Research Article

Akanksha Bansal Chopra* and Veer Sain Dixit*

Detecting biased user-product ratings for online products using opinion mining

https://doi.org/10.1515/jisys-2022-9030
received January 11, 2022; accepted October 09, 2022

Abstract: Collaborative filtering recommender system (CFRS) plays a vital role in today's e-commerce industry. CFRSs collect ratings from the users and predict recommendations for the targeted product. Conventionally, CFRS uses the user-product ratings to make recommendations. Often these user-product ratings are biased. The higher ratings are called push ratings (PRs) and the lower ratings are called nuke ratings (NRs). PRs and NRs are injected by factitious users with an intention either to aggravate or degrade the recommendations of a product. Hence, it is necessary to investigate PRs or NRs and discard them. In this work, opinion mining approach is applied on textual reviews that are given by users for a product to detect the PRs and NRs. The work also examines the effect of PRs and NRs on the performance of CFRS by evaluating various measures such as precision, recall, *F*-measure and accuracy.

Keywords: collaborative filtering recommender system, push ratings, nuke ratings, opinion mining

1 Introduction

E-commerce websites such as Amazon, Flipkart, etc., uses decision support systems, commonly known as recommender systems (RSs). These systems help the users to make decisions while purchasing a product. Majority of e-commerce websites use collaborative filtering approach. As RSs are open in nature, they face a major challenge of biased ratings [1]. These biased ratings are injected by malicious users. These biased ratings are required to be identified as these ratings alter the recommendations of the system resulting in false predictions of products to its users. The biased ratings are of two types, push ratings (PRs) and nuke ratings (NRs). These biased ratings may also be termed as push attack or nuke attack. Injection of PRs results in overestimation in prediction, while NRs result in an under prediction of a product. These PRs and NRs contribute to false predictions and affect the performance of the collaborative filtering recommender system (CFRS). In this work, the opinion mining technique is used to identify these PRs and NRs in CFRS.

The opinion mining analyses the textual information given by users to decide whether the given review is positive, negative or neutral [2]. Opinion mining mines and examines the users' opinion for a product [3]. The working of this technique involves five basic steps: Tokenization: A statement is divided into various sub-statements. For instance, the statement "The food is yummy!" is divided into sub-statements as "The," "food," "is," "yummy" and "!." Data cleaning: Removal of special characters is done. Sub-statement "!" would be removed in this step. Stop words removal: Stop words like the, is, was, he, she, they, etc., that do not contribute to the review are removed. The words "the" and "is" are removed from the given instance. Classification: Supervised algorithm is applied to the remaining sub-statements. A sentiment score of "+1"

* **Corresponding author: Akanksha Bansal Chopra,** Shyama Prasad Mukherji College for Women, Punjabi Bagh, New Delhi 110026, India, e-mail: akankshabansal.asm@gmail.com
* **Corresponding author: Veer Sain Dixit,** Atma Ram Sanatan Dharma College, University of Delhi, Dhaula Kuan, New Delhi 110021, India, e-mail: veersaindixit@rediffmail.com

for positive, “–1” for negative and “0” for neutral sub-statements is assigned and the model is trained with bag of words. As in the example, “0” to “restaurant” and “+1” to “awesome” are assigned as sentiment score. Calculating polarity and subjectivity: Polarity and subjectivity are calculated using descriptive statistics. Polarity is a metric in opinion mining that evaluates the amount of positive, negative or neutral emotions that appear in a given text. The value of polarity lies within the range of –1 to +1. Value approaching toward –1 implies more negative emotion; value approaching toward +1 implies more positive emotion and value 0 implies neutral emotion. Subjectivity whereas refers to how meaningful is the statement. Subjectivity is the metric that evaluates the amount of meaningfulness in a given text. The value of subjectivity lies within the range of 0 to +1. A value approaching 0 implies lesser meaning in the text, while a value approaching +1 implies text is meaningful. In this work, to evaluate polarity and subjectivity, text blob feature of python is used.

1.1 Problem definition and contribution

CFRS collects user-product ratings to generate recommendations. Often these ratings are injected by factitious users. The factitious users inject biased ratings and alter the recommendations. The purpose to alter the recommendation list is either to promote the sales of the target product or to demote the sales of the product from the competitor company. Users are highly dependent on the recommendation list to select the target product. It is therefore very necessary to detect these biased ratings and remove them from the list. This work collects written reviews for a product given by its users. On the given written reviews, opinion mining is applied to verify whether the given rating is genuine or being pushed or nuked. PR or NR, if found, is discarded and only valid ratings are stored in the database to generate the recommendations.

The analysis of performance of the RS is done in four possible ways – actual dataset including both PRs and NRs; actual dataset excluding PRs but inclusive of NRs; actual dataset excluding NRs but inclusive of PRs; and actual dataset excluding both PRS and NRs. Various metrics such as precision, recall, *F*-measure and accuracy are evaluated to perform the analysis.

PRs (or NRs) inserted into CFRS results in wrong recommendations of the products. These ratings affect the performance of the CFRS either by over-estimating or by under-estimating the system’s accuracy in generating the recommendations. The study, hence, aims to identify these two ratings, PRs and NRs, and analyze the effect on the performance of CFRS with or without the involvement of the biased ratings.

