International Journal of Engineering & Scientific Research
Vol.11 Issue 08, Aug 2023
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Fraud Detection in Online Reviews through Network Effects and Informed Bayesian Priors

Lokesh Lagudu

Abstract (10pt)
Online reviews significantly influence consumer behavior in e-commerce, guiding decisions about product quality and seller reliability. However, the rise of fraudulent reviews—often created for financial gain—threatens the trustworthiness of these platforms. This paper proposes a novel fraud detection approach that leverages network effects and relational analysis, focusing on the structural connections between reviewers, reviews, and businesses. Unlike traditional methods that rely on textual or behavioral cues, our method uses graph-based modeling to identify patterns of coordinated deception. We enhance an existing framework by incorporating informed Bayesian priors, which integrate temporal and behavioral irregularities to improve detection accuracy. Experimental results indicate that our approach effectively uncovers fraudulent activity, contributing to more transparent and reliable online review systems.

Keywords:
Online Reviews;
Fraud Detection;
Opinion Spam;
Relational Learning;
Bayesian Inference;
Informed Priors;
Graph-Based Models;
Deceptive Review Detection;

Author correspondence:
Lokesh Lagudu,
Email: lokeshlagudu@gmail.com

1. Introduction

In recent years, the detection of **deceptive opinion spam** has emerged as a significant research focus within the domain of online information integrity. This surge in interest is driven by the increasing reliance of consumers on online reviews to guide purchasing decisions. Products or services that accumulate a large number of positive reviews tend to attract more customers, making them a target for manipulation. Consequently, malicious actors attempt to exploit these systems by submitting fake or misleading reviews intended to deceive potential buyers. Such fraudulent activities typically take two forms: (i) posting overly positive reviews to artificially promote certain products or services, and (ii) posting negative reviews to undermine competitors and damage reputations. Research indicates that fake or manipulated online reviews exceed 30% in particular product categories thus damaging both customer trust and business financial performance

Before delving into contemporary methods for opinion spam detection, it is essential to classify the types of deceptive reviews. As outlined in [7], these can be broadly categorized into three types:

(1) Untruthful reviews,
(2) Biased reviews, and
(3) Non-reviews.

Among these, untruthful reviews—often referred to as *opinion spam* or *fraudulent reviews*—are the primary focus of this study. These can further be divided into two subcategories: hype spam, which aims to promote a target entity, and defaming spam, which aims to discredit it.

Several approaches have been proposed for detecting such spam, falling into three major paradigms: Supervised Spam Detection ([1], [2],
[7], [9]), Unsupervised Spam Detection ([3], [4],
[5], [6], [11] [12]), and Group Spam Detection [8] [13]. Broadly, existing methods can be categorized into three methodological domains:

1. Language Stylometry Analysis– focusing on textual patterns and linguistic cues
2. Behavioral Analysis – examining user activity patterns and temporal anomalies
3. Relational and Network-Based Analysis – leveraging the structural relationships among users, reviews, and entities.

In this work, we aim to extend the FRAUDEAGLE framework proposed by Akoglu et al. [1], which employs relational modeling and network effects for unsupervised fraud detection. The original framework utilizes a modified Loopy Belief Propagation (LBP) algorithm on signed bipartite graphs to identify suspicious reviewers and reviews. At convergence, the algorithm assigns scores based on maximum likelihood estimates, effectively distinguishing between trustworthy and fraudulent actors. Our proposed extension incorporates **informed priors**, derived from temporal behavioral anomalies and other auxiliary features, to enhance the robustness and accuracy of the framework. These enhancements are detailed in Section IV.

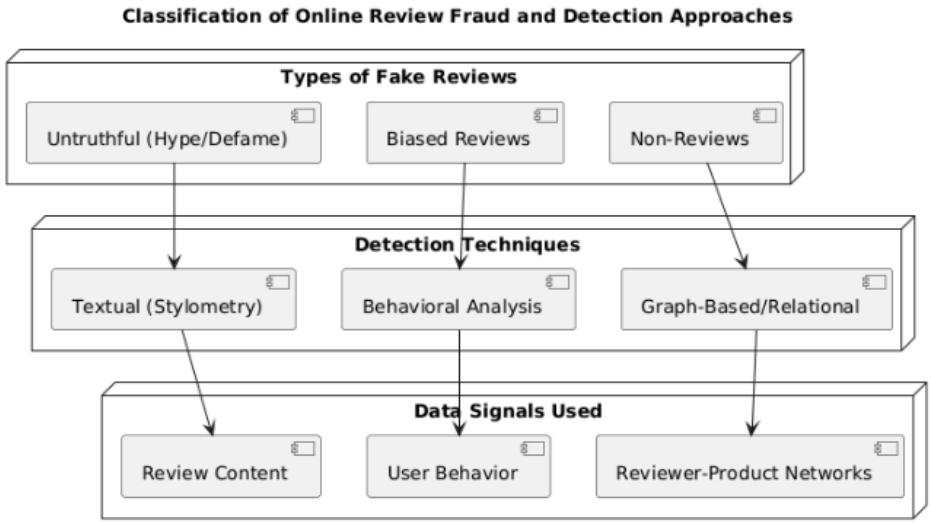

Figure 1. Classification of Online Review Fraud Types and Detection Techniques.

The above schematic diagram enables the identification of review fraud components while demonstrating how detection methods match the characteristics of available data.

2. Prior Work

2.1 Opinion Fraud Detection in Online Reviews by Network Effects

In [Akoglu et al., 2013a], authors have a proposed a new framework called FRAUDEAGLE and this method makes use of the network effect between reviewers and products for detecting fake reviews/reviewers i.e. it employs propagation based algorithm called Loopy Belief propagation (LBP). Basic idea here is to develop a model which automatically labels the reviews as fake or genuine [6].

