no outlier reviewers in the group.

Summary of the literature survey is provided in the Table 1, which represents the techniques used by the authors. All of the studies listed in the literature are attempted to generate spammer groups based on the perspective of reviewers either with temporal patterns or Content Similarity and Graph Structure (refer Table 2). But our proposed GrFrauder method which equally contributes on all features and it also makes equal contribution on group detection and ranking of groups.

## 3. GROUP SPAM INDICATORS

This section presents six fraud indicators [12, 19, 20] used in to assess a group's spam score based on language, behavior, structure, and time. They are Review Tightness (RT), Neighbor Tightness (NT), Product Tightness (PT), Rating Variance (RV), Reviewer Ratio (RR) on Product, Average Time Window (TW). All indicator values will be under [0, 1]. Where higher value indicates greater spamming action. Assume that R and P represent all of the reviewers, products. Let P(g) represents the number of targeted goods examined by R, and R(g) represent a group of reviewers. Each reviewer i has rated a certain group of products (P).

### 3.1 Review Tightness (RT)

The *RT(g)* denotes the review tightness for a particular spamming group *g*. It is the proportion between the number of reviews and the number of the reviewer and product sets in a group. The computation is represented in Eq. (1):

$$RT(g) = \frac{|\mathrm{V_g}|}{|\mathrm{R_g}||\mathrm{P_g}|}(L_g) \tag{1}$$

where, $V_g = \sum_{i \in R_g} P_i$, $L_g$ is 0.5 for the smallest spammer group, which consists of two reviewers and one product, and asymptotically approaches 1.0 for larger groups.

### 3.2 Neighbor Tightness (NT)

Compared with genuine reviewer groups the collusion relationship among reviewers in spammer groups is stronger. The *NT* is given in Eq. (2):

$$NT(g) = avg_{r_1, r_2 \in R_g} \frac{|\mathrm{P_{r1}} \bigcap \mathrm{P_{r2}}|}{|\mathrm{P_{r1}} \bigcup \mathrm{P_{r2}}|} \tag{2}$$

where, $P_{r1}$ represents number of products reviewed by reviewer *r1*, $P_{r2}$ represents number of products reviewed by reviewer *r2*.

### 3.3 Product Tightness (PT)

If a group just analyses a small number of items and has never assessed any other products, it is likely that it is an actual opinion spammer group. The total number of products that group g members have all evaluated in common divided by all products reviewed by group members is the product tightness of group *g* and is presented in Eq. (3):

$$PT(g) = \frac{|\bigcap_{r \in R_g} \mathrm{P_r}|}{|\bigcup_{r \in R_g} \mathrm{P_r}|} \tag{3}$$

### 3.4 Rating Variance (RV)

Members of the group intend to target components are

elevated or decreased, thus their rating scores should be equal or alike. *RV* is calculated as shown in Eq. (4):

$$RV(g) = 2\left(1 - \frac{1}{1 + e^{-\text{avg}(_{p \in P_g}) \text{ var(p,g)}}}\right) L(g) \quad (4)$$

where, *var(p,g)* be the variance of the rating scores of product *p* by reviewers in *g*. The larger value of the variance represents the lower spamicity.

### 3.5 Reviewer Ratio (RR) on product

RR is described as, highest proportion of product reviewers in *Rg* on product *p* where all the reviewers of $p \in P_g$ and is represented in Eq. (5):

$$RR(g) = \max_{p \in P_g} \frac{|R_{gp}|}{|R_p|} \quad (5)$$

### 3.6 Average Time Window (TW)

Fraudsters in a group are likely to post fake reviews during a short-time interval. Given a group *g*, and a product $p \in P_g$, we define the time-window based spamicity as:

$$TW(g, p) = \begin{cases} 1 - \frac{SD_p}{T}, SD_P \le T. \\ 0, SD_p > T \end{cases} \quad (6)$$

where, $SD_p$ is the standard deviation of review time for a product *p* reviewed by reviewers in group *g*. *T* is a time threshold (set to 30 days in our experiments suggested in [20]). The group *g TW* is calculated using Eq. (7):