1.2 Structure of this study

This study is structured in five sections. Section 2 discusses the literature review. Section 3 explains various measures that are used in the study. The experimental evaluation and results are given in Sections 4 and 5, respectively. Finally, Section 6 gives the conclusion and outlook for future work.

2 Literature review

The performance of RSs account for the accuracy of the generated recommendations for a product [4]. Various authors have proposed various user-based and item-based algorithms [5] to improve the performance of RSs. *F*-measure metric is one of the technique that evaluates the quality of RS [6]. The performance can be evaluated by using user satisfaction levels by including items under multiple topics in the system [7]. A highly accurate RS can be achieved by protecting the users’ private data while making the recommendations [8]. A system sometimes exploits the information that has been given by the users as their feedbacks and provides multi-faceted representation of users' interest [9]. The users login behavior at users' location preference model can be used to enhance the personalized recommendation [10]. Another technique to improve the performance of RS is to run a similarity model on the local information of users [11]. In this regard, entropy and similarity measures can be used to achieve the effective results [12,13].

Opinion mining, also known as sentiment analysis, is another approach that has been used by researchers to improve the performance of the RSs. The user ratings and user sentiments (or reviews) can be combined to generate the recommendations for CFRS [14–19] and for content-based RS [20,21]. Sentiment analysis metrics such as Sentimeter-Br2 [22] and polarity [23] use the emotional state of the person and determine the enhanced performance of the CFRS.

Besides improving the performance of RSs, the researchers have thrown light on the weaknesses [24], limitations [25] and problems such as cold start [26,27] of RSs also. The malicious users take advantage of such weaknesses and attack the RS by injecting fake profiles. The influence analysis [28] can be used to detect these attacks. Other techniques such as classification approach and unsupervised clustering can be used to detect these attacks [29,30].

It may be concluded that there is no agreement between the existing academic literature about RS approaches. Some approaches are used to enhance the performance of the RSs, while others are used to detect the attacks. In our research, we have bridged this research gap by integrating the sentiment analysis approach with collaborative recommender system. The sentiments of the users are used to detect the attacks and then the performance of the system is analyzed.

3 Measures used

RSs have proved as a successful tool in ensuring user satisfaction all the time. Recommendations are generated by finding the similarity between user-product ratings. The user-product ratings with high degree of similarity are clustered using a neighborhood algorithm. Finally, the recommendations are generated for product I by finding the average values of nearest neighbors. Mathematically, degree of similarity between user-product ratings can be calculated by using Pearson's correlation coefficient.

$$r_{u_i,j} = \frac{N\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{i,k}u_{j,k} - (\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{i,k})(\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{j,k})}{[N\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{i,k}^{2} - (\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{i,k})^{2}][N\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{j,k}^{2} - (\sum_{i\neq j=1}^{n}\sum_{k=1}^{N}u_{j,k})^{2}]}.$$

In recommendation systems, it is desired to recommend only the top *n* products to the user. To achieve this, precision and recall metrics are computed by considering the topmost *N* results. In addition to this metric, accuracy is also determined by using confusion matrix. In notion, they are defined as follows:

$$\text{Precision} = \frac{t_p}{t_p + f_p};\ \text{Recall} = \frac{t_p}{t_p + f_n};\ \text{Accuracy} = \frac{t_p + t_n}{t_p + f_p + t_n + f_n}.$$

Precision and recall are sometimes used together with *F*-measure, which is the harmonic mean of precision and recall. The significance of using *F*-measure is that the harmonic mean in comparison to the average mean, penalizes the extreme values. Mathematically,

$$F - \text{measure} = 2\star\frac{\text{Precision}\star\text{Recall}}{\text{Precision} + \text{Recall}}.$$

4 Implementation

Opinion mining mines and examines user's opinion of a product. It is used to analyze the textual information given by users to decide whether the given review is positive, negative or neutral. Based on this approach, the proposed technique collects the user data. The user profile includes both the genuine profiles and attack profiles. Attack profiles contain PRs or NRs. From the collected user data, relevant attributes that are related to the product, such as product review and product rating, along with user basic details such as user Id are identified. Any other user demographic attributes such as age, sex, location etc., are discarded. The filtered data enter Phase 1 where pre-processing and feature extraction are applied. The pre-processing and feature extraction techniques used are discussed in the following sub-section. The model trained during Phase 1, then enters Phase 2 where opinion mining measures polarity and subjectivity values, which are calculated for all the reviews that are given by the users. Based on the evaluated values of these two measures, PRs and NRs are detected and the ratings given by the users are classified as PR, NR and normal ratings. The model is implemented in Python platform. The polarity and subjectivity measures are evaluated using the TextBlob module of Python Library.

4.1 Pre-processing

The pre-processing technique is used to provide a meaning to the raw data. It can also be used to filter the irrelevant data in the dataset. It can be used to give a shape to the data. One case may be that the product is rated with value “0.” The pre-processing technique involves various steps. The steps are explained as follows:

(*1*) ***Tokenization:*** It is the first step in which a statement is divided into various sub-statements. For example, a statement “The dish is very spicy and hot!” is divided into sub-statements as “The,” “dish,” “is,” “very,” “spicy,” “and,” “hot,” “!.”