It consists of two major steps.

1. Scoring users and reviews for fraud detection
2. Grouping them for visualizing and making sense

Extended LBP algorithm handles signed networks and at convergence it uses the maximum likelihood probabilities for scoring users and reviews. Dataset used here is a collection of app reviews for the entertainment category from an online app store database. Rating distribution for reviews has a characteristic 'J' shape.

FRAUDEAGLE detects fake reviews and users successfully and this method is totally complementary to other works that have used text and behavioral patterns to identify fake reviews/reviewers.

Relevancy

This study is particularly relevant to our work, as we build upon the FRAUDEAGLE framework by incorporating more informative priors for both reviewers and businesses, enhancing its ability to detect fraudulent behavior with greater contextual awareness.

2.2 What Yelp Fake Review Filter Might Be Doing?

A notable recent study employing behavioral analysis to detect opinion spam in Yelp data is presented by Mukherjee et al. (2013). The authors investigated Yelp's internal filtering mechanism by analyzing its filtered reviews, aiming to understand how the platform identifies and suppresses fraudulent content [2]. Their approach primarily utilized supervised learning techniques, focusing on two broad categories of features:

- Linguistic features
- Behavioral features.

While both types were explored, behavioral features proved significantly more effective in distinguishing deceptive reviews. The analysis revealed that Yelp's algorithm likely emphasizes anomalies in user behavior when flagging suspicious activity.

The study was conducted on a dataset comprising both filtered and unfiltered reviews from 85 hotels and 130 restaurants in Chicago. A linear kernel Support Vector Machine (SVM) was used for classification. Key behavioral attributes analyzed included:

1. Maximum Number of Reviews
2. Percentage of Positive Reviews
3. Review Length
4. Review Deviation
5. Maximum Content Similarity.

These features were found to be highly predictive of review authenticity. The model achieved an impressive 86% accuracy in identifying fake reviews, underscoring the strength of behavioral signals in opinion spam detection.

Relevancy

We incorporate several behavioral indicators from this study to design more informed priors within the FRAUDEAGLE framework. Key features utilized include the maximum number of reviews submitted by a user, the proportion of positive reviews, and the average length of reviews—each serving as a signal to better capture anomalous reviewer behavior.

2.3 Exploiting Burstiness in Reviews for Review Spammer Detection [10]

In [Fei at al, 2013], authors aim at exploiting the burstiness in reviews for detecting fake reviewers. There could be two reasons for sudden bursts, either due to sudden popularity of products or due to spam attacks. General trend is that reviews/reviewers that appear in a burst are often related, i.e. either fake reviewers work with other fake reviewers and genuine reviewers appear together with other genuine reviewers. This has helped authors to build a network of reviewers and then model these reviewers and their co- occurrences in different bursts as Markov Random Field (MRF). They have used Loopy Belief Propagation (LBP) algorithm to detect whether a reviewer is fake or not in the graph.

In this paper, along with considering the relationships of reviewers, reviews and stores in the graph, they have also considered relationships between reviewers themselves by linking reviewers in a burst. Authors have used Kernel Density Estimation (KDE) method to detect bursts.

Now coming to the spammer behavior features:

- Ratio of Amazon Verified Purchase (RAVP)
- Rating Deviation (RD)
- Burst Review Ratio (BRR)
- Review Content Similarity (RCS)
- Reviewer Burstiness (RB)

And these features are normalized to [0, 1]. Authors here have employed supervised learning and they aimed at exploiting the bursts in the reviews to detect fake reviews using graphs. They have also performed human evaluation and their results are consistent. But this isn't a cost- effective solution. However authors have constructed effective spammer behavior features which helped improve the results

Relevancy

Again in this paper, we have picked up spammer behavior features such as Rating Deviation, Review Content Similarity etc. and used these features to construct suspiciousness scores for both users and products and used these scores as additional prior information for the FRAUDEAGLE framework.

3. DATASET

Data Description

The Yelp Challenge Dataset includes data from Phoenix, Las Vegas, Madison, Waterloo and Edinburgh [14]. It consists of

- 42,153 businesses
- 320,002 business attributes
- 31,617 check-in sets
- 252,898 users
- 955,999 edge social graph
- 403,210 tips
- 1,125,458 reviews

Data Analysis

Figure 1 shows the star rating distribution for reviews in the Yelp Challenge dataset, with 1 being the worst and 5 being the best. Basically the reviews are more skewed towards positive ratings in this dataset.

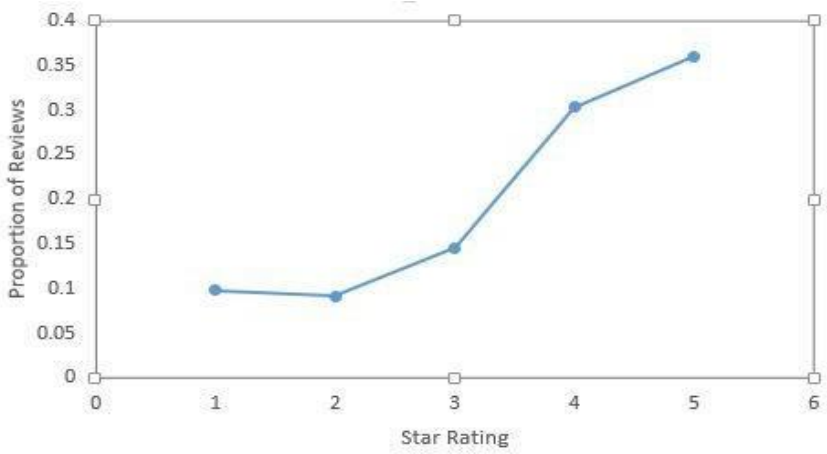

Fig 1. Star Rating Distribution

Figure 2 shows the degree distribution of users and products. As you can see that there are many product nodes with high degree. And there are several users with very few reviews.