$$TW(g) = avg_{p \in P_g}(TW_p(g, p)L(g)) \quad (7)$$

**Table 1.** Comparative analysis on existing approaches

| Sl no. | Author | Model/ Algorithm | Merits | Demerits |
|---|---|---|---|---|
| 1. | Hyungho Byun | SC-Com framework | Identify 2-way anomalies, supervised classification. | Can be enhanced using deep neural networks. |
| 2. | Himangshu Paul | SLR (Systematic Literature Review) on both technical approaches and FRD modeling methodologies | Evaluated all the existing models and approaches. | Didn't mention the deep learning models. |
| 3. | Naveed Hussain | DSR | Saves reviewers' time, used K-fold cross validation technique to asses. | Can use more parameters for specificity. |
| 4. | P. Bhuvaneswari | CNN BI-LSTM model | Weight of each word is calculated, dense layer of ReLU is used. | Also include aspect level rating for further classification. |
| 5. | Kiril Danil C | GCN with two layers | Edge sparcification is applied. Used Wolfram Language. | Didn't test on other platforms |
| 6. | Saeedreza Shehnepoor | CRSD-net (Clique Reviewer Spammer Detection Network) model | Grouped the users, CBoW is also used. | Should consider grouping relations. |

**Table 2.** Three features using by GrFrauder obtained from user attributes and graph structure

| | GGSpam | SGD | GSCPM | GrFrauder |
|---|---|---|---|---|
| Temporal | ✔ | ✔ | | ✔ |
| Content | | ✔ | ✔ | ✔ |
| Graph | ✔ | | ✔ | ✔ |

## 4. PROPOSED GRFRAUDER FRAMEWORK

The proposed model "GrFrauder" is a two-step approach. The first step identifies the group of candidate spammers by examining the tightness of the product and reviewers based on their neighbourhood from the graph. In the second step, ranking groups depending on the group spam score calculated using co-reviewing pattern in the embedding space.

### 4.1 Detection of candidate spam groups

A group of spammers have similar properties in terms of: a) Reviewed products; b) ratings given to products; c) time spent reviewing products. By incorporating above three factors the model initially construct a graph – G (V, E). It segregates the users based on the products purchased; then, it generates sub-groups based on the similar reviews given by the different users. The proposed system aims to find the group of spammers by finding the neighbourhood properties. The algorithm for the identification of SPAM groups is presented below.

**Algorithm 1: Generate Groups**
**Input:**
P – Set of Products
R – Set of Reviewers
G (V, E) – Product – Product Graph
CGgroups – Empty set of Candidate Groups
**Output:**
Candidate Groups
**Description:**
i. **for** every isolated node $v_i$ in G **do**
ii. CGgroups← remove ($v_i$), add $v_i$ from Graph G
iii. **end for**

iv. **for** every pair $(e_i, e_j) \in E \times E$ do
v. **if** $a_i^e \subset a_j^e$ **then**
vi. **If** $J\left(P_{a_j^e}, P_{a_j^e}\right) > \delta$ **then**
vii. CGgroups.add$\left(a_i^e \cup a_j^e\right)$
viii. **end if**
ix. **end if**
x. remove $a_i^e$ from G
xi. **end for**
xii. **for** every group $g \in CGgroups$**do**
xiii. **if** SpamScore(g) <= ¥$_{spam,}$ **then**
xiv. CGgroups.delete(g)
xv. **end if**
xvi. **end for**

As mentioned, the graph – G(V,E), where $v_{ij} \in V$ represents a pair of product–rating ($p_i$, $r_j$) and this node attribute $a_{ij}^v$ gives $r_j$ rating given reviewers set on $p_i$. Example: P1-1 {R1, R2, R3}, it means Product1 with rating 1 given by set of reviewers {R1, R2, R3}. Edge $e_{(ij,mn)} = (u_{ij}, v_{mn}) \in E$, it explains two ratings (j, n) and two products (i, m) co-rating & co-reviewing patterns. The edge attribute $a_{(ij,mn)}^e$ represents the co-reviewers $R_{(ij,mn)}$ who gave review on same Γt time with similar ratings rj and rn on pi and pm products. It is important to note that an edge linking the same product with various rating values will not exist in G since we consider that a reviewer is not entitled to provide many reviews/ratings for a single product.