(*2*) ***Data cleaning****: The next step is cleaning of data. This is a two-step sub procedure as follows:*

(*a*) ***Removal of special characters:*** First, the special characters such as “;”, “.”, “?” etc., are removed from the tokenized data. In the above given example, the sub-statement “!” would be removed from the tokenized data.

(*b*) ***Removal of stop words:*** *After the removal of special characters, s*top words like “the,” “was,” “he,” “she,” “they,” etc., are removed. These stop words are removed because these words do not add value to the review. In our example, the words “the,” “is” and “and” will be removed.

(*3*) ***Construct Lexicon****: Finally, s*upervised algorithm is applied on the remaining sub-statements. A sentiment score is assigned to each review. For a positive sub-statement, a sentiment score of “+1” is assigned and for a negative sub–statement, a sentiment score of “−1” is assigned. A sentiment score of “0’”is assigned for a neutral sub-statement. In the given example, a sentiment score of “0” will be assigned to “dish” and “+1” to “spicy” and “hot.” The constructed lexicon is trained with bag of words (BoW) model.

4.2 Feature extraction

Feature extraction is a technique which is used to reduce the dimensionality of data. In this process, the raw data are divided and organized into more manageable groups. This technique is more useful in larger datasets because it reduces the number of variables and thus requires lesser computing time. The BoW model can be explained with an example, as follows:

Let us consider the following three reviews:

R1: This dish is very spicy and hot

R2: This dish is not spicy and not hot

R3: This dish is yummy and good

In BoW technique, every review is represented as a string of numbers. The unique words are extracted from all the given reviews and a bag of vectors is created. Table 1 shows the BoW for the given three reviews.

In the proposed technique, BoW technique is used for the train dataset after pre-processing. The train dataset passes onto Phase 2, where opinion mining measures – polarity and subjectivity are applied.

Table 1: Example of BoW

	Unique words from all three reviews										
Review	This	Dish	Is	Very	Spicy	And	Hot	Not	Yummy	Good	Length of review
R1	1	1	1	1	1	1	1	0	0	0	7
R2	1	1	1	0	1	1	1	1	0	0	7
R3	1	1	1	0	0	1	0	0	1	1	6

4.3 Polarity

Polarity is a metric in opinion mining that evaluates the amount of polarity of emotions that are given as text review for a product by the user. The polarity of emotions may be "positive," "negative" or "neutral." Mathematically, the value of polarity lies within the range of –1 to +1. The value for more negative emotion approaches toward –1, for more positive emotion approaches toward +1 and 0 for neutral emotion. Polarity, thus, evaluates the amount of positive, negative or neutral emotions that appear in the given text review. In the proposed model, "overall polarity score" is computed for the trained BoW model by adding the score of each word of the text review.

4.4 Subjectivity

Subjectivity is another metric of opinion mining. It refers to the amount of meaningfulness of a statement, given as a text review by a user for a product. The value of subjectivity lies within the range of 0 to +1. The value approaching toward 0 implies that the text has a lesser meaning, while the value approaching toward +1 implies that the text is meaningful. In the proposed model, metric subjectivity is computed along with the metric polarity for the trained BoW model.

4.5 Flowchart

The flowchart for the proposed technique is presented in this sub-section. Figure 1 represents the flowchart for the detection of PRs and NRs. The flowchart shows that the ratings of a product are on the rating scale of 1–5. It is assumed in the proposed technique that if the rating is an average, that is, 3 then it is a genuine rating. The detection is done for higher ratings, that are "4" and "5" and for lower ratings that are "1" and "2." The subjectivity of the text review is first calculated. If the value lies within the range (0.1, 1), then further conditions are checked. But if subjectivity value is not within the range, then the corresponding rating is discarded. This rating is discarded because if the subjectivity is 0, then it means that the review is meaningless. For a review with subjectivity value lying within the specified range of (0.1, 1), the polarity is calculated. If the polarity value of the given text review is less than "0" and the rating is either "4" or "5," then it implies that the emotion of the review is negative. It further follows that a negative opinion cannot be given higher rating of "4" or "5." Hence, it is detected as a PR and is discarded. Similarly, if the polarity value is greater than "0" for the rating value "1" or "2," then it implies that the opinion is positive. But a positive opinion, logically, cannot be given lower ratings. Thus, these ratings are detected as NRs and are discarded as nuke attack. Finally, the normal ratings are stored in the database and the recommendations are generated. The proposed algorithm works on the parameters user ID, user rating given for the product (also called product rating) and product reviews (i.e., reviews given by the user for the product).

