Fraudulent review detection model focusing on emotional expressions and explicit aspects: investigating the potential of feature engineering

Abstract

Reading customer reviews before purchasing items online has become a common practice; however, some companies use machine learning (ML) algorithms to generate false reviews in order to create positive brand images of their own products and negative images of competitors' offerings. Existing techniques use review content to identify fraudulent reviewers; however, spammers become more intelligent, started to learn from their mistakes, and changed their tactics in order to avoid detection techniques. Thus, investigating fraudulent accounts' behaviour of generating fake negative or positive reviews for competitors or themselves and the necessity of ML classifiers to identify fraudulent reviews, is more important than ever. In this research, we present a novel feature engineering approach in which we (1) extract several "review-centric" and "reviewer-centric" features from a dataset; (2) combine the cumulative effects of features distributions into a unified model that represents overall behavior of the fraudulent reviewers; (3) investigate the role of effective data pre-processing to improve detection accuracy; and (4) develop a probabilistic approach to detect fraudulent reviewers by learning a novel M-SMOTE model over a derived balanced dataset and feature distributions, which outperforms other ML models.

Keywords: online reviews, digital platforms, review manipulation, machine learning, opinion spamming, feature engineering

"Coming up with features is difficult, time-consuming, requires expert knowledge. 'Applied machine learning' is basically feature engineering." – Prof. Andrew Ng

Introduction

Product reviews can make or break a business in the world of e-commerce. According to a study by BrightLocal [46], 91% of consumers report that positive reviews make them more likely to use a business, 82% will reject a business based on negative reviews, and 76% trust online reviews as much as personal recommendations. Consumers depend on online reviews to decide which movies they should watch, where to go for dinner, what to read, where to go on vacation, and which company's products they should purchase. Products with higher percentages of good reviews top the search results on websites like Amazon and Yelp, which essentially earns them an abundance of free exposure, as they are "highlighted" or "recommended" and even promoted in emails. Many companies feel unable to respond to negative reviews, which can destroy their reputations and diminish their profits.

In 2016, Alibaba highlighted the increasing proliferation of false reviews when it sued a third-party service provider for linking its merchants with people who were willing to falsify purchases and post positive comments to boost their rankings [9]. In what is commonly called a "brushing operation", real customers paid for their transactions on Alibaba, submitted their positive reviews and ratings, and then recouped their money through electronic credits or other e-currency forms. Similarly, Amazon filed a lawsuit in 2015 against over a thousand Fiverr users for offering to post fraudulent reviews [10]. In 2019, the US Federal Trade Commission successfully brought the first-ever case of using fraudulent Amazon reviews to advertise an online product, whereby a company paid a third-party website to write five-star Amazon reviews for a weight-loss supplement that was widely advertised as enhancing weight loss, but actually causes acute liver failure [15].

The impact of such operations is twofold: they swell transaction figures on the vendor platform and artificially lift sellers' rankings. In the former case, the platform's reputation can be damaged, as news of such activities makes customers question its legitimacy. Whereas existing technologies can often detect individual incentivized reviewers, on a larger scale, opinion spammers tamper with existing systems to produce negative reviews, and the amount of fraudulent reviews produced by machines is substantially increasing.

In order to combat online review manipulation, many companies are seeking more effective strategies for identifying fraudulent reviews and reviewers [28]. Information systems (IS) domain is an emerging area of research in this area. Opinion mining and sentiment analysis have increased, and

a wide range of algorithms is used to detect spammer's activity. However, spammers have become increasingly diligent in adapting their tactics to avoid detection techniques [20].

One of the most important elements of fraudulent review detection models is feature engineering, which entails a procedure of transforming raw data into novel features and selecting the best variables that improve the ML classifier's accuracy. A number of existing studies of fraudulent review detection have applied supervised ML algorithms as their baseline either with or without performing feature-engineering techniques [3, 41, 49, 52], and there have also been some contributions on unsupervised ML algorithms [2, 14, 29, 51, 58]. A few studies have examined non-ML techniques with derived features [30, 48]. However, comprehensive studies of feature engineering, particularly the distributional characteristics of reviewers' features with data pre-processing challenges and class imbalanced problems, remain limited in the literature.