Then, model build up Candidate Groups, from a unique group detection method which accepts G as an input. Lines 1-3 are used to remove isolated nodes in a given Graph G. Then each edge on graph with the subset of reviewers who reviewed products (i, j) (Lines 4-11) such reviewers Jaccard Similarity (JS) value is above the edge weight threshold $\delta$=0.4 as taken [20] are merged as a group and edge is removed from the graph. The iteration process is repeated until no edges stay in the resulting G and a list of candidate groups are returned. SpamScore is defined as the average of RT, NT, PT, RV, RR and TW mentioned in section 3, where the spam score will be used to assess the suspiciousness of candidate groups, and believe those groups as potential groups whose SpamScore is above threshold value ¥spam (Lines 12-16).

### 4.2 Ranking of candidate spam groups

GrFrauder evaluates candidateSpamgroups based on spamicity after detecting them. It consists of two steps: based on their co-reviewing tendencies, placing reviewers into an embedding space, ranking groups according to how near the embedding space the member reviewers are. Our proposed ranking strategy is different than other researchers ranking strategy [11, 18] authors used average of group spam indicator values to generate rank of a spammers group.

**Reviewer2Vec: Embedding Reviewers**

The motivation for proposed research suggested Reviewer2Vec embedding technique [11]. To identify the collusive spamicity between the reviewers identified from Candidate Groups, we constructed an individual reviewer-reviewer spammer graph for each group Gg=(Vg, Eg), which is bi-connected and weighted. Here, Vg stands for the set of reviewers R, while Eg is the edge that connects two reviewers $i$ and $j$ with the weight $W_{ij}^g = \omega(i,j)$.

The weight of the edge represents the overall collusive spamicity between two reviewers.

When two ($i$, $j$) reviewers given reviews $r_p^i$ and $r_p^j$, ratings $b_p^i$ and $b_p^j$ at different times $t_p^i$ and $t_p^j$, on product $p$, then the collusive spamicity of $i, j$ with regard to product $p$ is identified as in Eq. (8):

$$Coll(i,j,p) = \begin{cases} 0, |\, t_p^i - t_p^j \,| > \tau_t \vee |\, b_p^i - b_p^j \,| \geq \tau_b \\ \zeta(i,j,p), else \end{cases} \quad (8a)$$

$$\zeta(i,j,p) \equiv s^p \left[ \alpha\left(1 - \frac{|\, t_p^i - t_p^j \,|}{\tau_t}\right) + \beta\left(1 - \frac{|\, b_p^i - b_p^j \,|}{\tau_b}\right) + \gamma.Cosine\left(r_p^i, r_p^j\right) \right] \quad (8b)$$

$\alpha, \beta, \gamma$ are the coefficients used to control the significance of time, rating and review text similarity ($0 \leq \alpha, \beta, \gamma \leq 1$ and sum of $\alpha, \beta, \gamma$=1). $\zeta(i,j,p)$ gives collusion of reviewer's in Eq. (9).

$$s^p = \frac{2}{1 + e^{-(MP - \deg(p))^\theta + 2^\theta}} - 1 \quad (9)$$

$S^p$ represents the product $p$ suspicion degree and the parameters of the model are $\tau_t$, $\tau_r$, $\theta$.

Model does not take into account if the co-reviewing patterns of the ratings difference ($\tau_r$) or time difference ($\tau_t$) between reviewers $i$ and $j$ on $p$ deviates threshold value. After translating two reviews using Word2Vec into an embedding space, model take the cosine of each. After adding the collusive spamicity of a pair of reviewers, model is able to calculate the in general collusive spamicity between reviewers $i, j$ as in 10:

$$\omega(i,j) = \frac{2}{1 + e^{-\sigma(i,j)}} - 1 \quad (10a)$$

where,

$$\sigma(i,j) = \left[ \sum_{p \in \{P_i \cap P_j\}} coll(i,j,p) \right] \frac{|\, P_i \cap P_j \,|}{|\, P_i \cup P_j \,|} \quad (10b)$$

When the evaluation is more similar among two reviewers on majority of items then collusion ill occur as shown by $\sigma(i, j)$. Using the Sigmoid function, $\sigma(i, j)$ may be seen as the normalization of $\omega(i, j)$.