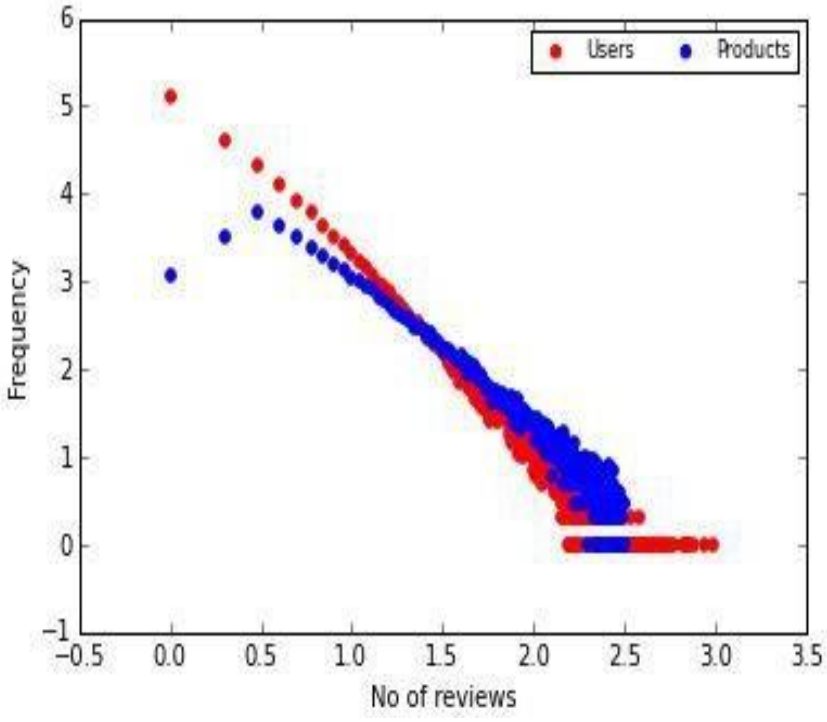

Figure 2. Degree Distribution of users and products

4. FEATURE CONSTRUCTION

Each of the features discussed below are constructed for both reviewers and products/businesses. To improve interpretability and align features with their functional role in spam detection, we organize them into four logical categories:

1. Temporal Features (e.g., burst patterns),
2. Behavioral Features (e.g., review frequency),
3. Rating-based Features (e.g., rating deviation), and
4. Textual Features (e.g., content similarity).

This categorization enables better tracking of how different dimensions contribute to user or business suspiciousness.

1. Entropy of temporal gaps (ETG) over entire timeline: We compute temporal gap between consecutive pairs of each reviewer's reviews (similarly for businesses). This temporal gap is between days (0-1, 1-2, 2-4, 4-8, 8-16, 16-32) and

then compute entropy of these gaps over entire time line and use this as a feature. The intuition here is that low temporal entropy (ETG) and large number of reviews is suspicious.

2. Entropy of temporal gaps (ETG) in windows (month-long): Here instead of considering the reviewer's/business's reviews over their entire time line, we only look at their reviews in their first month and so on and compute their ETG. We have constructed this feature up till their first 6 months in windows and have decided to use only the first month ETG as fake users do not appear in subsequent months.
3. Review length (RL): Average number of words per review indicates the review length for both reviewers and businesses [2]. It's unlikely that a spammer has much to write as it's a fake experience for him and also he might not want to spend too much time writing the review. So review length is expected to be relatively on the lesser side.
4. Maximum number of reviews (MNR): This is maximum number of reviews written in a day by reviewers or MNR written for a business in a day (normalized by dividing by the maximum value in the data across reviewers/businesses). Higher the value, more suspicious.
5. Rating Deviation (RD): Intuition here is that fake reviewers are more likely to deviate from the general rating of the businesses. So we compute the absolute rating deviation of a user's review on a business from average rating of the same business and then average the computed rating deviation across all his/her reviews.
6. Weighted Rating Deviation (WRD): This is same as the rating deviation but while computing the rating deviation we weigh each review by its recency as early reviews have more impact on businesses.
7. Average Content Similarity (ACS): Here we compute pairwise cosine similarity between all reviews of a reviewer and take the average among all pairs. More the ACS, more suspicious the reviewers.
8. Maximum Content Similarity (MCS): Instead of averaging the pairwise cosine similarity between all reviews of a reviewer, we take the maximum cosine similarity.
9. Entropy of Rating Distribution (ERD): Intuition here is that, low ERD and large number of reviews is suspicious. Including both PR and NR, more details explained in below two features.
10. Percentage of positive reviews (PR): This is basically the ratio of positive reviews(4-5star) of reviewers/businesses [2]. We expect to see spammers rating most of their reviews as 4-5 and non-spammers rating their reviews at different rating levels so that their ratings are likely to be evenly distributed.
11. Percentage of negative reviews (NR): This feature is the ratio of negative reviews(1-2star) of reviewers/businesses This feature might reveal those spammers who target to defame a particular business/product. Majority of their reviews would be 1-2 unlike non-spammers.

The features used are summarized in Table 3 which includes their categories and behavioral rationale and normalization status for quick reference. The tabular summary enables reproducibility and demonstrates how each feature supports fraud detection through Bayesian priors.

