A mathematical model to cluster reviewers for online review system

Runa Ganguli[1], Akash Mehta[2], Takaaki Goto[3], Soumya Sen[1]
[1]A.K.Choudhury School of Information Technology, University of Calcutta, Kolkata, India
[2]Department of Computer Science, The Bhawanipur Education Society College, Kolkata, India
[3]Department of Information Sciences and Arts, Toyo University, Saitama, Japan

Article Info

Article history:

Received Nov 29, 2024
Revised Oct 31, 2025
Accepted Dec 6, 2025

Keywords:

Average opinion variance
Jenks natural breaks optimization
Online review system
Reviewer clustering
Weighted reviewer score

ABSTRACT

Online business models accept reviews or feedback from customers which are processed and analyzed for important business decisions. Online reviews are helpful to understand the usefulness or popularity of a product. However, it has been observed that sometimes fake reviews are frequently used to boost the popularity of one's own product or to damage reputation of competitors' products. Henceforth it is an interesting research problem to validate reviews or trustworthiness of reviewers. In this paper, a mathematical model is introduced to rate and cluster reviewers based on relevant parameters. It has been observed from business intelligence perspective, that grouping reviewers into different clusters, rather than ranking them individually based on their authenticity, would be more beneficial for potential buyers to understand the quality of reviewers. In the proposed model, clustering is performed using two weighted scores based on average opinion variance and product price. The mean shift clustering algorithm is used to dynamically slab the product price attribute while Jenks Natural Breaks Optimization (JNBO) method and K-means algorithm are applied for the reviewer clustering. Further this research work analyses the impact of product price on reviewer rating and validates the result using t-test statistical method. The proposed methodology is experimented on Amazon datasets to show efficacy of the model.

Corresponding Author:

Runa Ganguli
A.K.Choudhury School of Information Technology, University of Calcutta
JD-2 Sector-3, Saltlake, Kolkata-700106, West Bengal, India
Email: runa.ganguli@gmail.com

1. INTRODUCTION

Online marketing offers direct, effective ways to reach target consumers and grow business. Electronic markets [1], [2] provide buyers 24/7 access to compare and choose from a wide range of products based on prices, images and descriptions for making informed decisions. Customers want real feedback and recommendations [3] about the products from previous buyers for making purchase. With growing volume of online marketing, the number of reviews made by the customers about any product or service is also growing significantly. There is even a growing tendency among merchants to hire professionals to write deceptive reviews. This triggers many reviewers to become dishonest and post fake reviews. Researchers have developed various methods to detect spam or fake reviews [4], [5] in the last few years to provide customers as well as companies with genuine reviews. Companies like Amazon and Flipkart often sell the same type of product from different brands. By analyzing customer feedback, they can identify poorly rated brands and eliminate them from their listings. Authentic reviews are essential in helping customers to make informed decisions and also support e-commerce [6] businesses in refining their strategies and offerings. The review system [7] can be classified into two types [8]. An open review system lets anyone post feedback without verification, making it vulnerable to manipulation by real or fake users. In contrast, a closed review system restricts reviews to verified buyers, but still may contain fake or spam reviews. Thus spam reviewer detection [9]-[11] is an interesting and relevant research area.

Nowadays, recommendation systems [12], [13] have become a extensively researched topic in both computational systems and business intelligence. This is primarily because of their extensive applications in the field of advanced science and technology. Machine learning [14]-[16] and statistical methods [17]-[19] are widely used in the study of recommendation systems. Online review analysis [20], [21] is crucial for trusted information regarding e-commerce recommendations [22], [23]. The authenticity of these reviews and identifying fake reviewers contribute to informed business decisions. Hussain *et al.* [24] introduced two spam review detection methods: behavioural (SRD-BM) and linguistic (SRD-LM) assessing thirteen spammers' behavioural features and few linguistic parameters. In a study by Zhong *et al.* [25], reviewer reputation scores were calculated based on content-related factors and reviewer activity. Saini *et al.* [26] presented a combined model of K-means clustering and artificial bee colony algorithm for feature selection and cluster head optimization for detecting spam reviews. Gupta *et al.* [27] proposed a feature-based supervised model to classify candidate groups as extremist reviewer groups in online product reviews. The authors have used the frequent itemset mining (FIM) method followed by a three-layer perceptron-based classifier. In a study, Bai *et al.* [28] introduced a margin-based embedding ranking model (MERM) to predict a group of early reviewers for more effective product marketing. Xing and Zhao [29] proposed a collaborative training-based algorithm for detecting spammer groups using DBSCAN clustering. In another work by Wang *et al.* [30], a Markov random field (MRF)-based method named ColluEagleis proposed to detect collusive review spammers, as well as review spam campaigns. Here authors have exploited co-review behaviour and used loopy belief propagation to evaluate the suspiciousness of reviewers. Zhang *et al.* [31] in their paper introduced a new ranking aggregation method based on the characteristics of collusive attacks by spammer groups. Their objective is to optimize spammer ranking algorithm by re-calculating the spamicity score for each reviewer using spam indicators. Graph theoretic methods have proven to be valuable for analyzing data in recommendation systems, and extensive research has been conducted in this area. In a study by Xu *et al.* [32] a graph theoretic model called Group Spammers Clique Percolation Method (GSCPM) is proposed to identify group spammers. Clique percolation method (CPM) models behavioral and relational data as a graph of suspicious reviewers, forming k-clique clusters of potential spammers. In another study by Chenoori and Kavuri [9], an unsupervised method named GrFrauder is proposed which initially works of product-product review graph. The authors used coherent behavioral signals to detect fraudulent groups, then applied reviewer embedding and group ranking. A fake reviewer group detection method was proposed in a paper by Cao *et al.* [33] named REAL (modularity based graph clustering). The method uses the concept of Graph convolutional neural network [34] and spectral modularity for graph clustering, finding candidate groups. Rathore *et al.* [35] has shown use of DeepWalk embedding based approach followed by Modified PCKMeans to identify group of fake reviewers. Sundar *et al.* [36] used a deep dynamic structure learning on an extrapolated bipartite graph with unsupervised learning techniques for detecting fake reviewers. Wang *et al.* [37] proposed an algorithm for detecting overlapping spammer groups called DRL-OSG, which utilizes deep reinforcement learning. Verifying the authenticity and trustworthiness of reviewers is crucial to prevent misleading potential online customers and also for better business. It has been found that various existing research works focused on identifying, ranking, and grouping review spammers, whereas no such work is done on analyzing them based on the price of the reviewed product. In this research work, our focus is on identifying trustworthy group of reviewers rather than content-based spam review detection.