**Ranking the Groups:**

$$Gspam(g) = \frac{1}{|\, R_{(g)} \,|} \sum_{k \in R_{(g)}} \left[ \vec{k} - \left[ \frac{1}{|\, R_{(g)} \,|} \sum_{k \in R_{(g)}} \vec{k} \right] \right]^2 \quad (11)$$

To rank detected groups model measure the (Euclidean) distance between each reviewer in the group and the centroid of the group within the embedding space is used to calculate the density of each detected group. Let $\vec{k}$ is the reviewer $k$ representation in embedding space. The mean of all distances is used to determine a group's spamicity. The group spam score Gspam(g) of any group $g$ is calculated as in Eq. (11). The

group with high spam score is represented as first spam group and so on. The top ranked spammers are more dangerous in e-commerce applications.

## 5. EXPERIMENTAL EVALUATION

### 5.1 Datasets

We use four real world datasets to validate our GrFrauder technique. Table 3 shows the summary of the four datasets, as first column explains the used two labeled datasets (YelpNYC and YelpZip) and used two non labeled datasets (AmazonBooks, AmazonCDs) and remaining columns shows the number of reviewers, reviews, products and time span of collected reviews in each dataset. The two Yelp datasets compiled [21, 22] that include hotel/restaurant ratings from New York City and a range of places with zip-codes starting in New York City. The two datasets contain both filtered and recommended reviews detected by the Yelp anti-fraud detection mechanism, putting them close to ground truth. Multiple reviews are not required for the Yelp databases. AmazonBooks, on the other hand, provides book reviews derived from the Amazon dataset collected in 2006, which includes reviews from 1996 to 2006, also used in references [1, 11, 12]. We also examined a more current Amazon dataset, AmazonCDs, which includes CD and Vinyl evaluations from 2012 to 2014. This is because spam methods have probably improved significantly since 2006.

**Table 3.** Statistics of datasets

| Datasets | #Reviews (% of filtered) | #Reviewers (% of spammer) | #Products | Time span | Labeled |
|---|---|---|---|---|---|
| YelpNYC | 3,59,052<br>10.27% | 1,60,225<br>17.79% | 923 | Nov.2004 to Jan. 2015 | Yes |
| YelpZip | 6,08,598<br>13.22% | 2,60,277<br>23.91% | 5,044 | Nov.2004 to Jan. 2015 | Yes |
| AmazonBooks | 11,58,930 | 1,45,942 | 1,97,038 | May 1996 to May2006 | No |
| AmazonCDs | 9,72,105 | 1,05,536 | 1,18,122 | Jan.2012 to July 2014 | No |

### 5.2 Compared baselines

To check the effectiveness of GrFrauder, we compare three graph-based approaches. They are SGD [15], GGSpam [18], GSCPM [11]. In which GGSpam, GSCPM and SGD were performed on Yelp Datasets.

**Table 4.** Competing methods comparison among SGD, GGSpam, GSCPM, GrFrauder on number of group's detected from YelpNYC, YelpZip datasets

| Methods | YelpNYC No of Groups | YelpZip No of Groups |
|---|---|---|
| **GGSpam** | 1218 | 1167 |
| **GSCPM** | 1200 | 1650 |
| **SGD** | 1180 | 1720 |
| **GrFrauder** | 1120 | 1924 |

### 5.3 Performance on labeled datasets

According to reference [18], the optimal assessment criteria for spam groups are the cumulative distribution (CDF) of review content similarity (RCS). Model removes groups of less than two people.

**Review Content Similarity (RCS)** is used to find review text similarity among reviewers in a group; it is calculated from Eq. (12):

$$RCS(g) = \max_{p \in P(g)} \left\{ \frac{1}{|R(g)|^2} \sum_{(i,j) \in R(g) X R(g)} cosine(c_p^i, c_p^j) \right\} \quad (12)$$

The larger each distribution's divergence from the vertical axis (measured in terms of Earth Mover's Distance (EMD)), the better the detection technique [20]. Figure 3 represents the RCS value comparison of GrFrauder along with other base line models on both YelpNYC, YelpZIP datasets. Based on parameter selection technique, model choose the following default parameter values as $\tau_t$=20days, $\tau_{spam}$=0.4. Table 4 reports the number of groups model find across various datasets and quantitative summary (in terms of EMD).