Feature	**Category**	**Intuition**	**Normalized?**
ETG (Entire)	Temporal	Low entropy with high volume is suspicious	Yes
ETG (Monthly)	Temporal	Bursty early activity is suspicious	Yes
Review Length (RL)	Behavioral	Fake reviews are usually short	Yes
Max No. of Reviews (MNR)	Behavioral	Too many in a day is abnormal	Yes
Rating Deviation (RD)	Rating	Deviation from norm = suspicion	Yes
Weighted RD (WRD)	Rating	Adds recency weight to RD	Yes
Average Content Similarity (ACS)	Textual	High similarity = templated reviews	Yes
Max Content Similarity (MCS)	Textual	Peak similarity = red flag	Yes
Entropy of Rating Distribution (ERD)	Rating	Low entropy = possible manipulation	Yes
% Positive Reviews (PR)	Rating	Too many 5-star ratings	Yes
% Negative Reviews (NR)	Rating	Too many 1-star ratings	Yes

Table 3. Summary of Engineered Features Used for Prior Construction in the FRAUDEAGLE Framework.

5. ANALYSIS

The analysis of this section evaluates statistical patterns in engineered features to prove their ability to differentiate between fraudulent and legitimate users and businesses. The analysis of patterns through visualizations leads to intuitive findings which serve as the basis for constructing the FRAUDEAGLE framework.

Entropy of temporal gaps (ETG) over entire timeline

Figure 3 shows us that there are several reviewers with low entropy and high number of reviews, this indicates suspicious behavior and for businesses.

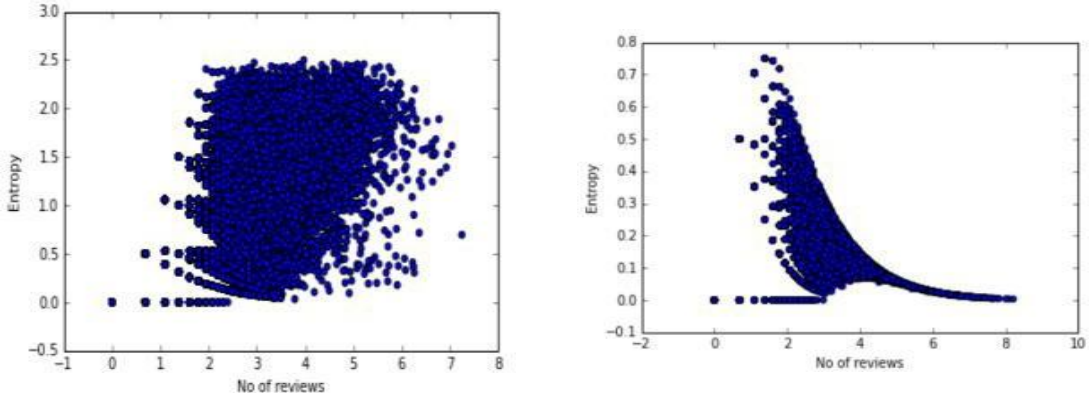

Figure 3 Entropy of entire timeline of Reviewers/Businesses Vs. No of reviews

Entropy of temporal gaps (ETG) in windows (month-long)

Figure 4 is clearly showing us that there are lot of reviewers who have written reviews in 0 or 1 day gaps and hence their entropy being very low and at the same time their number of reviews being very high. After further analysis we found out that these users with such behavior in first month have disappeared in consecutive months.

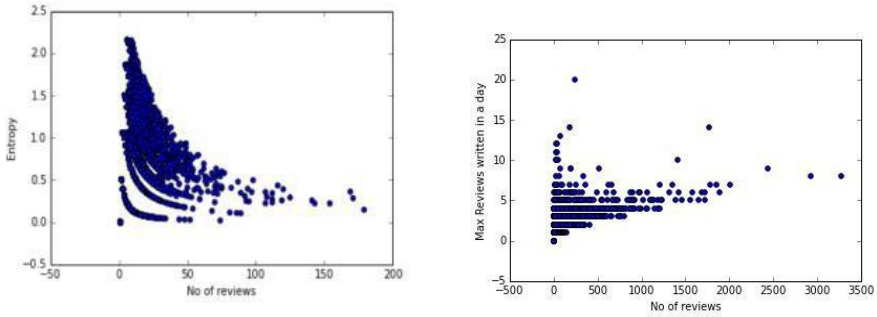

Figure 4 - ETG of first month Vs. Review count for Reviewers/Businesses

Maximum Number of Reviews (MNR)

Figure 5 is clearly showing us that we have several users who have written huge number of reviews in a day. Any user who has written more than 5 reviews in a day on an average is suspicious and also with review count being high along with the MNR. Intuition here is that writing several reviews in a day is abnormal.

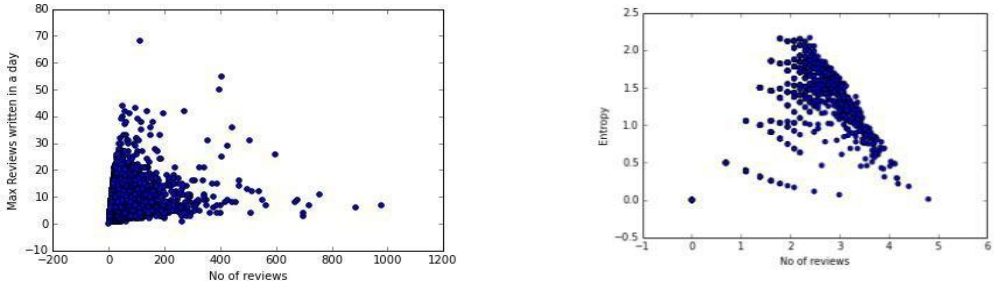

Figure 5MNR vs Review Count for Reviewers/Businesses

Review Length (RL)

Figure 6 clearly shows that there are several users with very short review length and also have written large number of reviews. They are suspicious as spammers doesn't want to spend a lot of time writing reviews, so they tend to keep the reviews short.