The objective of our research work is to determine online reviewers who have a high value of authenticity in writing reviews. The problem is to cluster online reviewers based on certain parameters into groups marked as good, bad and average. Additionally, a specific investigation is whether product price has any role to impact quality of reviews. Our specific objectives are mentioned below:

- Analyzing the reviewers to find out who are the active reviewers with high value of trustworthiness.
- Analyzing the products to find out which are the popular products reviewed by many reviewers.
- Use a computed reviewer score as a measure to detect the authenticity of the reviewers.
- Clustering the reviewers based on rating opinion and reviewed product price.
- Analyzing whether review quality is dependent on product price.

In this research paper, we give emphasis on reviewer rating pattern, deviation from average rating opinion and other relevant parameters to assign a score to each reviewer. This score-based computation helps to identify clusters of reviewers and also examines the significance of a product's price in influencing reviewers' evaluations. Product price is chosen as a feature in this research work as it is one of the decisive factors for the customers purchasing product. It is crucial for organizations to assess whether customer reviews are driven by the price of product. If it is found that price is influencing review quality, then in that case companies need to ensure rigorous testing and validation for high priced products. Conversely, if price is not influencing review quality, companies need to treat every product equally regardless of price. Failing to do so may lead to customers forming incorrect perceptions about the company, potentially harming their business. This is the novelty of our work that aims at studying reviewer behavior based on both average opinion variance and product price in online platforms and helping buyers to focus on reliable reviews only. The specific contributions of this research work are listed below:

– Determining the trustworthiness of the reviewers and to cluster them in different categories.
– Helping the possible customers who are checking the reviews in online system to understand the quality of reviewers.
– Analyzing whether the quality of review by reviewers are driven by the product price.

The paper is organized into the following sections. Section 2 discusses about the proposed method and algorithm for reviewer clustering. The implementation and the experimental results are illustrated in section 3 along with comparative analysis. Finally, the paper is concluded in section 4.

2. METHOD

In this section, the proposed method is described. Subsection 2.1 discusses about a reviewer ranking methodology which is extended to reviewer clustering concept and price analysis in this research work.

2.1. Reviewer ranking

A score-based reviewer ranking model was presented to assign a score and rank online reviewers based on their product rating pattern [38]. This computed score is helpful to trust reviewers and their product/service reviews. The rating score was marked on a scale of 0 to 5. This analysis was performed on a popular dataset [38] where reviewers were judged based on the reviewer rating attribute. If the review score given by a user is close to the average review score of that product the reviewers get higher priority (as it matches the opinion of the majority). Based on this computed priority, ranking is done which acts as the weight for calculating the reviewer score.

The main objective of reviewer ranking was based on two factors: i) |reviewer rating – average product rating| (lower difference gets higher priority) and ii) the rating difference gets more weightage for products with high number of reviews.

The product rank based on review count (Prod_Rank) was calculated by assigning a rank beginning from 1 in ascending order based on the product's review count in descending order. Next, weighted rating difference (WD) against each Product ID is defined by (1):

$$WDi = (Prod\_Ranki) * |Reviewer\ Rating - Average\ Product\ Rating| \tag{1}$$

For each reviewer, a reviewer score (R_Score) was calculated which gave the average difference in rating opinion defined by (2):

$$R\_Score = Average\ of\ all\ WDi\ for\ Reviewer\ Ri \tag{2}$$

Based on review count, reviewer rank (Rev_Rank) was also similarly calculated as product rank. Finally, weighted reviewer score (WR_Score) for each reviewer as defined by (3) was calculated giving high priority to more active reviewers over others. A reviewer having a low value of WR_Score is considered to be more helpful and ranked accordingly (a reviewer with lowest WR_Score is ranked 1 and so on).

$$WR\_Score = Rev\_Rank * R\_Score \tag{3}$$

This model [38] works fine when fewer or a limited number of reviewers are to be judged. Ranking or assigning a score to individual reviewers is meaningful for identifying them separately. However, in most of the online based systems the number of reviews and number of reviewers are huge in size. The individual ranking of reviewers will be very slow for continuously growing size of data as well as for the customers it will be difficult to understand the rating of individual reviewers and hence the importance of every review for a particular system. Rather it will be better for a customer to go through the review which has been written by authentic reviewers. Thus, instead of ranking individually, grouping the reviewers based on some score or other parameters would be more relevant and useful.

2.2. Proposed method for reviewer clustering

The proposed model presents a reviewer clustering method in the form of groups or clusters. Three types of clusters will be identified to find out the top-quality reviewers, medium quality reviewers, fake or spam reviewers. This requirement motivated us to propose a clustering framework to group the reviewers by analyzing the quality of the ratings given by the reviewers. The parameters that are given priority based on which the clustering has been performed are: product price and opinion variance. Product price is an interesting feature that speaks a lot about reviewers' characteristics. From a business analysis perspective also, high priced products demand to have more trusted reviews. For example, a good quality and high-priced product can be marked with poor reviews by group spammers with a deliberate intention to downgrade its market. This could mislead many customers and refrain them from buying the product, hence resulting in significant business losses. Opinion variance on the other hand is one such parameter which ensures the degree of authenticity of reviews written by reviewers. If a review rating given by a reviewer for a specific product match with the maximum numbers of other reviews for the same product, then it gets higher weightage. This is the core idea of the research work. In order to apply the analysis steps, the dataset needs to be cleaned in the pre-processing stage to get a dataset having significant values in all fields for the effective analysis.

2.2.1. Data pre-processing

The following steps are covered during data pre-processing stage:

- Removal of anonymous reviewers-users or reviewers having unknown reviewer ID and their records.
- Removal of products having unknown product price.
- Removal of duplicate records–this step is necessary since Amazon.com maintains duplicate products.

2.2.2. Data analysis

The entire data analysis is based on two different parameters: calculation of *Reviewer_Score* based on average opinion variance and *Price_Score* based on product price as shown in Figures 1 and 2 respectively. Figure 1 shows how the review count per reviewer and review count per product is used and combined with average opinion variance to compute the reviewer score. Figure 2 shows the steps in detail by means of which dynamic slab on product price is applied and finally the price score generated with respect to each reviewer in the dataset.