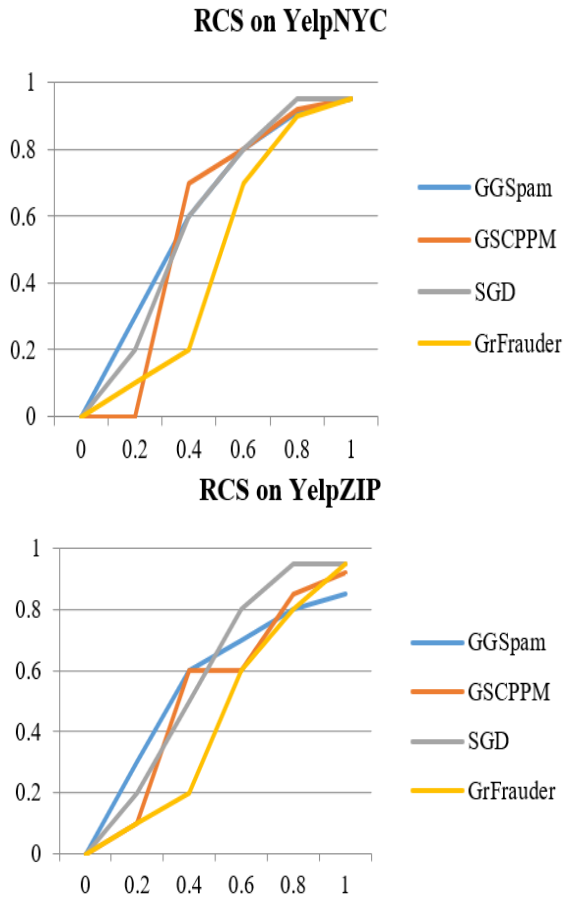


**Figure 3.** RCS comparison among different models on YelpNYC and YelpZIP datasets

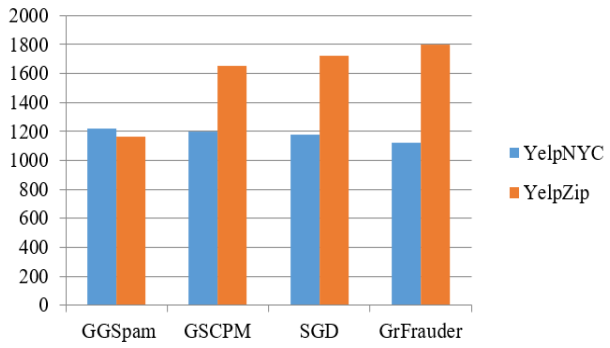


**Figure 4.** Number of Identified Groups

Figure 4 represents the number of groups recognized by the various traditional algorithms along with the proposed algorithm. Out of all, the proposed algorithm has recognized more number of groups on two datasets. X-axis denotes the algorithms and Y-axis denotes the number of groups identified.

As said Yelp datasets contain near-ground-truth, we can evaluate the precision of GrFrauder by checking each review(er) in the detected spammer groups. We take a reviewer as fake if and only if she/he has written at least 1 fake review in the group. Therefore, we can check the top-ranked review(er)s according to the detected top 1,000 groups they belong to Figure 5, Figure 6 shows the reviewer precision and review precision at top k review(er)s ($k \leq 1{,}000$) for GrFrauder and other three baselines on YelpNYC and YelpZIP dataset. We can see that the precision decreases as the number of review(er)s gets larger. In general, GrFrauder outperforms all the 3 baselines.