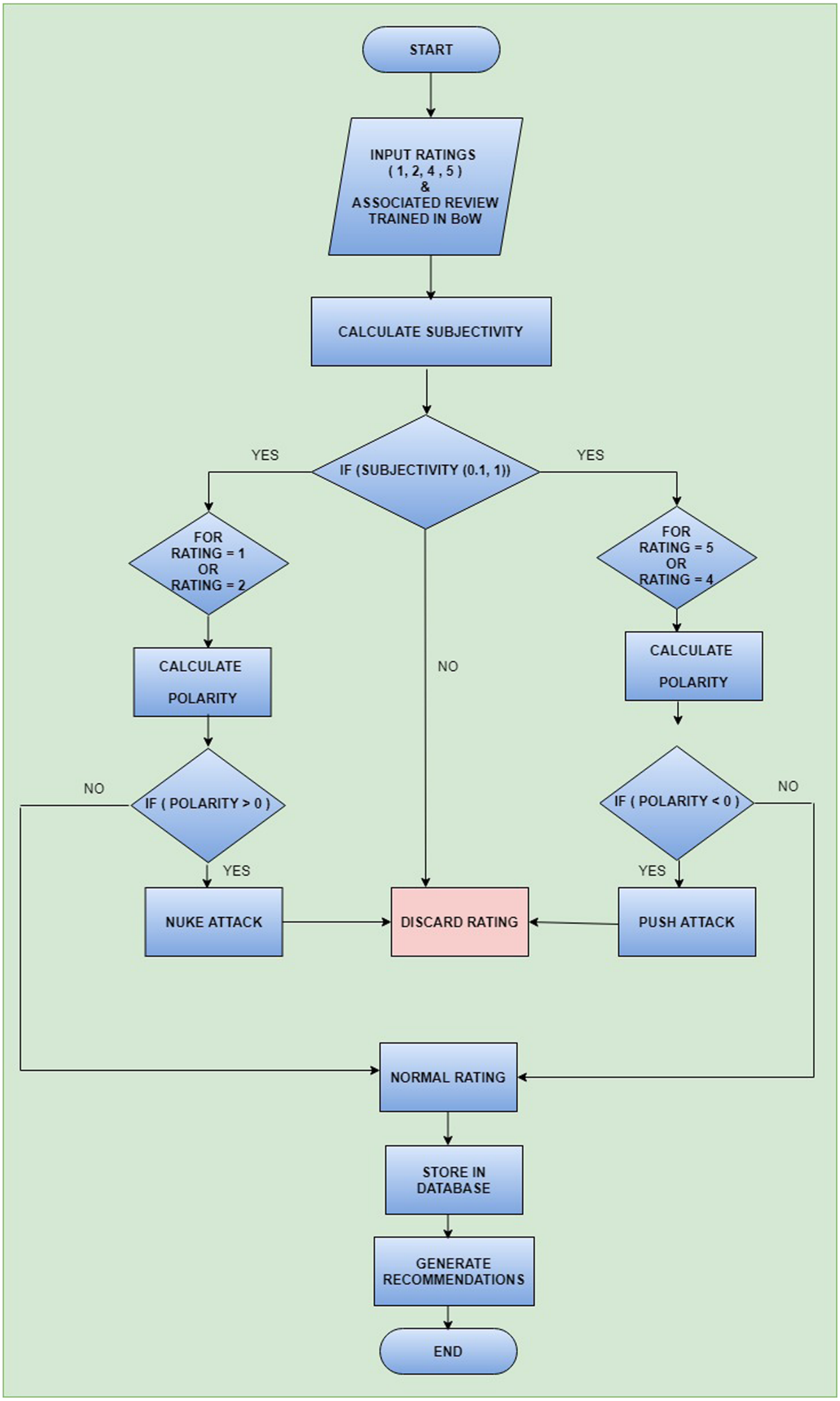

Figure 1: Flowchart to detect PRs and NRs in CFRS using opinion mining.

5 Results and discussion

To detect PRs and NRs in CFRS, opinion mining technique is used. The proposed work has been implemented in python platform and performance is evaluated using precision, recall, *F*-measure and accuracy measures.

5.1 Dataset

The datasets are retrieved from Kaggle website *https://www.kaggle.com*. Amazon dataset consists of 5,68,454 unique review tuples and the corresponding ratings in the rating scale of 1–5. Another dataset Yelp has 10,000 unique review tuples with ratings in the rating scale of 1–5. The datasets are split into training and test datasets to perform the experiment.

5.2 Experiment

The experiment is performed on two datasets namely Amazon and Yelp. Both datasets are divided into data-subsets. Train datasets of 90 and 80% each and Test datasets of 20 and 10% each are created. Measure metrics, precision, recall, accuracy and *F*-measure are evaluated for train datasets and test datasets for both Amazon and Yelp datasets.

The experiment is conducted in four iterations. At first iteration, experiment is performed on actual datasets. This means PRs and NRs are present in the dataset. At second iteration, PRs are extracted (but not NRs) from the datasets and then the experiment is conducted on remaining datasets. Similarly, at third iteration, NRs (not PRs) are extracted from the dataset and the experiment is performed on remaining datasets. Finally, at fourth iteration, both PRs and NRs are extracted and removed from the datasets and the experiment is conducted on remaining valid datasets. The results obtained in four iterations are given in Tables 2–5. Graphical comparisons of the evaluated results are shown in Figures 2–9.