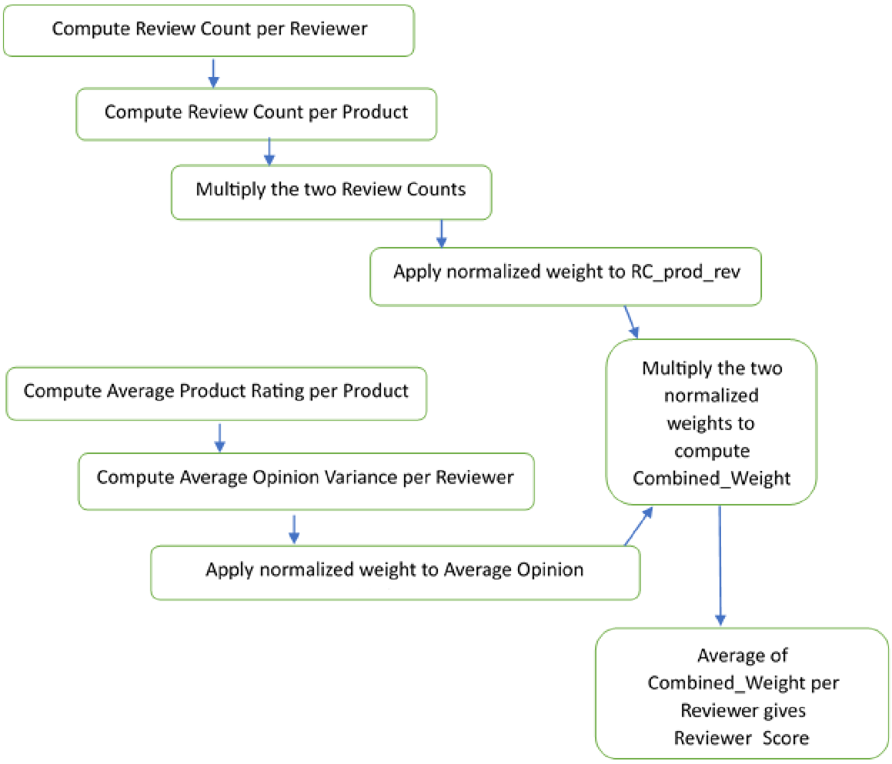
Compute Review Count per Reviewer
Compute Review Count per Product
Multiply the two Review Counts
Apply normalized weight to RC_prod_rev
Multiply the two normalized weights to compute Combined_Weight
Compute Average Product Rating per Product
Compute Average Opinion Variance per Reviewer
Apply normalized weight to Average Opinion
Average of Combined_Weight per Reviewer gives Reviewer Score

Figure 1. Workflow diagram for reviewer score calculation

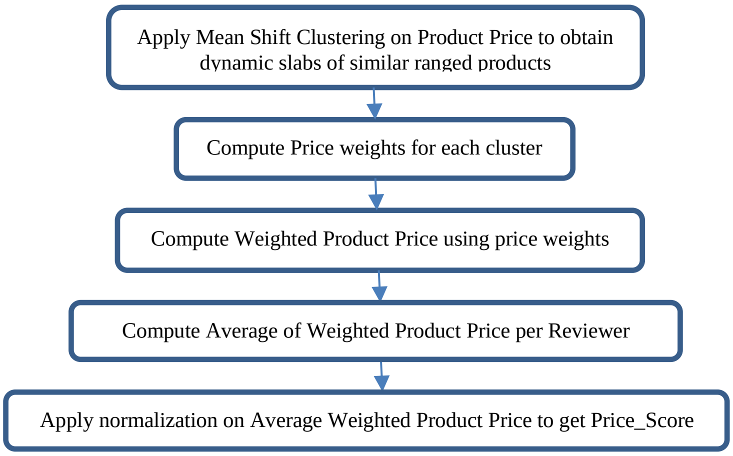
Apply Mean Shift Clustering on Product Price to obtain dynamic slabs of similar ranged products
Compute Price weights for each cluster
Compute Weighted Product Price using price weights
Compute Average of Weighted Product Price per Reviewer
Apply normalization on Average Weighted Product Price to get Price_Score

Figure 2. Workflow diagram for price score calculation

– Reviewer score generation

The average opinion variance calculated for every reviewer is the key parameter for Reviewer_Score calculation. It depends on the following two values majorly: i) review count per reviewer (RC_reviewer) and ii) review count per product (RC_product).

Records from the dataset having both products and reviewers with high value of review count get high priority/weightage as these records determine active reviewers reviewing popular products. We multiply both the counts to get a combined value.

A normalized weighting method is used to associate a priority or weightage value in the range of [0,1] against the RC_prod_rev value of each record. A high value of weight is applied to those records which have a high value of RC_prod_rev.

On the other hand, opinion variance denotes deviation in user rating and is calculated for each (reviewer, product) pair. But when we want to judge a particular reviewer on an integrated ground based on his/her general rating tendency then an average of all the opinion variance for each reviewed product is needed against that reviewer.

Reviewers having low average opinion variance are desirable as they signify a higher degree of trustworthiness. Consequently, they are assigned high normalized weight. But it should be noted here that this low average opinion variance is desirable only when the corresponding RC_prod_rev value is also high. The reason being, average opinion variance is calculated in such a way that these two counts highly determine the result. A product which is reviewed by many users, i.e., a popular product and a reviewer who has written a good number of reviews, i.e., an active reviewer is vital to our observation.

The combined weight against each (reviewer and product) pair, obtained after multiplying the normalized weight for RC_prod_rev with the normalized weight of average opinion variance incorporates the entire parameter dependency aspect of our analysis. The final Reviewer_Score for each reviewer is based on the analysis of average opinion variance.

Algorithm for reviewer score generation

```
1. For each reviewer $R_i$, i=1 to n:     [n is the number of reviewers in the dataset]
        RC_reviewer = Review count per reviewer
 [EndFor]
2. For each product $P_j$, j=1 to m:      [m is the number of products in the dataset]
        2.1 RC_product= Review count per product   [number of reviews for $P_j$]
        2.2 $Average\_Product\_Rating=( \sum_{k=1}^{RC\_product} Rating(Pj))/ RC\_product$
 [EndFor]
3. For each reviewer $R_i$, i=1 to n :    [n is the number of reviewers in the dataset]
        For each product $P_j$, j=1 to m: [m is the number of products reviewed by $R_i$]
                3.1 $X_{ij}$= Rating given by reviewer $R_i$ to product $P_j$
                3.2 $Opinion\_Variance_{i,j} = | X_{ij} – Average\_Product\_rating(P_j)$ [Rating Difference]
                3.3 $RC\_prod\_rev_{i,j}= RC\_reviewer_i * RC\_product_j$
                        [Both review counts multiplied to get a combined count]
        [EndFor]
 [EndFor]
```

```
4. Compute Weight_RC_Prod_Rev by:
        4.1 Sort RC_prod_rev in ascending order of their values.
            4.2 Assign Rank to RC_prod_rev, starting from 1 onwards giving high priority (rank) to high
             value.
            4.3 Normalize the ranks in the range [0,1] by dividing each value by the maximum value in the
             range.
5. For each reviewer $R_i$, i=1 to n:     [n is the number of reviewers in the dataset]
        5.1 m = $RC\_reviewer_i$      [m is the number of reviews written by $R_i$]
        5.2 $Average\_Opinion\_Variance_i = (\sum_{j=1}^{m} Opinion\_variance)/m$
[EndFor]
6. Compute Weight_Avg_Variance by:
        6.1 Sort Average_Opinion_Variance in descending order of their values.
        6.2 Assign Rank to Average_Opinion_Variance, starting from 1 onwards giving high priority (rank)
             to low value.
        6.3 Normalize the ranks in the range [0,1] by dividing each value by the maximum value in the
             range.
7. For each reviewer $R_i$, i=1 to n:     [n is the number of reviewers in the dataset]
        For each product $P_j$, j=1 to m: [m is the number of products in the dataset]
        $Combined\_Weight = (Weight\_RC\_Prod\_Rev) * (Weight\_Avg\_Variance)$
        [EndFor]
 [EndFor]
8. For each reviewer $R_i$, i=1 to n:     [n is the number of reviewers in the dataset]
        8.1 m = $RC\_reviewer_i$ [m is the number of reviews written by $R_i$]
        8.2 $Reviewer\_Score_i = (\sum_{j=1}^{m} Combined\_Weight)/m$
[EndFor]
```