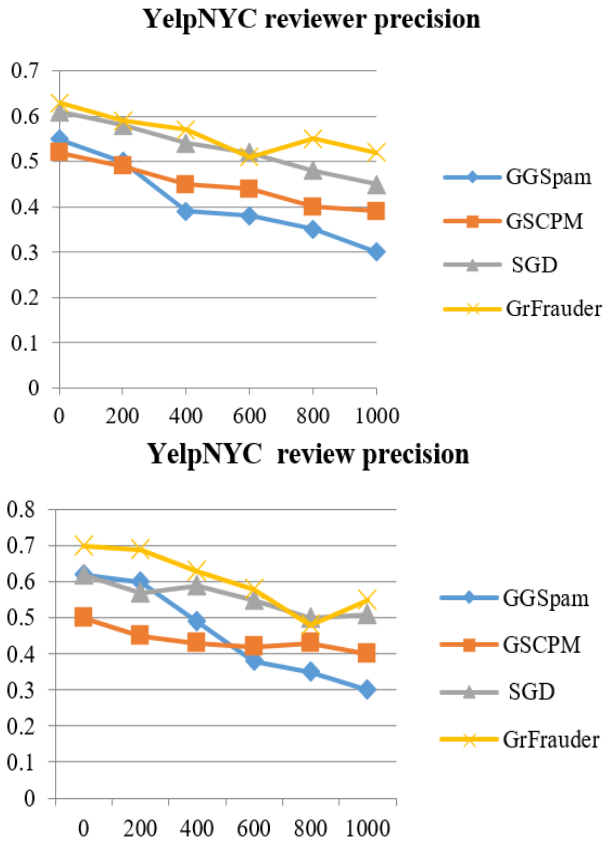


**Figure 5.** Comparison of precision for top review(er)s on the YelpNYC dataset

**Ranking Algorithm Evaluation**

In Figure 7, NDCG@k used to evaluate algorithm, which is a standardized assessment metric. Model used ground-truth score for every group as the percentage of reviewers in that group that are identified as fraudulent since all reviewers are given a fraud/genuine designation. The competing methods are then used to rank the candidate groups, and the top k groups are evaluated using NDCG.

For embedding, model makes use of Node2Vec. Since GrFrauder generates superior groups, model additionally examines how each baseline's ranking approach works on the GrFrauder-detected groupings. Figure 8 shows that, at initial stage GGSpam and GSBP outperform better compared to GrFrauder, after that GrFrauder outperforms the others. However, GrFrauder performs 17.11% higher compared to NDCG@50 than the best baseline (varies among datasets) (averaged over all the datasets).