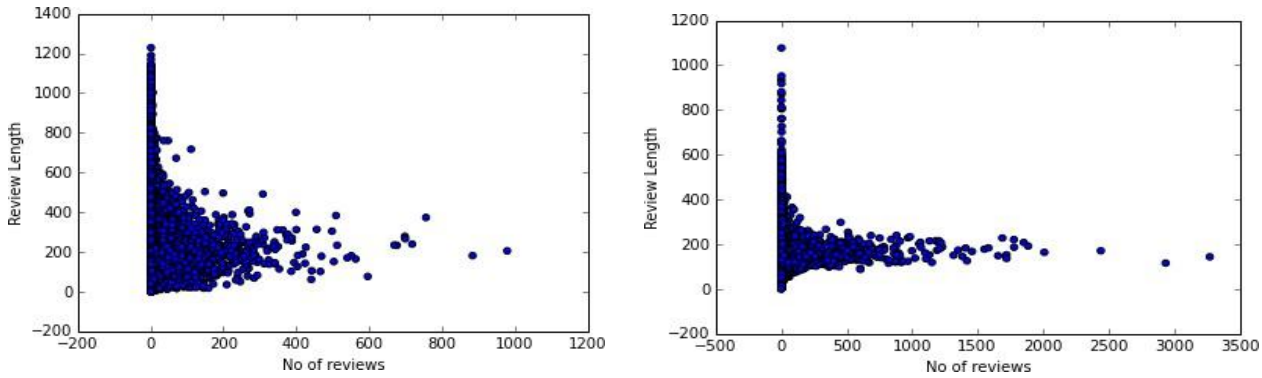

Figure 6 Review Length vs Review Count for Reviewers/Businesses

Entropy of Rating Distribution (ERD)

Figure 7 shows us that low ERD and large number of reviews is suspicious. This is lot more evident for products as shown in figure.

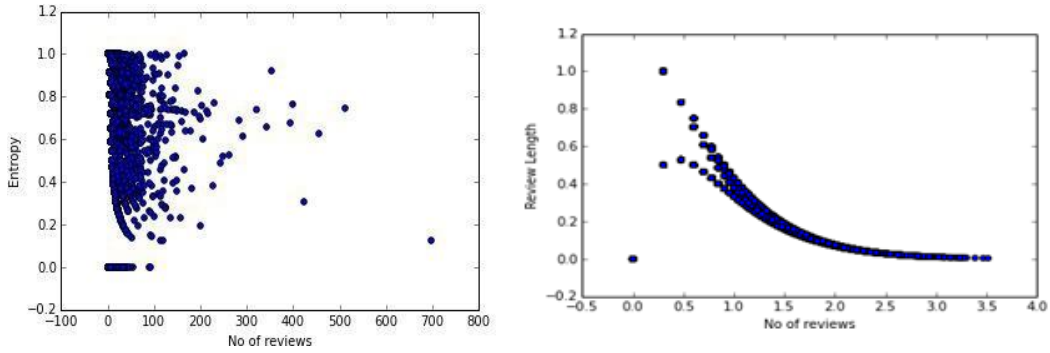

Figure 7. ERD vs Review Count for Reviewers/Businesses

Percentage of Positive Reviews (PR)

- Total No of Users: 252898
- Number of Users with >80% of PR (4-5*): 8841
- Total No of Products: 41958
- Number of Products with >80% of PR (4-5*): 6952

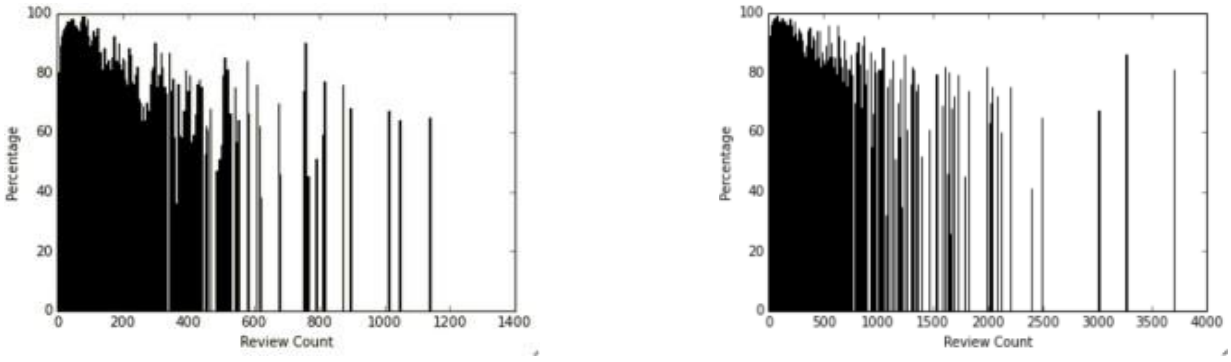

Figure 8 PR vs. Review Count for Reviewers/Businesses

Percentage of Negative Reviews (NR)

- Total No of Users: 252898
- Number of Users with >80% of NR (1-2*): 178
- Total No of Products: 41958
- Number of Products with >80% of NR (1-2*): 337
- It doesn't turn out be as indicative as PR for this dataset.