– Price score generation

Price score generation emphasizes product price. Often from a business analysis perspective, customers tend to search for products within a certain price range based on their budget. This gives a fair idea about their buying pattern. Grouping customers based on the price of their reviewed products thus helps in better understanding of the business. The way we have assigned normalized weights to our parameters for reviewer score calculation is slightly different from the way we assign weight to product price. Hence, instead of treating product price individually, we have divided the entire dataset into dynamic slabs of similar priced products and then applied normalized ranking on the different slabs. This is needed as new products with new price ranges, if added to the dataset, should not bother the existing algorithm. Dynamic slabbing is preferred over fixed sized divisions as it is independent of the dataset distribution. We have used the Mean Shift Clustering algorithm for clustering products based on prices. This clustering algorithm is chosen as pre-specifying the number of desired clusters is not needed here. Ranking was applied to the resultant clusters to prioritize high priced products with high normalized weights. The normalization of product price weights is needed to limit the values in the scale of [0,1]. Then, this weight is used to generate weighted product prices (Weighted_PP) and the final Price_Score for each reviewer is calculated based on the analysis of this product price. Table 1 gives first few records from the dataset.

Algorithm for price score generation

```
1. Call Method MeanShift_Clustering_Algorithm(Product_Price)
                                        [function call to obtain product clusters of dynamic slabbing]
2. Compute Normalized_weight_PP by:
        2.1 Sort the resultant cluster in ascending order of their product price values.
        2.2 Assign Rank on the resultant clusters starting from 1 onwards giving high priority (rank) to high
             value.
        2.3 Normalize the ranks in the range [0,1] by dividing each value by the maximum value in the
             range.
3. For each product $P_i$, i=1 to n: [n is the number of products in the dataset]
                $Weighted\_PP = ProdPrice * Normalized\_weight\_PP$
 [EndFor]
4. For each reviewer $R_i$, i=1 to n:     [n is the number of reviewers in the dataset]
        4.1 m = $RC\_reviewer_i$ [m is the number of products reviewed by $R_i$]
        4.2 $Price\_Score_i = (\sum_{j=1}^{m} Weighted\_PP)/m$
[EndFor]
```

```
Method MeanShift_Clustering_Algorithm(Product_Price)
Input: A dataset D containing n objects of product ID and product price.
Output: A set of k clusters.

1. Start
2. Initialize estimate x.
3. $K(x-x_i) = e^{c||xi-x||}$,                    [K is a kernel function]
 The weighted mean of the density in the window determined by K is given by the formula:

    $m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$

        [N(x) is the neighbourhood of x, a set of points for which K(x)!= 0]
4. $x_i = m(x_i)$                    [Repeat Steps 3 and 4 till $m(x_i)$ converges]
5. End
```

Table 1. Product price clustering and normalized weighting

PID	ProdPrice	Cluster	Normalized weight_PP	Weighted_PP
B0001YLG44	325	9	1	325
B00020X3GG	325	9	1	325
B0001YLG5I	324.01	9	1	324.01
B0008EZETK	286.85	8	0.9	258.165
B000P91P1E	286.24	8	0.9	257.616
B0007YUOAK	263.24	7	0.8	210.592

– Clustering

The main objective of the research work is to identify a group of reviewers with a certain level of trustworthiness with respect to their product rating pattern. Additionally, product price is also included to check how it impacts the rating tendency of the reviewer.

Now, the actual objective or purpose of our research work is to group reviewers based on their two generated scores. For this reason, a clustering algorithm is applied on the prepared Reviewer sheet containing Reviewer ID, Reviewer_Score and Price_Score. Few reviewers of the Reviewer sheet sorted by Reviewer ID have been presented in Table 2.

Table 2. Reviewer sheet

Reviewer ID	Reviewer_Score	Price_Score
A0009060FA8P413511WS	0.148799958	0.000769231
A005978815H13HB90PP3D	0.284820627	0.006367692
A01741982OW89WE77YKAJ	0.407931063	0.007692308
A01873002E9N4RUV4EW0E	0.025678264	0.009215385
A0238875Y5SLPW18T91C	0.062518837	0.002089231
A02755422E9NI29TCQ5W3	0.169885296	0.000870769

In this paper, two clustering algorithms namely: i) K-means and ii) Jenks natural breaks optimization method (JNBO) are used. At first, the K-means clustering algorithm is used to perform clustering on reviewers based on both Reviewer_Score and Price_Score whereas the second method is applied on the dataset where the clustering is done only on the Reviewer_Score (one-dimensional clustering). The clustering of reviewers on the same dataset is done twice based on different parameters to find the impact of product price (if any) on reviewer rating pattern. This study aims to check whether the result of reviewer clusters obtained after applying K-means on both Reviewer_Score and Price_Score differs with that of the result obtained after applying JNBO with Reviewer_Score only.

Cluster analysis is usually a multivariate technique. Applying k-means on one dimensional data is not meaningful, unless we put in enough effort to optimize them for 1-D data. The JNBO method is a data-clustering method designed to predominantly work on 1-D data. It is generally used to determine the best partition of values into different classes. JNBO achieves its goal by trying to:

– Minimize each class' average deviation from the class mean;

- Maximize each class' deviation from the mean of the other classes.