Fraudulent reviews tend to deform a feature's underlying natural distribution [30], and a few studies have analyzed underlying distributions in online product reviews such as power law and J-shaped distributions [11, 23]. Feng et al. [16] elucidated the characteristics of natural opinion distribution with respect to TripAdvisor hotel reviews and Amazon product reviews, and Dalvi et al. [11] devised an average rating distribution in domains such as restaurants and movies. These studies concluded that rating distributions are heavily skewed by highly imbalanced datasets; however, they did not specifically devote attention to understanding how underlying distributions are related to the detection of fraudulent reviewers.

Overall, we found that no comprehensive distributions that consider specific aspects of feature engineering and imbalanced classification have been conducted to identify fraudulent reviews and opinion spammers. To address this gap, we aimed to devise an approach for detecting potentially fraudulent reviews using an innovative feature engineering approach. A M-SMOTE (modified-synthetic minority over-sampling technique) model inclusive of a combination of several univariate user-reviewing distribution-based transformation of features is developed to detect fraudulent reviews based on patterns in online review data produced by the skewing of features. Our primary objective is to investigate the potential for using the review- and reviewer-centric features of users to develop fraudulent review detection models, and our second objective was to investigate the impact of data pre-processing and feature engineering tasks on ML classifiers to identify fraudulent reviews with high accuracy and develop an M-SMOTE model to solve the class imbalance problem. We empirically tested the model's ability to glean underlying distributional aspects of reviewer behavior and then applied it on a Yelp dataset and several other benchmarking datasets to analyze the impact of data pre-processing challenges in classification performance.

Related literature

A number of existing studies have examined ML techniques to detect fraudulent reviews and reviewers. Supervised ML methods include support vector machines (SVM) [20, 32, 35, 41], logistic regression (6, 20, 26, 41] ordinal multilevel regression [43], k-nearest neighbor [49, 50], random forest [22, 60], decision trees and J48 [5, 26] naïve Bayes [50, 61], boosting & bagging algorithms [18, 19], and artificial neural networks (ANN) [36, 52]. The considerable body of literature on unsupervised ML algorithms includes the FRAUDEAGLE clustering method [3], SPEAGLE clustering [51], the unified unsupervised review deviation model [40], dynamic k-value aggregation [21], the lexicon-based unsupervised model [24], a statistics-based unsupervised clustering algorithm [13], the unsupervised topic-sentiment joint probabilistic model [14], mixture models [29], and the unsupervised matrix iteration algorithm [58].

The majority of previous research has applied traditional supervised and unsupervised ML techniques to solve the problem of fraudulent reviewer detection using linguistic features (word unigrams and bigrams, LIWC features and POS features), user-behavior features (average review length, standard deviation in ratings etc.) and others reviewer-centric features. Jindal and Liu [23] experimented with supervised ML techniques to identify opinion spam by manually labelling an Amazon dataset based on a linguistic and behavioral analysis of fraudulent reviewers. Ott et al. [47] used basic features such as unigrams, bigrams, trigrams and part-of-speech (POS) to develop a SVM-based classifier to detect fraudulent reviews on a TripAdvisor dataset, and Li et al. [34] developed a supervised ML-based framework with a co-training method to identify fraudulent reviewers and reviews based on pre-identified features. Lin et al. [39] proposed six features (personal content similarity, product review similarity, similarity with reviews on other products, reviewer's review frequency, product review frequency, and repeatability) and experimented with traditional supervised ML algorithms to detect fraudulent reviews. Abbasi et al. [2] extracted several unique features and used statistical learning theory to develop a fraudulent website detector system. Zhang et al. [60] used supervised ML techniques (SVM, decision tree and random forest) with verbal (n-grams and POS) and non-verbal features to detect fraudulent reviews on a balanced (equal number of fraudulent and real reviews) Yelp dataset. Kumar et al. [28] similarly used a Yelp dataset to apply several traditional ML techniques (logistic regression, k-NN, boosting algorithms, SVM etc.) with univariate features (including review gap, review count, rating entropy, rating deviation, time of review, and user tenure) to detect fraudulent reviews. Siering et al. [53] used several linguistic

features (complexity, expressivity and diversity) in textual information with ML methods to examine the manipulation in crowdfunding projects, and Ren and Ji [52] developed a LIWC framework based on POS, n-grams, and psychological features. Finally, Li et al. [36] used unigram, POS, and LIWC to create a deep learning model to detect fraudulent reviews on hotel and restaurant datasets.