Table 2: Train dataset: 90%

Dataset name	Datasets	Measures (in %)			
		Precision	Recall	*F*-measure	Accuracy
Amazon	Actual	65.2	87.2	74.6	89.29
	Actual − PR	53.6	75.7	62.7	89.68
	Actual − NR	63.1	80.2	70.6	89.33
	Actual − (PR + NR)	43.4	65.8	52.3	88.89
Yelp	Actual	12.2	35.0	18.1	70.79
	Actual − PR	12.6	35.2	18.5	70.76
	Actual − NR	12.6	35.2	18.5	70.76
	Actual − (PR + NR)	12.6	35.2	18.5	70.76

Table 2 shows the evaluation results for four iterations performed at the train dataset with 90% values of the given dataset. It may be observed, for Amazon dataset, that the *F*-measure value is 74.6% and accuracy is 89.29% when both PRs and NRs are involved, whereas *F*-measure value reduces to 52.3% and accuracy reduces to 88.89% when both PRs and NRs are removed. The higher accuracy value of a product would attract the users in making a positive decision for a product. But, in reality, this is a false recommendation with 0.4% higher value than the true value.

Table 3: Test dataset: 20%

Test dataset	Datasets	Measures (in %)			
		Precision	Recall	*F*-measure	Accuracy
Amazon	Actual	44.8	66.9	53.7	79.18
	Actual – PR	43.2	67.0	52.5	78.59
	Actual – NR	44.8	66.9	53.6	79.23
	Actual – (PR + NR)	43.0	66.8	52.3	78.42
Yelp	Actual	11.5	33.9	17.2	70.60
	Actual – PR	11.7	34.2	17.4	70.55
	Actual – NR	12.2	35.0	18.1	70.79
	Actual – (PR + NR)	11.7	34.2	17.4	70.55

Table 4: Train Dataset: 80%

Dataset name	Datasets	Measures (in %)			
		Precision	Recall	*F*-measure	Accuracy
Amazon	Actual	65.4	87.4	74.8	89.38
	Actual – PR	53.5	76.0	62.7	88.74
	Actual – NR	62.2	79.9	69.9	89.42
	Actual – (PR + NR)	43.7	66.1	52.6	88.78
Yelp	Actual	12.2	35.0	18.1	70.81
	Actual – PR	12.4	35.3	18.4	70.77
	Actual – NR	12.4	35.3	18.4	70.78
	Actual – (PR + NR)	12.4	35.3	18.4	70.78

Table 5: Test dataset: 10%

Dataset name	Datasets	Measures (in %)			
		Precision	Recall	*F*-measure	Accuracy
Amazon	Actual	42.6	65.2	51.5	78.43
	Actual – PR	43.1	67.3	52.5	78.6
	Actual – NR	42.6	65.2	51.5	78.44
	Actual – (PR + NR)	43.2	67.1	52.5	78.44
Yelp	Actual	11.8	34.4	17.6	70.78
	Actual – PR	12.0	34.6	17.8	70.74
	Actual – NR	12.0	34.6	17.8	70.75
	Actual – (PR + NR)	12.0	34.6	17.8	70.75

Table 2 shows the evaluation results for 90% train dataset for Yelp. The results show that for smaller dataset, PRs and NRs do not contribute significantly. It may be observed from the table that *F*-measure value for actual dataset (with PRs and NRs) is 18.1% and accuracy is 70.79%, while it is 18.5 and 70.76%, respectively, after these PRs and NRs are removed. This contributes to a negligible difference of 0.03% in a smaller dataset and is no point of worry. This would not affect the user in making a decision for a product. The plots for the two train datasets are shown graphically in Figures 2 and 3, respectively.

To conclude the experiment, the same four iterations are performed on 20% train datasets for Amazon and Yelp. The evaluation results are shown in Table 3. The results show that for larger datasets (Amazon) the *F*-measure value is 53.7% and accuracy is 79.18% when both PRs and NRs are present. But these values

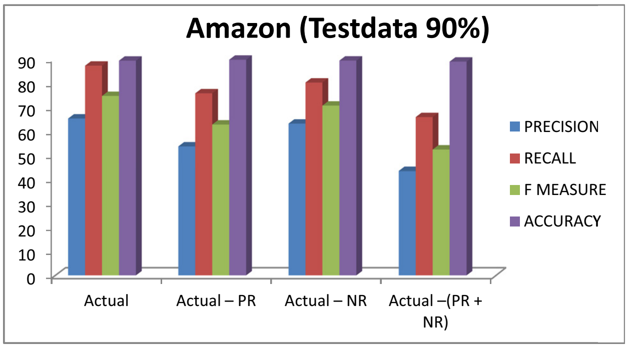

Figure 2: Plot for Amazon 90% train dataset.

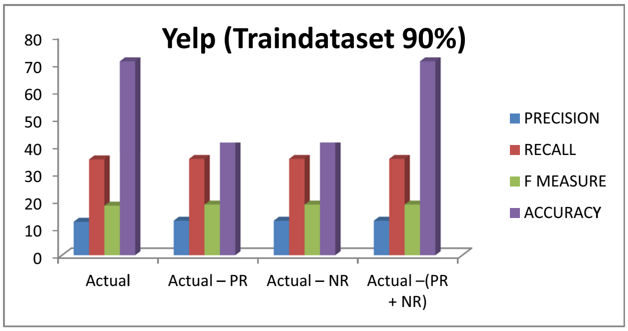

Figure 3: Plot for Yelp 90% train dataset.