In other words, the method seeks to reduce the variance within classes and maximize the variance between classes. JNBO tries to optimize the cluster borders. This ensures that each point will be allocated to the most appropriate class. K-means tests each object to see if it belongs to its current class or not, which is inapplicable for 1-D data, since it is only the points at the border of the interval that needs to be checked. This is where JNBO is faster than K-means on 1-D data.

```
Method K-means_Clustering_Algorithm(Reviewer_Score, Price_Score)
Input:
    k: The number of clusters.
    D: A dataset containing n objects of Reviewer ID, Reviewer_Score and Price_Score.
Output: A set of k clusters.

1. Start
2. Arbitrarily choose k objects from D as the initial cluster centers.
        [each cluster's centre is represented by the mean value of the objects in the cluster]
3. Repeat until no change:
            3.1 (Re)assign each object to the cluster to which the object is most similar, based on the mean
            value of the objects in the cluster.
            3.2 Update the cluster means.
                        [calculate the mean value of the objects for each cluster]
 [EndLoop]
4. End
```

```
Method Jenks_Natural_Break_Optimization(Reviewer_Score)
Input:
    bins: The number of clusters/bins
    D: A dataset containing n objects of Reviewer_Score.
Output: A set of k clusters.

1. Start
2. Arbitrarily divide the ordered data into k classes.
3. Repeat until the sum of within-class deviation reaches the minimum value:
            3.1 Calculate the sum of squared deviations from the class means (SDCM);
            3.2 Redistribute the data into the classes based upon the newly calculated class deviations (by
            moving data-points from one class to another)
 [EndLoop]
4. End
```

Based on this proposed method, a case study is demonstrated in the following section.

3. RESULTS AND DISCUSSION

Dataset description–we have used the Amazon Fashion dataset from Amazon Review Data (2018) (collected from Github [39]). The dataset initially had fields namely: product ID, product title, product price, Reviewer/User ID, review/profile name, review helpfulness, review score, review time, review summary and review text. We have chosen four fields for our project work namely: Product ID, Reviewer/User ID, Product Price, and Review Score/Rating.

After data pre-processing, we worked with the dataset having 19,130 records. This work of reviewer clustering is targeted to cluster overall 14,848 reviewers based on Product price of 2,271 products and average opinion variance.

3.1. Product price dynamic clusters obtained after using mean shift clustering algorithm

The result depicted in Table 3 shows the product clusters dynamically slabbed based on their prices. After applying the Mean Shift Clustering Algorithm, we obtained 10 clusters of product prices. The result sheet mentioned in the table shows the cluster numbers (9 to 0) sorted in descending order of product price values. We have also shown the range of product price belonging to each cluster in the same table.

Table 3. Product cluster distribution based on price

Product price	Cluster number
325 to 324.01	9
286.85 to 286.24	8
263.24 to 255.77	7
234.12	6
206.84 to 174.76	5
159.95 to 139.74	4
135.84 to 113.4	3
109.99 to 86	2
84.95 to 44.95	1
44.31 to 0.32	0

3.2. Cluster of reviewers based on both reviewer score and price score using K-means clustering algorithm

The result shown in Table 4 displays some of the reviewers grouped based on the reviewer score calculated over average opinion variance and price score computed over product price. We have applied the K-means clustering algorithm on our dataset having cluster size equal to 3. The fourth column of the result sheet shows the cluster numbers (0, 1, and 2) specifying the corresponding cluster assignments to the reviewers. Here we have shown only a few reviewers out of 14,848 reviewers due to space limitations. However, Table 5 shows the cluster distribution and reviewer categories marked as good, average, and less significant reviewers according to the reviewer score range similar to the last clustering. The graph plot of the output shown in Figure 3, is obtained after clustering where the cluster centers are highlighted with circles. The two axes of the graph plot stand each for the price score (PScore) and reviewer score (RScore) both scaled in the range [0,1].

Table 4. Reviewer cluster allocation based on reviewer score and price score

RID	PScore	RScore	Cluster
A3IR834T7AROBT	0.097458462	0.064668975	0
A2Y2AZD36V9USQ	0.007676923	0.025232068	0
A23ZZL3C7NCBDD	0.005538462	0.540198985	2
A3RQNIRFKEGHVR	0.010581538	0.190275946	0
A28QX9NZL2O3AF	0.030738462	0.292336399	1
A1I08AQQAG9TT5	0.005532308	0.450684857	1

Table 5. Reviewer cluster distribution based on reviewer score and price score

Reviewer score	Cluster size	Cluster number	Reviewer category
0.964459049 to 0.501726539	1933 (~13%)	2	Good
0.497563427 to 0.207437713	4190 (~28.22%)	1	Average
0.206630686 to 0.000052274	8725 (~59.76%)	0	Less significant

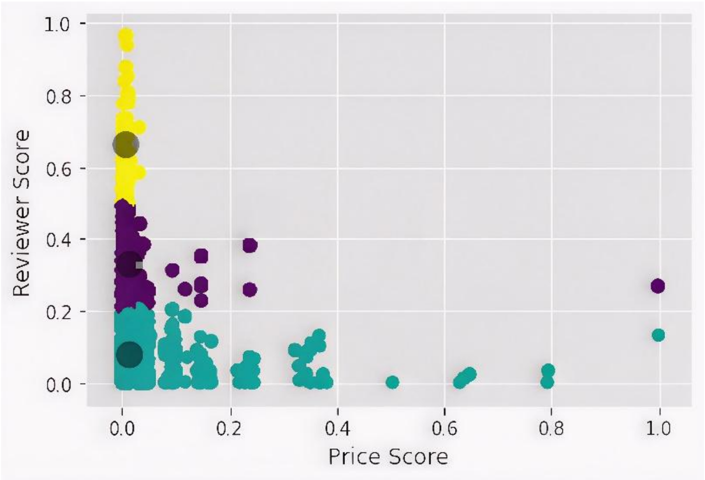

Figure 3. Graph plot of reviewer clusters based on reviewer score and price score

3.3. Clusters of reviewers based on only reviewer score using Jenks natural breaks optimization method

The result illustrated in Table 6 shows some of the reviewers grouped based on the reviewer score only where the score is generated using average opinion variance. We have applied Jenks natural breaks classification method (or JNBO method) on our dataset, with the number of classes set to 3. The third column of the result sheet shows the cluster numbers (0, 1, and 2) specifying the corresponding class/interval assignments to the reviewers. Here we have shown only a few reviewers due to space limitation. However, Table 7 shows the class distribution and reviewer categories marked as good, average and less significant reviewers according to the reviewer score range. The range or class having high reviewer scores are marked as good reviewers, whereas the low value of reviewer score denotes the less significant reviewers. The values lying between these two groups are for the average reviewers. The graph plot of the output shown in Figure 4, is obtained after dividing the data points into different intervals after applying the JNBO method. Reviewer number acts as the horizontal axis of the graph plot, whereas reviewer score is the vertical axis.