We have observed several limitations in the information systems (IS) literature for detecting fraudulent reviews using feature engineering with review-centric features, reviewer-centric features, and behavioral characteristics. Although a few studies have demonstrated the use of advanced ML and deep learning techniques to detect fraudulent reviews [7, 36, 56, 59, 62], research with novel feature sets and feature engineering remains uncommon. In addition, to the best of our knowledge, although a few studies in the IS domain have used traditional ML techniques with pre-existing feature sets to detect fraudulent reviews or reviewers, accounting for data imbalance problem and other data pre-processing challenges remains insufficiently explored.

Our study diverges from previous research in that rather than using pre-determined features, we incorporate several review- and reviewer-centric features as well as fraudulent reviewer behavioral characteristics into a ML model. The main advantages of our proposed model over other ML algorithms are its capacity to learn high-level features from balanced datasets and execute part of the feature engineering on its own. The proposed model scans the data to search for important features that have some correlations and combine them to improve the accuracy of the ML classifiers without being specifically instructed to do so. Similar studies by Kumar et al. [28, 29] cannot be considered conclusive because they did not focus on solving any data pre-processing challenges and they only highlighted the skewed distribution of some features to improve the accuracy of machine learning classifiers.

Our proposed research also differs from the existing literature by focusing on fraudulent review detection in a hierarchical manner using a novel feature engineering method that is very helpful for improving the accuracy of classifiers. Zhang et al.'s [60] research can be considered a first step towards a more profound understanding of review-centric and reviewer-centric features; however, a number of questions regarding data pre-processing and improving the accuracy of ML classifiers a remain to be addressed. Similarly, Kumar et al.'s [28, 29] contribution to the engineering literature mainly focused on only one dimension (positive and negative skewed distribution of features), and although they used some transformed reviewer-centric features to understand the univariate and joint behavior of features, they devoted less attention toward improving the accuracy of ML classifiers with other feature engineering tasks. To the best of our knowledge, no previous research has investigated the importance of feature engineering and class imbalance for detecting fraudulent

reviewer's behavior. Thus, the use of data pre-processing methods to exploit features in fraudulent review prediction is a promising area of research.

Conceptual background and research hypotheses

Features drawn from skewed distributions with imbalanced classifications that are directly used in ML algorithms tend to decrease prediction accuracy, which is illustrated in our baseline model with an undisturbed natural distribution of features. Initially, we built ML models with natural transformations and imbalanced datasets, and then we added new, transformed features and a balanced dataset to perform machine learning.

Figure 1 illustrates the proposed research framework, showing both scenarios, i.e., a) class imbalanced and b) transformation using distributional characteristics of relevant univariate features. The results will show that our proposed model with specific review and reviewer-centric features outperforms the traditional ML models that do not use normalized, transformed features and balanced datasets.

Data pre-processing and feature engineering are the most crucial parts of any data science project, and the data on which it operates is the heart and soul of any machine-learning problem. The data used to construct the ML model plays the most critical part in determining its predictive power. The better the features we generate, the more accurate the results we obtain. However, some questions continue to be debated [1, 31]: To what extent can data be operated on; How far can we generate the features; and How many features are sufficient? Data pre-processing is highly impacted by the hypothesis generation, which is a desire to utilize existing data to learn a trend that would best map the inputs to outputs. The more we invest in hypothesis-development, the better features we generate, and the greater accuracy we achieve for our predictive model.

Accordingly, we propose several research hypotheses and contributions. First, we developed a comprehensive set of features, including some novel review-centric features as well as reviewer-centric features that previous studies have used to identify the characteristics of fraudulent reviewer accounts. Second, we tried to address the question of whether reviewer-centric features are more useful than review-centric features for detecting fraudulent reviews, as stated in the following hypotheses:

Hypothesis 1: *Compared with review-centric features, reviewer-centric features of users are more important for detecting fraudulent reviewers' behavior.*

Hypothesis 2: *Combining reviewer-centric with review-centric features will improve the accuracy of ML detection models in comparison with using review-centric features alone.*

Third, we developed a rule-based ML framework that highlights the role of data pre-processing tasks for enabling marketing managers to easily detect fraudulent reviews, as stated in the following hypotheses:

Hypothesis 3: *Compared with symmetric distribution, features from skewed distributions that are directly used in machine learning will reduce the accuracy of predictive models.*

Hypothesis 4: *Compared with imbalanced and raw (not pre-processed) data, balanced and processed data (after solving all pre-processing challenges) will lead to better prediction and improved accuracy.*

To summarize, our main contributions are as follows:

1. We developed a novel probabilistic approach to detect fraudulent reviewers by learning a proposed M-SMOTE ML algorithm over a derived balanced dataset and feature distributions. We tested four hypotheses and extracted some unique review- and reviewer-centric features and behavioral characteristics of opinion spammers that are very helpful for detecting the fraudulent reviewers in the proposed framework.
2. We performed a comprehensive experimental evaluation of our approach on real-world restaurant reviews taken from Yelp.com and combined the cumulative effects of several feature distributions into a unified model that represents the overall behavior of the fraudulent reviewer. Specifically, we extracted 12 unique features and developed several ML models on a balanced dataset. Furthermore, we compare our approach with those of six previous studies that have evaluated Yelp and Amazon reviews, namely Feng et al [16], Akoglu et al. [3], Rayana and Akoglu [51], Zhang et al [60], Kumar et al. [28, 29], demonstrating that our model outperforms all of them in terms of accuracy and other statistical metrics.

Our findings provide several managerial and practical implications to help practitioners and marketing managers more effectively combat fraudulent reviews on e-commerce websites. Overall, our proposed model will improve revenue-generating opportunities and customer experience for both digital platforms and businesses.

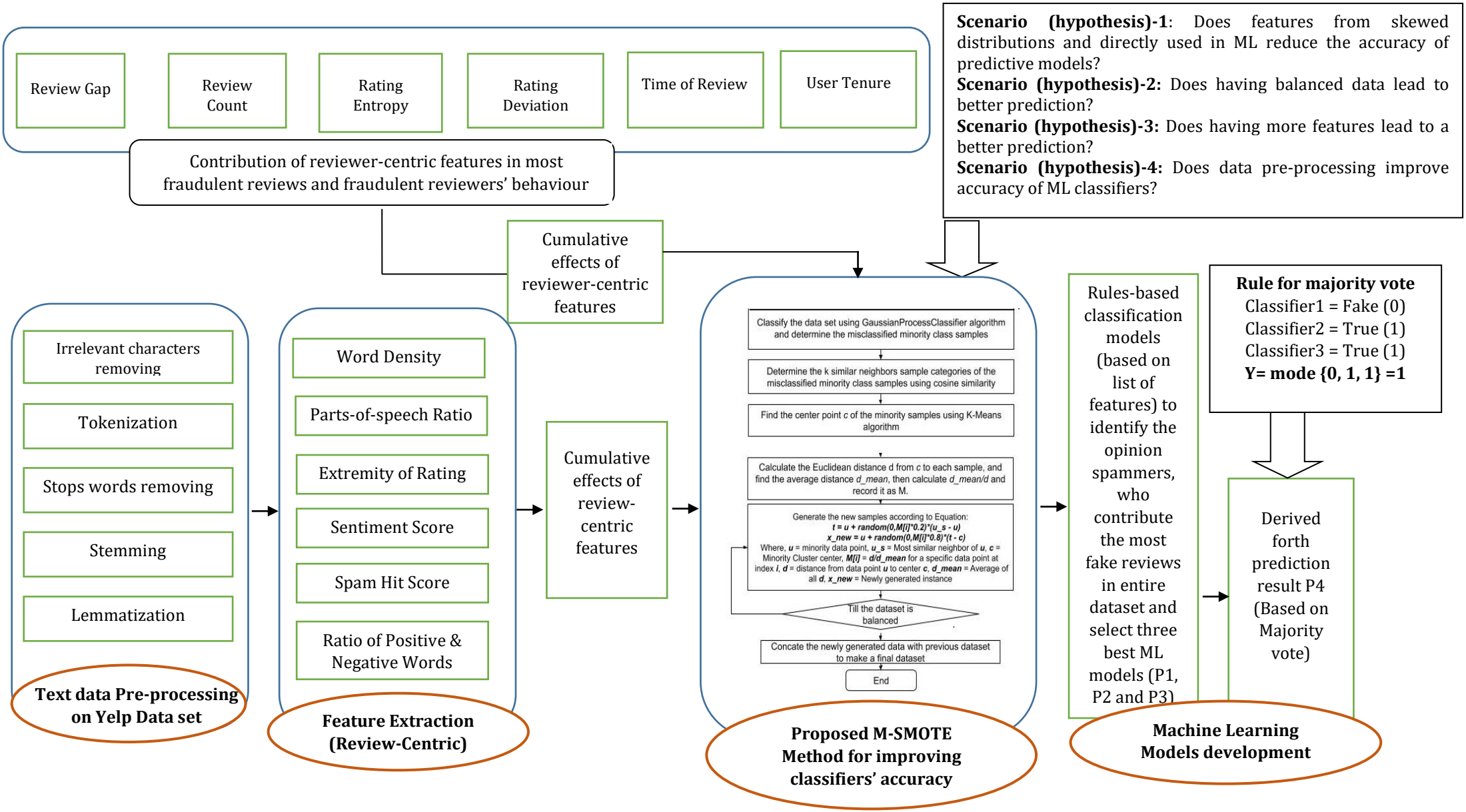


Figure 1: Proposed fraudulent review detection model based on M-SMOTE algorithm

Data Description and model development

One of the important challenges of building ML models for identifying fraudulent reviews is obtaining reviews that have been clearly identified as being fraudulent. Several existing models have used pseudo-fraudulent reviews that were either manually annotated or generated by Amazon Mechanical Truckers rather than officially filtered fraudulent reviews [44, 45]. However, using a manually annotated dataset in our research would have been fundamentally problematic. Previous studies [44, 60] suggested that when Amazon Truckers write fraudulent reviews in comparison to actual fraudulent writers, the accuracy of predictive models is generally much higher than the accuracy of models developed on real-world fraudulent review because Amazon MT have different psychological states of mind when they write the fake reviews in comparison to actual fraudulent review writers..

We decided to use the real-world dataset that Rayana and Akoglu [51] collected from Yelp.com, which provides several behavioral characteristics of fraudulent accounts. Our data encompasses 5044 restaurants in four U.S. states, namely Connecticut, New Jersey, Pennsylvania, and Virginia, and entries from 260,277 reviewers who posted from 2010-2014. The original dataset had only six variables: user ID, product ID, rating, label, date, and text review; however, we extracted 12 new features (six review-centric and six reviewer-centric features) for our fraudulent review detector models. We observed that 35,600 reviews (~6 %) have single star ratings, 42,985 s (~7 %) have two-star ratings, 83,139 (~14%) have three-star ratings, 217,465 (~35 %) have four-star ratings, and 229,409 reviews (~38%) have five-star ratings.