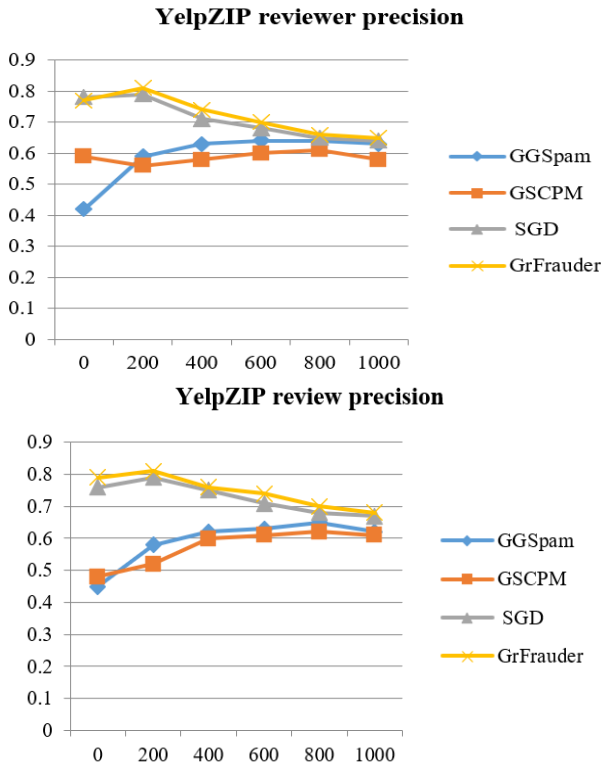


**Figure 6.** Comparison of precision for top review(er)s on the YelpZIP dataset

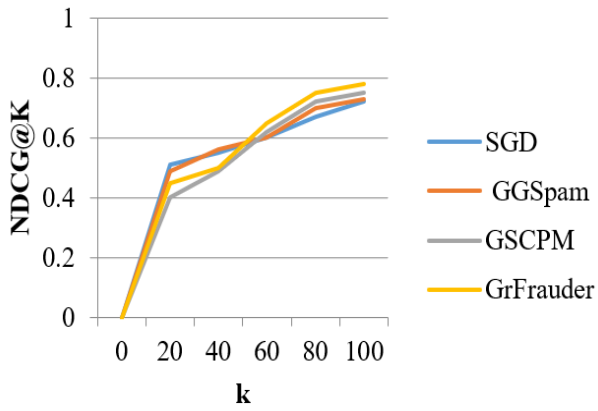


**Figure 7.** Performance on YelpNYC dataset

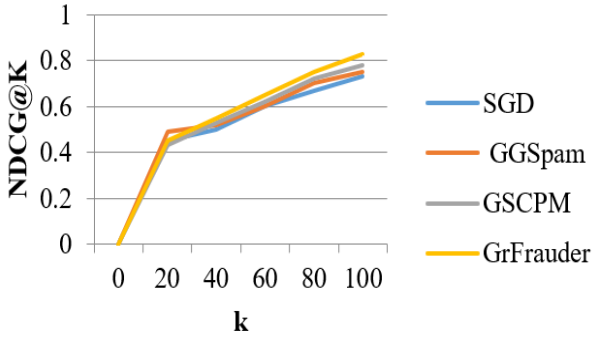


**Figure 8.** Performance on YelpZip dataset

### 5.4 Performance on unlabeled datasets

We evaluated GrFrauder against GGSpam and GSBP [8] on the two unlabeled Amazon datasets. It is challenging to directly assess the efficacy of various spam detection techniques since datasets are unlabeled. We evaluated the cumulative distribution function (CDF) values for spam indicators for the spammer groups found by GrFrauder, using the concept from [23] which helps to assesses the spamicity of detected spammer groups via spam indicators, and compared with GGSpam and GSBP.

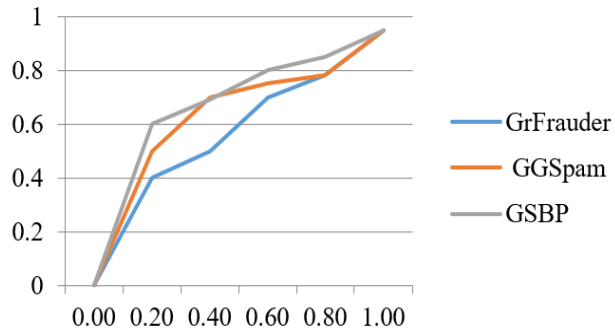


**Figure 9.** CDF based group size (GS) of GrFrauder, GGSpam and GSBP on AmazonBooks

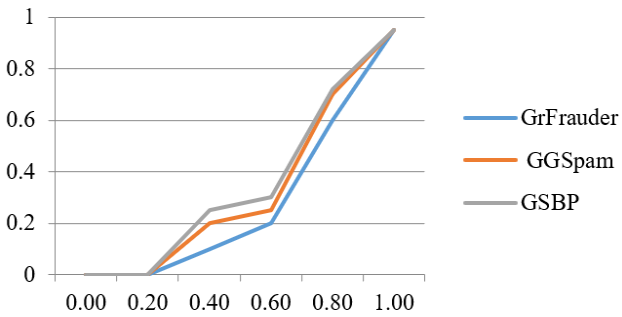


**Figure 10.** CDF based group size (GS) of GrFrauder, GGSpam and GSBP on AmazonCDs

For fairness, we fetched the top 500 spammer groups detected by GrFrauder, GGSpam and GSBP. The six group spam indicators listed in Section 3 are then computed. Their average value (AVG) for spammer groups detected by GGSpam(red line), GSBP (ash line) and GrFrauder (blue line) on dataset AmazonBooks and AmazonCDs are represented in Figures 9 and 10.

The closer the curve is to the vertical axis, the smaller their indicator values will be. We can see that, for most of the indicators, GrFrauder generates higher scores than GSBP and GGSpam.

## 6. CONCLUSIONS

For online review systems, the challenge of fake review identification has grown in importance. The predominant method of review spamming today is group spamming. According to the notion of coherence, which is formed among spam reviewers in a group in terms of co-reviewing patterns, model introduced GrFrauder, a novel approach to locate and rank spam reviewer groups. Two Yelp.com real-world review datasets are utilized to test the effectiveness of suggested strategy. According to experimental findings, suggested strategy outperforms three other methods when it comes to ranking and group detection. Future work includes, applying the GrFrauder as an automatic identification of group spammer's method in recommender system and developing a genuine recommendations to the users. This will improve the genuinely in recommendations and performance of the recommender system.

## REFERENCES

[1] Jindal, N., Liu, B. (2008). Opinion spam and analysis. In: Proceedings of the (2008), international conference on web search and data mining, ACM, New York, pp. 219-230. https://doi.org/10.1145/1341531.1341560

[2] Rathan Kumar, C.H., Dr. Radhika, K. (2018). Detection of review spam and review spammers group survey. International Journal of Research In Electronics and Computer Engineering, 6(4).