Table 6. Reviewer cluster allocation based on reviewer score

Reviewer ID	Reviewer score	Cluster
A0009060FA8P413511WS	0.148799958	0
A005978815H13HB90PP3D	0.284820627	1
A01741982OW89WE77YKAJ	0.407931063	1
A01873002E9N4RUV4EW0E	0.025678264	0
A0238875Y5SLPWI8T91C	0.062518837	0

Table 7. Reviewer cluster distribution based on reviewer score

Reviewer score	Cluster size	Cluster number	Reviewer category
5.2e-05 to 0.206631	8725 (58.76%)	0	Less significant
0.206631 to 0.497563	4190 (28.22%)	1	Average
0.497563 to 0.964459	1933 (13.02%)	2	Good

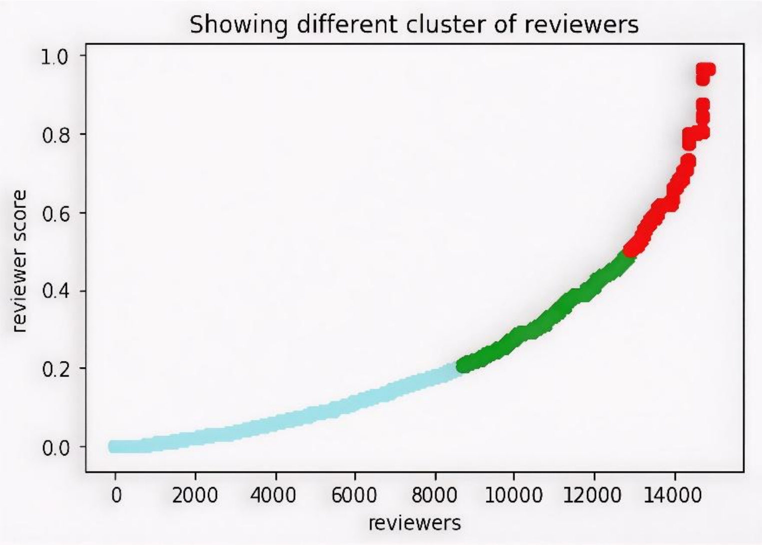

Figure 4. Graph plot of reviewer clusters based on reviewer score

3.4. Validation of result

In this study, we aim to evaluate whether the inclusion of an additional feature (PScore) significantly influence the quality of clustering results. Or in other words, the analysis tries to find whether higher-priced products tend to attract more polarized reviews, making product price a significant factor in assessing reviewer reliability.

A t-test is a statistical test used to compare the means of two groups to see if they are significantly different from each other. Here, we have two clustering results from almost identical datasets (one extra feature difference), thus we used t-test to check whether adding that extra feature significantly changes the clustering quality. In order to achieve this, we computed silhouette scores for clustering solutions obtained with and without the extra feature (PScore) across different sampling fractions of the dataset

(F=[0.1, 0.2, 0.25, 0.33, 0.5, 0.75, 1]). For each fraction, clustering was repeated 30 times to justify randomness, and the average silhouette scores were compared using *t*-test. Across all fractions, the model with the additional feature consistently achieved lower average silhouette scores (≈0.60) compared to the reduced feature set (≈0.62). The *p*-values in all cases were well below 0.05, indicating statistically significant differences. It is observed from the analysis shown in Table 8 for every fraction of the dataset, removing PScore consistently improves the silhouette score, and the improvement is statistically significant. As the inclusion of price score reduces the clustering quality it is not being considered as a controlling feature for clustering reviewers.

Table 8. Validation of result using t-test

Fraction (f)	Avg silhouette (with PScore)	Avg silhouette (without PScore)	p-value	Significance
0.10	0.6039	0.6234	7.07e-12	Significant
0.20	0.6015	0.6234	6.58e-18	Significant
0.25	0.6018	0.6212	7.55e-20	Significant
0.33	0.6022	0.6231	4.83e-27	Significant
0.50	0.6054	0.6230	7.76e-15	Significant
0.75	0.6035	0.6225	4.78e-42	Significant
1.00	0.6037	0.6225	7.26e-77	Significant

3.5. Comparative analysis of different methods of reviewer clustering

Reviewer clustering is a popular branch of study of online review spammer group detection. In addition, this research work introduces a new concept of analysis compared to the existing work as it considers impact of the product price on reviewer rating pattern. Existing literature worked with the reviewer rating pattern only. Though the proposed method cannot be directly compared with the existing methodologies due to the consideration of the cost of the product, the key parameters of this research is discussed and a comparison study is presented in Table 9 against some following factors:

- Activeness of reviewer–considering count of reviews posted by a particular reviewer to determine his/her activeness in reviewing.
- Popular product–counting number of reviews for a particular product to identify how frequently the product is purchased and reviewed.
- Product price–looking for any relationship between review quality and price of product.

After studying the existing literature work for determining online reviewer group spammers, it is found that researchers have highlighted on various aspects or features for clustering reviewers. However, no analysis is so far done which puts any light on the impact of the feature '*product price*' on reviewer rating pattern. A reviewer when gives feedback or rates a product or service, not only the product/service quality is evaluated, but equally valuation is made whether the purchase was worth the price. This is where lies the novelty of our paper which considers a new aspect that relates the quality of review with cost of the product.

Table 9. A comparison chart of reviewer clustering based on three factors

References	Consideration of activeness of reviewer	Consideration of rating of popular product	Identifying relationship between review quality and product price
[26]	×	×	×
[25]	✓	×	×
[27]	✓	×	×
[35]	×	×	×
[33]	×	✓	×
[9]	✓	✓	×
[32]	✓	✓	×
Proposed model	✓	✓	✓

4. CONCLUSION

This research work is focused to help the people who are searching the review of products in an online system. There are many reviews available but quality of all the reviews are not same. Before purchasing any product, if buyers get recommendations by a good set of reviewers, then they can be benefited. Considering these factors, the proposed framework is customized for any online system to segment the reviewers based on the review rating they gave to the different products. The data has been filtered based on the activeness of reviewers, popularity of product to capture the current scenario of the market for finetuned actions. The most significant aspect of this research work is to investigate the relationship between quality of review and product price. The in-depth analysis based on t-test reveals while customers are writing review, the product price does not impact the review quality. This signifies every organization needs to focus on the quality of the product irrespective of price. Otherwise, customers will have negative approach about the organization which will impact the sales of other products. The novelty of the work lies in the analysis of reviewer score along with price score to categorize reviewers and understand reviewer behaviour in online platforms. The proposed framework is flexible to be applied to diverse business domains like education system, travel and tourism industry, medical systems, and customer relationship management (CRM). where people's feedback and ratings play a vital role. It can also be used for spam reviewer detection and over the time improving recommendation systems. Due to infrastructural challenge, we could not work with large datasets which turned out to be one of the limitations of our work.

The future extension of this work includes analysis including attributes like review content, helpfulness votes, and reviewer characteristics. This work considers numeric contents only however most of the systems accept reviews in textual format. Henceforth natural language processing (NLP) based methods will be suitable to analyze the exact sentiment of the customer. It is often found that there is a gap between the given review score and the text message written by the reviewers. Sentiment analysis using NLP based methods could be applied to resolve these types of problems. Another interesting extension of this work is to identify trusted and untrusted reviewers and frame this problem as a typical binary classification problem. Moreover, this analysis could be applied on specific product and consider diversified business domains to explore new business challenges.