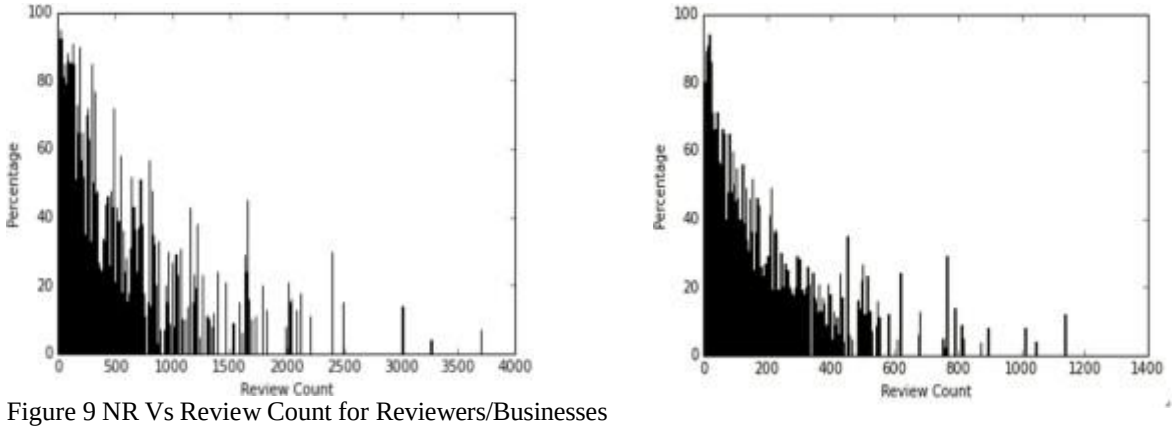

Figure 9 NR Vs Review Count for Reviewers/Businesses

The analysis shows that each feature provides distinct information for detecting deceptive behavior in online reviews. The combination of low temporal entropy with extreme rating ratios and high textual similarity and excessive daily review activity serves as indicators for spam-like behavior in users and businesses. The results support their application for building informed priors which are used in the belief propagation process of our methodology.

6. METHODOLOGY

Now for each of the features constructed above, we have come up with a suspiciousness score and used this score as a prior for the FRAUDEAGLE framework.

Scores/Priors using Entropy of Temporal Gaps (ETG vs. No of reviews)

For each user u (or similarly for products), we compute ***du = p (D>= d(u))*** where D is the overall degree distribution and d(u) is u's degree. This measures the probability of users with degree greater than or equal to the degree of u. For high d(u), du will be low.

Similarly for entropy, we compute ***eu = p(E<=e(u))*** where E is the overall entropy distribution and this measures the probability of users with entropy less than or equal to the entropy of u. For low e(u), eu will be low.

Now suspiciousness ***score = 1 - square-root [(du^2 + eu^2) / 2]*** where low du and low eu will yield high suspiciousness score.

The following high-level pseudocode summarizes the process of converting engineered features into priors and integrating them into the enhanced FRAUDEAGLE framework:

Algorithm 1: Enhanced FRAUDEAGLE with Informed Priors

Input: Review network G, Feature set F = {f1, f2, ..., fn}
Output: Suspiciousness Scores for Users, Reviews, Businesses

1. For each feature fi in F:
   a. Normalize fi into suspiciousness scores Si
   b. Initialize prior probabilities P_i using Si
2. For each prior P_i:
   a. Run modified Loopy Belief Propagation (LBP)
   b. Store belief scores B_i for nodes (users/reviews/businesses)
3. Combine belief scores (optional):
   a. Weighted or averaged across B_i
   b. Output final fraud likelihood score

1. Evaluate accuracy on labeled subset (Yelp labels or human tags)

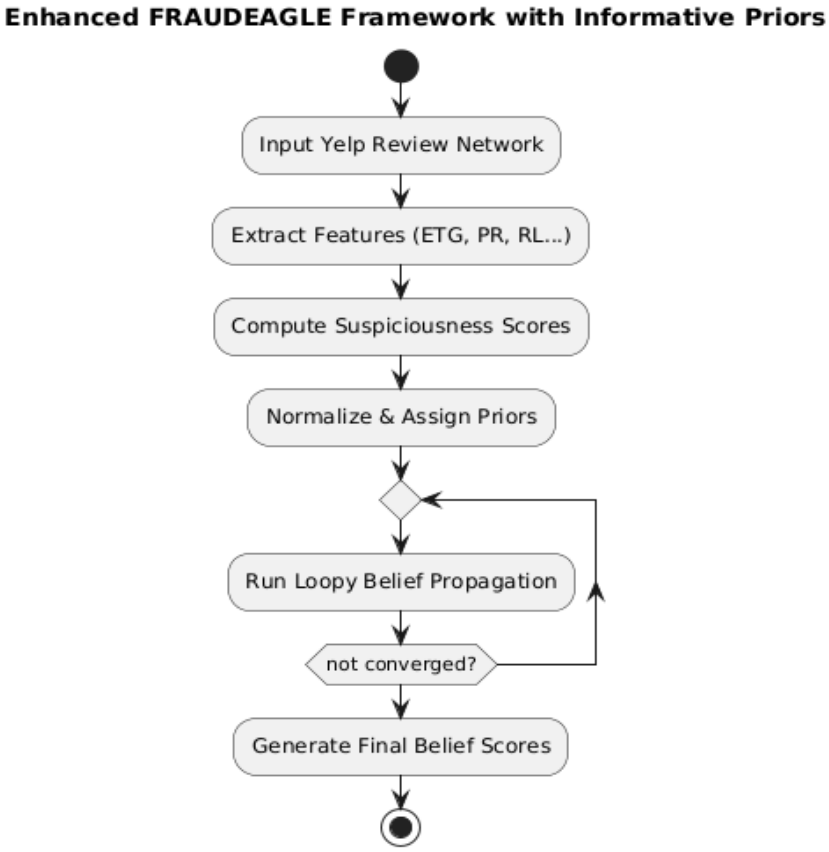

Figure 10 - The feature-to-inference pipeline visually, complementing the pseudocode steps that follow.

Scores/Priors using Percentage of positive/negative reviews

For each user u (or similarly for products), we compute ***du = p (D>= d(u))*** where D is the overall degree distribution and d(u) is u's degree. This measures the probability of users with degree greater than or equal to the degree of u. For high d(u), du will be low.