Figure 1 illustrates the four stages of model development: (i) text data pre-processing; (ii) feature extraction or new variable creation; (iii) feature engineering for improving the ML accuracy; and (iv) fraudulent review detection model development. This framework receives restaurant reviews and the relevant information of each reviewer as input. For the first task, we created six reviewer-centric features (rating entropy, review gap, review count, rating deviation, time of review, and user tenure) and six review-centric features (word density, review length (number of words), parts-of-speech ratio, ratio of positive and negative words, sentiment score, and SpamHitScore).

Several steps are needed to extract the review-centric features from raw text, namely tokenization, stop-word removal, stemming and lemmatization. Tokenization is the process of splitting a sentence, paragraph, or an entire text document into individual words or terms,

which are called tokens. Stop-word removal is the process of removing "noise" words (e.g., is, are, am, this, that, a, an, the), and stemming is the process of removing the word suffixes to retain base words. Lemmatization is similar to stemming in that it groups together the different forms of a word; however, it brings context to the words and links words that have similar meaning [4].

In the next stage of the proposed framework, we present a dashboard of several data pre-processing solutions that will help remove noise, inconsistency, outliers, missing values, skewed distribution, class imbalance, correlated variables, and multicollinearity and then create several hypotheses based on data pre-processing and machine learning classifier performance.

In the final stage, we develop a number of machine learning models for identifying fraudulent reviewers from the dataset and use the case of majority voting to assess the performance of the best three machine learning classifiers. We apply a simple rule of hard voting for predicting the class label Y via majority voting of three best classifiers [Y = mode{M1(X),M2(X),M3(X)}]. For example, if our three best ML classifiers give the prediction results if Classifier_1 = Fraudulent (1), Classifier_2 = True (0) and Classifier_3 = True (1), then we would use the majority vote concept and the final prediction result would be Y= mode {1, 0, 1} =1. This would be the fourth prediction result, and we would consider it the final output of our ensemble vote classifier. Along with developing the fraudulent review predictive model, we will test all four hypotheses in the final stage of the proposed framework.

Review- and reviewer-centric feature generation

We initially extract various univariate and multivariate features to predict suspicious activity based on different characteristics of reviewing behavior. We include six features, namely rating entropy, user tenure, review gap, time of review, review count and rating deviation, to which we added word density, parts-of-speech ratio, extremity of rating, ratio of positive and negative words, sentiment score and SpamHitScore. We empirically identify the best fitting distribution family for statistics related to both older and new features. Our study not only focuses on developing a ML detector model that introduces several new review-centric features, but also contributes to the distribution-based transformation of existing features. In this section, we explain the univariate and bivariate review- and reviewer-centric features used in our analysis.