FUNDING INFORMATION

This work was supported by Toyo University, Japan.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Runa Ganguli	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			
Akash Mehta	✓	✓	✓	✓	✓	✓		✓	✓		✓			
Takaaki Goto	✓				✓					✓		✓	✓	✓
Soumya Sen	✓	✓		✓	✓		✓		✓	✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created and the used dataset link is provided in the paper.

REFERENCES

[1] R. Misra, R. Mahajan, N. Singh, S. Khorana, and N. P. Rana, "Factors impacting behavioural intentions to adopt the electronic marketplace: findings from small businesses in India," *Electron. Mark.*, vol. 32, no. 3, pp. 1639–1660, Sep. 2022, doi: 10.1007/s12525-022-00578-4.

[2] H. Dhumras, P. K. Shukla, R. K. Bajaj, D. K. Jain, V. Shukla, and P. K. Shukla, "On Federated Learning-Oriented q -Rung Picture Fuzzy TOPSIS/VIKOR Decision-Making Approach in Electronic Marketing Strategic Plans," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2557–2565, Feb. 2024, doi: 10.1109/TCE.2023.3325434.

[3] Z. Khan, M. I. Hussain, N. Iltaf, J. Kim, and M. Jeon, "Contextual recommender system for E-commerce applications," *Appl. Soft Comput.*, vol. 109, p. 107552, Sep. 2021, doi: 10.1016/j.asoc.2021.107552.

[4] R. Mohawesh *et al.*, “Fake Reviews Detection: A Survey,” *IEEE Access*, vol. 9, pp. 65771–65802, 2021, doi: 10.1109/ACCESS.2021.3075573.

[5] J. Salminen, C. Kandpal, A. M. Kamel, S. Jung, and B. J. Jansen, “Creating and detecting fake reviews of online products,” *J. Retail. Consum. Serv.*, vol. 64, p. 102771, Jan. 2022, doi: 10.1016/j.jretconser.2021.102771.

[6] W. Yu, L. Liu, X. Wang, O. Bagdasar, and J. Panneerselvam, “Modeling and Analyzing Logic Vulnerabilities of E-Commerce Systems at the Design Phase,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 53, no. 12, pp. 7719–7731, Dec. 2023, doi: 10.1109/TSMC.2023.3299605.

[7] G. Rong *et al.*, “Distilling Quality Enhancing Comments From Code Reviews to Underpin Reviewer Recommendation,” *IEEE Trans. Softw. Eng.*, vol. 50, no. 7, pp. 1658–1674, Jul. 2024, doi: 10.1109/TSE.2024.3356819.

[8] D. Gutt, J. Neumann, S. Zimmermann, D. Kundisch, and J. Chen, “Design of review systems – A strategic instrument to shape online reviewing behavior and economic outcomes,” *J. Strateg. Inf. Syst.*, vol. 28, no. 2, pp. 104–117, Jun. 2019, doi: 10.1016/j.jsis.2019.01.004.

[9] R. K. Chenoori and R. Kavuri, “GrFrauder: A Novel Unsupervised Clustering Algorithm for Identification Group Spam Reviewers,” *Ingénierie des systèmes d Inf.*, vol. 27, no. 6, pp. 1019–1027, Dec. 2022, doi: 10.18280/isi.270619.

[10] Q. Zhang, Z. Liang, S. Ji, B. Xing, and D. K. W. Chiu, “Detecting fake reviewers in heterogeneous networks of buyers and sellers: a collaborative training-based spammer group algorithm,” *Cybersecurity*, vol. 6, no. 1, p. 26, Oct. 2023, doi: 10.1186/s42400-023-00159-8.

[11] D. Zhang, W. Li, B. Niu, and C. Wu, “A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information,” *Decis. Support Syst.*, vol. 166, p. 113911, Mar. 2023, doi: 10.1016/j.dss.2022.113911.

[12] A. Da’u and N. Salim, “Recommendation system based on deep learning methods: a systematic review and new directions,” *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.

[13] Z. Wang, A. Maalla, and M. Liang, “Research on E-Commerce Personalized Recommendation System based on Big Data Technology,” in *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, IEEE, Dec. 2021, pp. 909–913, doi: 10.1109/ICIBA52610.2021.9687955.

[14] I. Portugal, P. Alencar, and D. Cowan, “The use of machine learning algorithms in recommender systems: A systematic review,” *Expert Syst. Appl.*, vol. 97, pp. 205–227, May 2018, doi: 10.1016/j.eswa.2017.12.020.

[15] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, “A systematic review: machine learning based recommendation systems for e-learning,” *Educ. Inf. Technol.*, vol. 25, no. 4, pp. 2635–2664, Jul. 2020, doi: 10.1007/s10639-019-10063-9.

[16] M. Loukili, F. Messaoudi, and M. El Ghazi, “Machine learning based recommender system for e-commerce,” *IAES Int. J. Artif. Intell.*, vol. 12, no. 4, pp. 1803–1811, Dec. 2023, doi: 10.11591/ijai.v12.i4.pp1803-1811.

[17] Y. Xu and F. Zhang, “Detecting shilling attacks in social recommender systems based on time series analysis and trust features,” *Knowledge-Based Syst.*, vol. 178, pp. 25–47, Aug. 2019, doi: 10.1016/j.knosys.2019.04.012.

[18] Y. Wang, W. Zuo, and Y. Wang, “Research on Opinion Spam Detection by Time Series Anomaly Detection,” *Int. Conf. Artif. Intell. Secur.*, 2019, pp. 182–193, doi: 10.1007/978-3-030-24274-9_16.

[19] H. Li *et al.*, “Bimodal Distribution and Co-Bursting in Review Spam Detection,” in *Proceedings of the 26th International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 1063–1072, doi: 10.1145/3038912.3052582.

[20] A. M. Shetty, M. F. Aljunid, and D. H. Manjaiah, “Unleashing the power of 2D CNN with attention and pre-trained embeddings for enhanced online review analysis,” *Int. J. Comput. Appl.*, vol. 46, no. 1, pp. 46–57, Jan. 2024, doi: 10.1080/1206212X.2023.2283647.

[21] L. Zheng, L. Sun, Z. He, and S. He, “Dynamic product competitive analysis based on online reviews,” *Decis. Support Syst.*, vol. 183, p. 114268, Aug. 2024, doi: 10.1016/j.dss.2024.114268.

[22] L. Chen, R. Xiong, and Y. Ji, “Application of SVM model based on collaborative filtering hybrid algorithm in e-commerce recommendation,” *Int. J. Comput. Appl.*, vol. 46, no. 5, pp. 292–300, May 2024, doi: 10.1080/1206212X.2024.2309809.

[23] Y. Xiao, W. Zhao, Y. Huang, T. Li, and Q. Li, “A Joint Learning Recommendation Model for E-Commerce Platforms Integrating Long-Term and Short-Term Interests,” *IEEE Trans. Serv. Comput.*, vol. 17, no. 4, pp. 1326–1339, Jul. 2024, doi: 10.1109/TSC.2024.3376232.