[3] Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W. (2010). Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 939-948. https://doi.org/10.1145/1871437.1871557

[4] Theodoros, L. (2012). Fake reviews: The malicious perspective natural language processing and information systems. In 17th International Conference on Applications of Natural Language to Information Systems, pp. 23-34. https://doi.org/10.1007/978-3-642-31178-9_3

[5] Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., Ghosh, R. (2013). Spotting opinion spammers using behavioral footprints. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 632-640. https://doi.org/10.1145/2487575.2487580

[6] Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R. (2013). Exploiting burstiness in reviews for review spammer detection. In Proceedings of the International AAAI Conference on Web and Social Media, 7(1): 175-184. https://ojs.aaai.org/index.php/ICWSM/article/view/14400

[7] Jindal, N., Liu, B., Lim, E.P. (2010). Finding unusual review patterns using unexpected rules. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1549-1552. https://doi.org/10.1145/1871437.1871669

[8] Wang, G., Xie, S., Liu, B., Philip, S.Y. (2011). Review

graph based online store review spammer detection. In 2011 IEEE 11th International Conference on Data Mining, pp. 1242-1247. https://doi.org/10.1109/ICDM.2011.124

[9] Kumar, C.H.R., Dr. Radhika, K. (2020). Timeline based trust dissemination towards review spam detection in online reviews. IJARET, 11(12). https://doi.org/10.34218/IJARET.11.12.2020.185

[10] Metlapalli, A.C., Muthusamy, T., Battula, B.P. (2020). Classification of social media text spam using VAE-CNN and LSTM model. Ingénierie des Systèmes d'Information, 25(6): 747-753. https://doi.org/10.18280/isi.250605

[11] Xu, G., Hu, M., Ma, C., Daneshmand, M. (2019). GSCPM: CPM-based group spamming detection in online product reviews. In ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-6. https://doi.org/10.1109/ICC.2019.8761650

[12] Wang, Z., Hou, T., Song, D., Li, Z., Kong, T. (2016). Detecting review spammer groups via bipartite graph projection. The Computer Journal, 59(6): 861-874. https://doi.org/10.1007/s10115-017-1068-7

[13] Byun, H., Jeong, S., Kim, C. (2021). SC-Com: Spotting collusive community in opinion spam detection. Information Processing & Management, 58(4): 102593. https://doi.org/10.1016/j.ipm.2021.102593

[14] Paul, H., Nikolaev, A. (2021). Fake review detection on online E-commerce platforms: A systematic literature review. Data Mining and Knowledge Discovery, 35(5): 1830-1881. https://doi.org/10.1007/s10618-021-00772-6

[15] Hussain, N., Mirza, H.T., Ali, A., Iqbal, F., Hussain, I., Kaleem, M. (2021). Spammer group detection and diversification of customers' reviews. PeerJ Computer Science, 7: e472. https://doi.org/10.7717/peerj-cs.472

[16] Bhuvaneshwari, P., Rao, A.N., Robinson, Y.H. (2021). Spam review detection using self attention based CNN and bi-directional LSTM. Multimedia Tools and Applications, 80(12): 18107-18124. https://doi.org/10.1007/s11042-021-10602-y

[17] Danilchenko, K., Segal, M., Vilenchik, D. (2022). Opinion spam detection: A new approach using machine learning and network-based algorithms. https://ojs.aaai.org/index.php/ICWSM/article/view/19278.

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

[19] Shehnepoor, S., Togneri, R., Liu, W., Bennamoun, M. (2022). Spatio-temporal graph representation learning for fraudster group detection. arXiv preprint arXiv:2201.02621. https://doi.org/10.48550/arXiv.2201.02621

[20] Mukherjee, A., Liu, B., Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. In Proceedings of the 21st International Conference on World Wide Web, pp. 191-200. https://doi.org/10.1145/2187836.2187863

[21] Rayana, S., Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. In Proceedings of the 21th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 985-994. https://doi.org/10.1145/2783258.2783370

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

[23] Choo, E., Yu, T., Chi, M. (2015). Detecting opinion spammer groups through community discovery and sentiment analysis. In IFIP Annual Conference on Data and Applications Security and Privacy, pp. 170-187. https://doi.org/10.1007/978-3-319-20810-7_11