Similarly for PR, we compute ***pru = p(PR>=pr(u))*** where PR is the overall percentage of positive reviews distribution and this measures the probability of users with percentage of positive reviews greater than or equal to the percentage of positive reviews of u. For low pr(u), pru will be low.

Now suspiciousness ***score = 1 - square-root [(du^2 + pru^2) / 2]*** where low du and low pru will yield high suspiciousness score. We compute the suspiciousness scores using percentage of negative reviews in the exact same way as for percentage of positive reviews.

Scores/Priors using Review Length

For each user u (or similarly for products), we compute ***du = p (D>= d(u))*** where D is the overall degree distribution and d(u) is u's degree. This measures the probability of users with degree greater than or equal to the degree of u. For high d(u), du will be low.

Similarly for review length, we compute ***rlu = p(RL<=rl(u))*** where RL is the overall Review Length distribution and this measures the probability of users with review length lesser than or equal to the review length of u. For low rl(u), rlu will be low.

Now suspiciousness ***score = 1 - square-root [(du^2 + rlu^2) / 2]*** where low du and low rlu will yield high suspiciousness score.

1. Scores/Priors using Maximum Number of Reviews (MNR) - We are just taking the normalized value of MNR here.
2. Scores/Priors using Entropy of Rating Distribution (ERD) - Scoring is exactly as we do it for ETG.
3. Scores/Priors using Rating Deviation (RD) – Normalizing the rating deviation by 4 as 4 is the highest deviation one can have and using those values as priors.
4. Scores/Priors using Average/Maximum Content Similarity (ACS/MCS) – Taking these numbers as it is, as higher the ACS/MCS more suspicious user is. And it is between 0 and 1.

After each of these scores are calculated, they are initialized as priors separately in the FRAUDEAGLE framework and the algorithm is run for 100 iterations and we finally get belief vectors which has the probability

score for each of the reviewers/businesses if they are fake or not and if they are good or bad and similarly for reviews if they are fake or not.

Now we calculate the kappa scores for all pairwise agreements and altogether agreement to see how much are they are in correlation or agreeing with each other. Further details about evaluation/results are described in next section.

7. RESULTS

The evaluation of our proposed method takes place in two different scenarios:

- On a labeled dataset using known "recommended" vs. "not recommended" Yelp reviews, and
- On the larger Yelp Challenge dataset without labels, using orthogonal text-based features and classifier agreement.

The evaluation of accuracy and kappa agreement assesses the effectiveness of each engineered feature in improving detection when used as a prior in the FRAUDEAGLE framework.

Evaluation on Labelled dataset

Yelp provides recommended and non-recommended reviews publicly on their website. So we have considered that as a ground truth and found out the suspiciousness scores for reviewers/businesses in the labelled dataset. We have used those scores as priors to the FRAUDEAGLE framework.

Before looking at the results, let us understand what Yelp Labelled dataset consists of. It consists of

- 35,048 users
- 202 businesses
- 58,209 reviews

Some quick statistics about this dataset are shown below

As you can see from figure 11, degree distribution is quite different from Yelp challenge dataset. Now let us look at the results in table 1 below to understand how our features performed

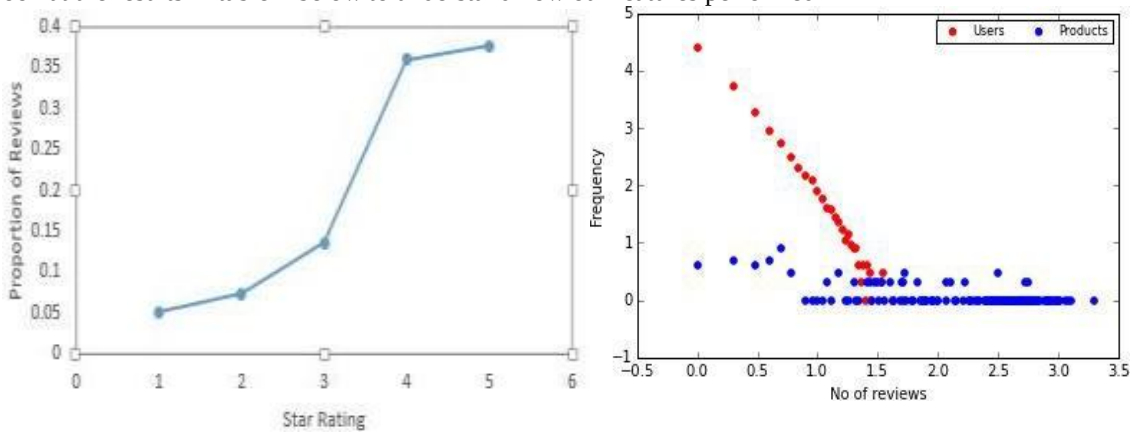

Fig 11 Star Rating Distribution Vs Degree Distribution of users/products

Sl.No	Features	Accuracy(%)
1.	Entropy of temporal gaps (ETG) over entire timeline	78.90
2.	Entropy of temporal gaps in windows (month-long)	79.95
3.	Entropy of rating distribution (ERD)	77.87
4.	Maximum number of reviews (MNR)	76.4
5.	Rating Deviation (RD)	78.5
6.	Weighted Rating Deviation (WRD)	79.1
7.	Average content similarity (ACS)	70.2
8.	Maximum content similarity (MCS)	69.8
9.	Percentage of positive reviews (PR)	75.2
10.	Combined Features (1-10)	81.5
11.	LBP + No Priors	76.2

Table 1: Labelled dataset results

As you can see LBP with no priors gave 76.2% accuracy. When we used priors for both reviewers and businesses (LBP + Priors), it gave us 81.5% accuracy. We weighed all the features equally right now when we combined the suspiciousness scores for different features. Clearly along with exploiting the graph structure, LBP with more informed priors improved the performance by 5%.