[24] 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. 53801–53816, 2020, doi: 10.1109/ACCESS.2020.2979226.

[25] M. Zhong, L. Tan, and X. Qu, “Identification of Opinion Spammers using Reviewer Reputation and Clustering Analysis,” *Int. J. Comput. Commun. Control*, vol. 14, no. 6, p. 759, Feb. 2020, doi: 10.15837/ijccc.2019.6.3704.

[26] P. Saini, S. Shringi, N. Sharma, and H. Sharma, “Spam Review Detection Using K-Means Artificial Bee Colony,” *Lecture Notes in Netw. and Sys.*, 2021, pp. 731–744, doi: 10.1007/978-981-16-1089-9_57.

[27] V. Gupta, A. Aggarwal, and T. Chakraborty, “Detecting and Characterizing Extremist Reviewer Groups in Online Product Reviews,” *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 3, pp. 741–750, Jun. 2020, doi: 10.1109/TCSS.2020.2988098.

[28] T. Bai, W. X. Zhao, Y. He, J.-Y. Nie, and J.-R. Wen, “Characterizing and Predicting Early Reviewers for Effective Product Marketing on E-Commerce Websites,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2271–2284, Dec. 2018, doi: 10.1109/TKDE.2018.2821671.

[29] Z. Xing and W. Zhao, “Block-Diagonal Guided DBSCAN Clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 5709–5722, Nov. 2024, doi: 10.1109/TKDE.2024.3401075.

[30] Z. Wang, R. Hu, Q. Chen, P. Gao, and X. Xu, “ColluEagle: collusive review spammer detection using Markov random fields,” *Data Min. Knowl. Discov.*, vol. 34, no. 6, pp. 1621–1641, Nov. 2020, doi: 10.1007/s10618-020-00693-w.

[31] Z. Zhang, M. Zhou, J. Wan, K. Lu, G. Chen, and H. Liao, “Spammer detection via ranking aggregation of group behavior,” *Expert Syst. Appl.*, vol. 216, p. 119454, Apr. 2023, doi: 10.1016/j.eswa.2022.119454.

[32] G. Xu, M. Hu, C. Ma, and M. Daneshmand, “GSCPM: CPM-Based Group Spamming Detection in Online Product Reviews,” in *ICC 2019 - 2019 IEEE Intern. Conf. on Commun.(ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761650.

[33] C. Cao, S. Li, S. Yu, and Z. Chen, “Fake Reviewer Group Detection in Online Review Systems,” in *2021 Int. Conf. on Data Mining Workshops (ICDMW)*, IEEE, Dec. 2021, pp. 935–942, doi: 10.1109/ICDMW53433.2021.00122.

[34] J. Chen, B. Li, and K. He, “Neighborhood convolutional graph neural network,” *Knowledge-Based Syst.*, vol. 295, p. 111861, Jul. 2024, doi: 10.1016/j.knosys.2024.111861.

[35] P. Rathore, J. Soni, N. Prabakar, M. Palaniswami, and P. Santi, “Identifying Groups of Fake Reviewers Using a Semisupervised Approach,” *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 6, pp. 1369–1378, Dec. 2021, doi: 10.1109/TCSS.2021.3085406.

[36] A. P. Sundar, F. Lilt, X. Zou, and T. Gao, “DeepDynamic Clustering of Spam Reviewers using Behavior-Anomaly-based Graph Embedding,” in *GLOBECOM 2020 - 2020 IEEE Global Communic. Conf.*, Dec. 2020, pp. 01–06, doi:

10.1109/GLOBECOM42002.2020.9322330.

[37] C. Wang, N. Li, S. Ji, X. Fang, and Z. Wang, “Enhancing fairness of trading environment: discovering overlapping spammer groups with dynamic co-review graph optimization,” *Cybersec.*, vol. 7, no. 1, pp. 1-28, Jun. 2024, doi: 10.1186/s42400-024-00230-y.

[38] R. Ganguli, P. Banerjee, S. Halder, and S. Sen, “A Mathematical Recommendation Model to Rank Reviewers Based on Weighted Score for Online Review System,” *Emerging Techn. in Data Mining and Inform.Security: Procee. of IEMIS 2020*, 2021, pp. 317–325, doi: 10.1007/978-981-15-9774-9_31.

[39] J. Ni, J. Li, and J. McAuley, “Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects,” in *Procee. of the 2019 Conf. on Empirical Methods in Natural Languag. Proc. and the 9th Intern. Joint Conf. on Nat. Lang. Process. (EMNLP-IJCNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 188–197.

BIOGRAPHIES OF AUTHORS

Ms. Runa Ganguli completed her Master of Technology in Computer Engineering and Applications from A.K. Chowdhury School of IT, University of Calcutta, India in 2020. She received her Master of Science degree in computer and information science from the University of Calcutta, India, in 2014. She is currently working as Assistant Professor in the Department of 1st year of 4-year B.Tech., University of Calcutta, India. She has several papers in reputed peer reviewed journals and international conferences. Her main research interest includes graph database, data mining, and recommendation systems. She has 10 years of teaching experience. She can be contacted at email: runa.ganguli@gmail.com.

Mr. Akash Mehta received his master’s degree in computer and information science from the University of Calcutta, Kolkata, India, in 2015. He received his master’s degree in computer science and engineering from the University of Calcutta, Kolkata, India, in 2017. He is currently working towards the Ph.D. degree with the Department of Computer Science and Engineering, University of Calcutta. He is working on the applications of computer vision for detecting behaviors of risk in people with Alzheimer’s disease. He can be contacted at email: akash.researchphd.mehta@gmail.com.

Dr. Takaaki Goto graduated with a Doctor of Engineering from Toyo University in 2009. He had been a Project Assistant Professor at the University of Electro-Communications in Japan from 2009 to 2015 and he joined the Ryutsu Keizai University in Japan in 2015 as a lecturer and served as an associate professor from 2016 to 2019. Now he is an Associate Professor at Toyo University, Japan. His main research interests are software engineering, IoT, and AI applications. He is a member of ACM, IEEE, ACIS, ISCA, IEICE, and IPSJ. He can be contacted at email: tg@gotolab.net.

Dr. Soumya Sen is an assistant professor in A.K.Choudhury School of Information Technology under University of Calcutta since 2009. He has around 135 research publications in International Journals and conferences. He has 3 international patents, 3 copyrights and also published 2 books. He is a member of IEEE, ACM and life member of Society for Data Science (S4DS). He is a fellow of IETE (Institution of Electronics and Telecommunication Engineers). His current research area interests are data warehouse and OLAP tools, machine learning, and data science. He can be contacted at email: iamsoumyasen@gmail.com.