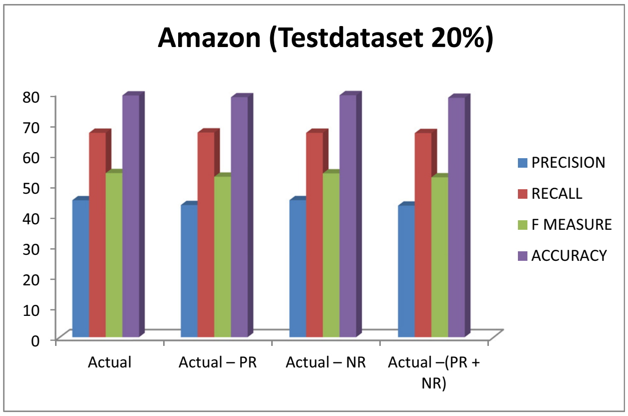

Figure 4: Plot for Amazon 20% test dataset.

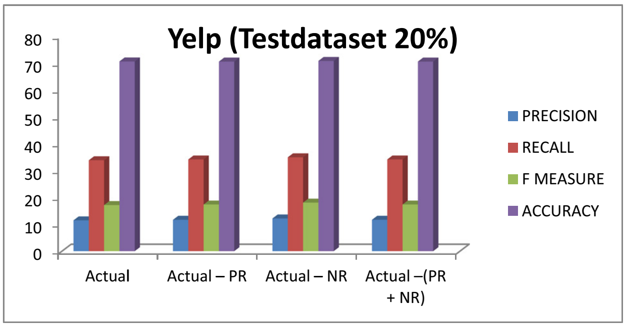

Figure 5: Plot for Yelp 20% test dataset.

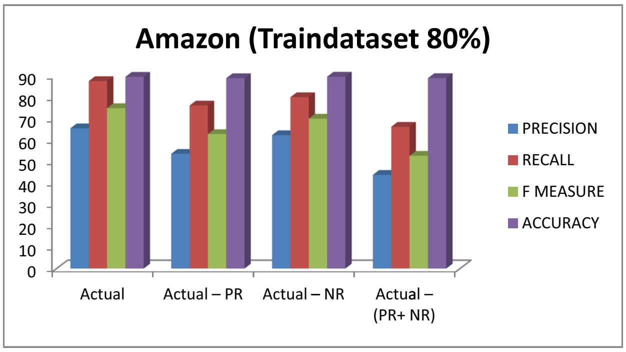

Figure 6: Plot for Amazon 80% train dataset.

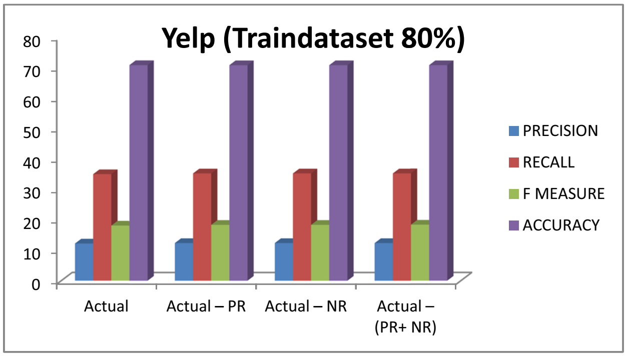

Figure 7: Plot for Yelp 80% train dataset.

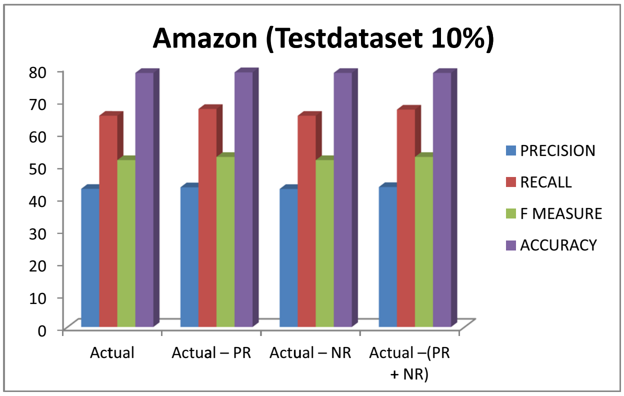

Figure 8: Plot for Amazon 10% test dataset.

reduce to 52.3 and 78.42%, respectively, when both PRs and NRs are removed. This leads to a serious concern in generating recommendations by CFRSs because if PRs and NRs are present in the dataset, wrong recommendations would be generated. This would misguide the users while making decisions for their product of interest. In future, as a result, the company may develop a negative reputation amongst its clients and users.

The Yelp dataset evaluation results are also shown in Table 3. The results show a negligible difference of 0.2% between *F*-measure value (17.2%) for actual dataset (with PRs and NRs) and *F*-measure value (17.4%) when both PRs and NRs are removed. It may also be observed that the accuracy for actual dataset is 70.60%, whereas when both biased ratings are removed, it is reduced to 70.55%. Figures 4 and 5 show the results graphically for both Amazon and Yelp datasets, respectively.