Also we evaluated on a much smaller dataset with degree distribution different when compared to Yelp Challenge dataset. So we would like to believe that the features are lot more powerful and are likely to give higher accuracies on larger labelled datasets.

Evaluation using Orthogonal Features

Now to evaluate Yelp Challenge dataset it was challenging as there is no ground truth associated with the reviews. So we have extracted below orthogonal features and trained SVM classifier using our labels from LBP + priors.

1. Unigrams
2. Bigrams
3. Distribution of POS tags
4. Percent of Positive Opinion Words
5. Percent of Negative Opinion Words
6. Percent of Capital Letters
7. Percent of Numerals

Class Distribution of Yelp data is skewed which is expected. So we performed under-sampling to randomly select a subset of instances and formed a balanced data before training the classifier.

We have performed 5-fold cross-validation and picked an optimal C value and used RBF kernel SVM. Please find the results below in table 2.

Sl.No	Features	Accuracy(%)
1.	Unigrams	75.80
2.	Bigrams	76.5
3.	Distribution of POS tags	71.3
4.	Percent of Positive Opinion Words	63.7
5.	Percent of Negative Opinion Words	64.2
6.	Percent of Capital Letters	66.8
7.	Percent of Numerals	61.2
8.	Combined Features (1 –7)	74.7

Table 2: Challenge Dataset Results

We cannot come up with precision-recall measures here as there is no ground truth. We see an agreement of about 75% to the labels we obtained from LBP + Priors. Again we are not sure here how we can take these numbers in terms of accuracy.

Fleiss Kappa Agreement

We computed the pairwise kappa agreement scores for each of the features discussed above and reported them in the table 3 below. Also the altogether kappa agreement is **0.74**

	ETG timeline	ETG first month	Review Length	ERD	MCS	Rating Deviation
ETG entire timeline	-	0.81	0.61	0.69	0.67	0.71
ETG first month	-	-	0.63	0.71	0.72	0.72
Review Length	-	-	-	0.64	0.77	0.63
ERD	-	-	-	-	0.79	0.81
MCS	-	-	-	-	-	0.77
Rating Deviation	-	-	-	-	-	-

Table 3: Kappa scores for all pairwise agreements

The results show that temporal and rating-based features—especially Entropy of Temporal Gaps (ETG) and Weighted Rating Deviation (WRD)—have strong individual predictive power. When used as priors in

FRAUDEAGLE, the combined model achieves an overall 5% accuracy improvement, showing the effectiveness of incorporating behavioral priors into a graph-based detection model.

8. CONCLUSION

So we extend FRAUDEAGLE framework by including more informed priors and this definitely seems to have a positive effect on the performance. For labelled dataset, there is a performance gain of about 5%. Along with all the advantages this framework has to offer, we have included priors from complementary approaches, in this case behavioral analysis. So we have exploited the network structure of the review network along with incorporating behavioral clues into the model. Now talking about next steps, in the current model we have given equal weights to all the features. We could/should definitely weigh features and try the LBP + Priors to see the improvement in performance. Also we want to employ Co-training which is a semi-supervised learning technique as we have large amounts of unlabeled data and very less labelled data.

1. We couldn't try out some of the features due to pressing time. It would be interesting to see how below features perform.
2. Rating distribution deviation (RDD): given all products reviewed by a reviewer i, find all other reviewers that reviewed atleast 90% of the same products, call them F for 'friend' or 'similar users who reviewed similar set of products
3. Review Burstiness (BST)
4. Ratio of Singletons (RS)

Also we could exploit user connections and find out anomalies based on the user-connection network. The enhanced FRAUDEAGLE framework demonstrates potential for fraud detection in dynamic review systems through its interpretable priors and unsupervised design and consistent performance. Future work will concentrate on weight optimization and cross-domain extension.

REFERENCES

[1] Ott, Myle, et al. "Finding deceptive opinion spam by any stretch of the imagination." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.
[2] Mukherjee, Arjun, et al. "What Yelp Fake Review Filter Might Be Doing?."ICWSM. 2013.
[3] Feng, Song, et al. "Distributional Footprints of Deceptive Product Reviews."ICWSM. 2012b.
[4] Jindal, Nitin, Bing Liu, and Ee-Peng Lim. "Finding unusual review patterns using unexpected rules." Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.
[5] Wang, Guan, et al. "Review graph based online store review spammer detection." Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011.
[6] Akoglu, Leman, Rishi Chandy, and Christos Faloutsos. "Opinion Fraud Detection in Online Reviews by Network Effects." ICWSM. 2013
[7] Jindal, Nitin, and Bing Liu. "Opinion spam and analysis." Proceedings of the 2008 International Conference on Web Search and Data Mining. ACM, 2008.
[8] Mukherjee, Arjun, Bing Liu, and Natalie Glance. "Spotting fake reviewer groups in consumer reviews." Proceedings of the 21st international conference on World Wide Web. ACM, 2012.
[9] Feng, Song, Ritwik Banerjee, and Yejin Choi. "Syntactic stylometry for deception detection." Short Papers-Volume 2. Association for Computational Linguistics, 2012a.
[10] Fei, Geli, et al. "Exploiting Burstiness in Reviews for Review Spammer Detection." ICWSM. 2013.
[11] Akoglu, Leman, Hanghang Tong, and Danai Koutra. "Graph-based Anomaly Detection and Description: A Survey." arXiv preprint arXiv:1404.4679 (2014).
[12] Li, Jiwei, et al. "Towards a General Rule for Identifying Deceptive Opinion Spam."
[13] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM Computing Surveys (CSUR) 41.3 (2009): 15.
[14] Yelp Challenge Dataset http://www.yelp.com/dataset_challenge