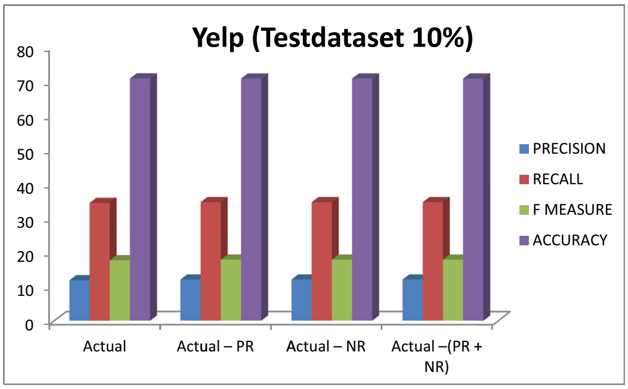

Figure 9: Plot for Yelp 10% test dataset.

Evaluation results in Tables 2 and 3 clearly show that, with PRs and NRs, there is a significant drift in *F*-measure values for larger datasets as compared to smaller datasets. The *F*-measure values are directly proportional to the recommendation of a product. When any one of these two ratings, push or nuke, or both these ratings are present, *F*-measure value leads to wrong recommendation of the product and misleads its user in decision making.

The experiment is repeated for 80% train dataset and 10% test dataset for both Amazon and Yelp datasets and is shown in Tables 3 and 4, respectively. For Amazon dataset, *F*-measure is 74.8% for actual dataset (with PRs and NRs) and reduces to 52.6% when both these ratings are removed. This validates our findings that presence of any of these ratings or presence of both of these ratings affects the recommendations of the product, misguiding to its users in selecting the product of interest. The experiment also validates that drift in *F*-measure value is directly proportional to the number of PRs and NRs present in the dataset. The more the number of PRs, the more would be the *F*-measure. The higher *F*-measure value would lead to aggravated recommendation of a product and thereby, misleading and misguiding its users in decision making of such product. Similarly, increase in the number of NRs would decrease the *F*-measure value, resulting in conciliated recommendation of a product.

The evaluation results in Table 4 also validates that for smaller datasets there is no significant drift in *F*-measure values when both push and NRs are present. Yelp dataset shows that there is a negligible difference of 0.3% between *F*-measure values for actual dataset when both the ratings are present and when both these ratings are removed. The results are depicted in Figures 6 and 7, respectively, for both Amazon and Yelp datasets, for all four iterations.

Table 5 validates the results for the experiment with 10% test dataset for both Amazon and Yelp. The evaluation results show significant drift in *F*-measure for larger dataset (Amazon) with 51.1% for actual dataset (with PRs and NRs) and 52.5% when both these ratings are removed.

Figures 8 and 9 depict the results for all four iterations performed on both Amazon and Yelp datasets, respectively. The evaluations validate that PRs and NRs affect larger datasets more significantly as compared to smaller datasets. These ratings amount to smaller drift from true recommendations that could be neglected, in smaller datasets.

The evaluation results show that it is necessary to identify and remove PRs and NRs because the presence of these ratings drifts the *F*-measure values from true values and contributes in false recommendations. These false recommendations do not only mislead the users while selecting a product but also contributes in spoiling the reputation of the company amongst its users.

5.3 Limitation of the proposed work

The proposed algorithm has two limitations:

1. The proposed work uses BoW technique for pre-processing which discards the sequence of words. The vocabulary size will increase if new reviews are added with new words. This would increase the vector length and thus will result in sparse matrix.
2. The proposed algorithm cannot deal with the human sarcasm.

6 Conclusion and future scope

In the study, the experiment is performed on two datasets, one with more than 5.5 Lacs unique user-product ratings and one with approximately 10,000 unique user-product ratings. PRs and NRs affect larger datasets more significantly as compared to smaller datasets. For larger datasets, PRs aggravate the accuracy of the RS resulting in false and aggravated predictions of the products for users. The number of NRs is observed to be lesser in number as compared to the number of PRs in the datasets. Presence of NRs alters the recommendations of the products resulting in false predictions of user-product recommendations. A more secured RS is obtained when both PRs and NRs are removed from the dataset. It is not necessary that a secured RS will have higher accuracy than a system with PRs or NRs. Secured system may have lesser accuracy. This depends on the number of PRs and NRs inserted in the dataset. For higher number of PRs, higher will be the accuracy of the system as compared to the secured RS. Lesser number of NRs does not show significant effect on the accuracy of the RS. But higher number of NRs affects the accuracy, resulting in false recommendations of products to the users. With the increase in the size of dataset, there is an increase in PRs and NRs. This results in decreased reliability for recommendations of products predicted by the RS.

This study confines the work to detect push attacks and nuke attacks in collaborative RS but it can be implemented on multi-criteria collaborative RSs as well. The multi-criteria collaborative RSs take into account multiple views collected on different features of a product. This helps to find the reason for why the user likes the product, in addition to how much the user likes the product. When the proposed work would be implemented on multiple-views, it is expected that stronger multi-criteria collaborative RS would be achieved. A more strong system would definitely yield a more secured system. Our future work will try to address this. The application of this work done in the e-commerce industry would yield good quality recommendations for the target product. This in turn, would benefit the users to select the correct product from the available online pool of products. The companies would be able to study the market trend more precisely, which is a key factor for their growth.

Conflict of interest: Authors state no conflict of interest.

References

[1] Chung CY, Hsu PY, Huang SH. βP: A novel approach to filter out malicious rating profiles from recommender systems. Decis Support Syst. 2013;55(1):314–25.
[2] Serrano-Guerrero J, Olivas JA, Romero FP, Herrera-Viedma E. Sentiment analysis: A review and comparative analysis of web services. Inf Sci. 2015;311:18–38.
[3] Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: A survey. Ain Shams Eng J. 2014;5(4):1093–113.
[4] Arazy O, Kumar N, Shapira B. Improving social recommender systems. IT Professional. 2009;11(4):38–44.
[5] Li F, Wang S, Liu S, Zhang M. Suit: A supervised user-product based topic model for sentiment analysis. In Twenty-Eighth AAAI Conference on Artificial Intelligence; 2014.
[6] Shinde SB, Potey MA. Research paper recommender system evaluation using coverage. Int Res J Eng Technol. 2016;3(6):1678–83.
[7] Ogawa Y, Suwa H, Yamamoto H, Okada I, Ohta T. Development of recommender systems using user preference tendencies: An algorithm for diversifying recommendation. In Towards Sustainable Society on Ubiquitous Networks. Boston, MA: Springer; 2008. p. 61–73.

[8] Li G, Cai Z, Yin G, He Z, Siddula M. Differentially private recommendation system based on community detection in social network applications. Security Commun Netw. 2018;2018.

[9] Musto C, de Gemmis M, Semeraro G, Lops P. A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In Proceedings of the Eleventh ACM Conference on Recommender Systems; 2017. p. 321–5.

[10] Yang D, Zhang D, Yu Z, Wang Z. A sentiment-enhanced personalized location recommendation system. In Proceedings of the 24th ACM Conference on Hypertext and Social Media; 2013. p. 119–28.

[11] Liu H, Hu Z, Mian A, Tian H, Zhu X. A new user similarity model to improve the accuracy of collaborative filtering. Knowl Syst. 2014;56:156–66.

[12] Dong R, O'Mahony MP, Schaal M, McCarthy K, Smyth B. Sentimental product recommendation. In Proceedings of the 7th ACM Conference on Recommender Systems; 2013. p. 411–4.

[13] Wang W, Zhang G, Lu J. Collaborative filtering with entropy-driven user similarity in recommender systems. Int J Intell Syst. 2015;30(8):854–70.

[14] Leung CW, Chan SC, Chung FL. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In Proceedings of the ECAI 2006 Workshop on Recommender Systems; 2006. p. 62–6.

[15] Koukourikos A, Stoitsis G, Karampiperis P. Sentiment analysis: A tool for rating attribution to content in recommender systems. In RecSysTEL@ EC-TEL; 2012. p. 61–70.

[16] Kumar A, Sebastian TM. Sentiment analysis: A perspective on its past, present and future. International Journal of Intelligent Systems and Applications. 2012;4(10):1.

[17] Wu CY, Diao Q, Qiu M, Jiang J, Wang C. Jointly modeling aspects, ratings and sentiments for movie recommendation. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD. vol. 14, 2014. p. 193–202.

[18] Alahmadi DH, Zeng X-J. ISTS: Implicit social trust and sentiment based approach to recommender systems. Expert Syst Appl. 2015;42(22):8840–9.

[19] Zhang Y. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining; 2015. p. 435–40.

[20] Singh VK, Mukherjee M, Mehta GK. Combining a content filtering heuristic and sentiment analysis for movie recommendations. In International Conference on Information Processing. Berlin, Heidelberg: Springer; 2011. p. 659–64.

[21] Musto C, Semeraro G, Gemmis MD, Lops P. Learning word embeddings from wikipedia for content-based recommender systems. In European Conference on Information Retrieval. Cham: Springer; 2016. p. 729–34.

[22] Rosa RL, Rodriguez DZ, Bressan G. Music recommendation system based on user's sentiments extracted from social networks. IEEE Trans Consum Electron. 2015;61(3):359–67.

[23] García-Cumbreras MÁ, Montejo-Ráez A, Díaz-Galiano MC. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. Expert Syst Appl. 2013;40(17):6758–65. doi: 10.1016/j.eswa.2013.06.049

[24] Beel J, Langer S, Genzmehr M, Gipp B, Breitinger C, Nürnberger A. Research paper recommender system evaluation: a quantitative literature survey. In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation; 2013. p. 15–22.

[25] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng. 2005;17(6):734–49.

[26] Zhou X, Xu Y, Li Y, Josang A, Cox C. The state-of-the-art in personalized recommender systems for social networking. Artif Intell Rev. 2012;37(2):119–32.

[27] Levi A, Mokryn O, Diot C, Taft N. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In Proceedings of the Sixth ACM Conference on Recommender Systems; 2012. p. 115–22.

[28] Morid MA, Shajari M, Hashemi AR. Defending recommender systems by influence analysis. Inf Retr. 2014;17(2):137–52.

[29] Bhaumik R, Mobasher B, Burke R. A clustering approach to unsupervised attack detection in collaborative recommender systems. In Proceedings of the International Conference on Data Science (ICDATA), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp); 2011. p. 1.

[30] Burke R, Mobasher B, Williams C, Bhaumik R. Classification features for attack detection in collaborative recommender systems. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006. p. 542–7.