Reviewer-centric features

(i) *Rating entropy:* Luca and Zervas [42] empirically proved that honest reviewers are highly likely to balance their activity between critical or noncritical reviews, whereas fraudulent reviewers generally post uniformly extreme reviews aiming to either artificially enhance or damage a company's ratings. The entropy rate for fraudulent reviews tends to be small due to the lack of balance, and Figure 2(a) illustrates their typically highly skewed distribution. In order to assess the reviews' randomness and genuineness, the entropy of rating scores is calculated as shown in Equation 1 [44]:

$$E_a = \sum_{b=1}^{n} P_{a,b} \log(P_{a,b}) \quad (1)$$

where $P_{a,b}$ = probability of user a if user has assigned the b review score.

(ii) *Review gap:* Mukherjee et al. [45] showed that fraudulent reviewers are usually not registered on an e-commerce website for a long period of time; thus, suspicious behavior is indicated in cases when a reviewer has posted all of their reviews within a short time span. Real reviewers generally only use their accounts from time to time to post reviews, so if reviews are posted over a relatively long timeframe, we can consider that normal activity. The highly skewed distribution of this feature can be seen in Figure 2(b) and we formalize review gap as shown in Equation (2):

$$G_a = \frac{1}{N_a - 1} \sum_{b=2}^{N_a} (T_{a,b} - T_{a,b-1)}), \quad (2)$$

where N_a = the number of reviews written by a particular reviewer and $T_{a,b}$ corresponds to the timestamp of each review for a particular reviewer.

(iii) *Review count*: Mukherjee et al. [44, 45] demonstrated that paid and fraudulent users generally write more reviews than real reviewers. Thus, number of reviews associated with a particular account could be an important factor to distinguish fraudulent and real reviews. Figure 2(c) shows the highly skewed empirical distribution of this feature.

(iv) *Rating deviation:* Let us take an example of a reviewer whose general trend is to give a low rating to every restaurant he/she reviews without accounting for the ratings given by others. These reviewers should be detected because their ratings deviate from the

restaurant's average ratings. If there is a case with a greater number of real than fraudulent reviewers, chances are high that we will detect instances in which a rating significantly deviates from all the other ratings. We count this metric as an absolute difference between the review score given by a user to a restaurant and an overall average score credited to the restaurant. Figure 2 (d) shows the skewed distribution of this feature, and we formalize it according to Lim et al. [38] as shown in Equation (3):

$$D_a = \frac{1}{N_a} \sum_{b=1}^{N_a} | R_{a,b} - \mu H_b | \quad (3)$$

where N_a = the number of reviews written by a particular reviewer and $R_{a,b}$ = the rating given by a particular reviewer.

(v) *Time of review:* Figure 2(e) shows the highly skewed empirical distribution of the time of review feature, which models the position of a user in the timeline of restaurant reviews. Lim et al. [38] and Mukherjee et al. [44] demonstrated that spammers tend to write reviews early after a product or service is introduced to maximize their impact on a consumer's perception of a product. Thus, we should be alert if we notice a user who always posts restaurant reviews before any other user. We capture this in our model with the use of difference between the time a reviewer reviews a restaurant and the very first review posted for that restaurant in the form of days, as illustrated in Equation (4):

$$Z_a = \frac{1}{N_a} \sum_{b=1}^{N} (T_{a,b} - D_{H(b)}), \quad (4)$$

where N_a = the number of reviews written by the particular reviewer, $T_{a,b}$ = the timestamp for the particular review written by an user, and $D_{H(b)}$ = the timestamp for the first review for a particular restaurant H(b).

(vi) *User tenure:* This feature denotes the amount of time a user is active on an online forum [17, 25]. Fraudulent reviewers generally use short-lived accounts with a comparatively high volume of reviews and handles on a particular e-commerce website to avoid detection via ML algorithms. Hence, to identify fraudulent reviewers, we take the time period in the form of number of days the user is active as a feature. Figure 2(f) shows the highly skewed empirical distribution of this feature, which is represented as shown in Equation (5):

$$Y_a = T_{a,N_a} - T_{a,0}\,, \tag{5}$$

where T_{a,N_a} = the timestamp for the last review by a particular reviewer and $T_{a,0}$ = the timestamp for the first review by a particular reviewer.

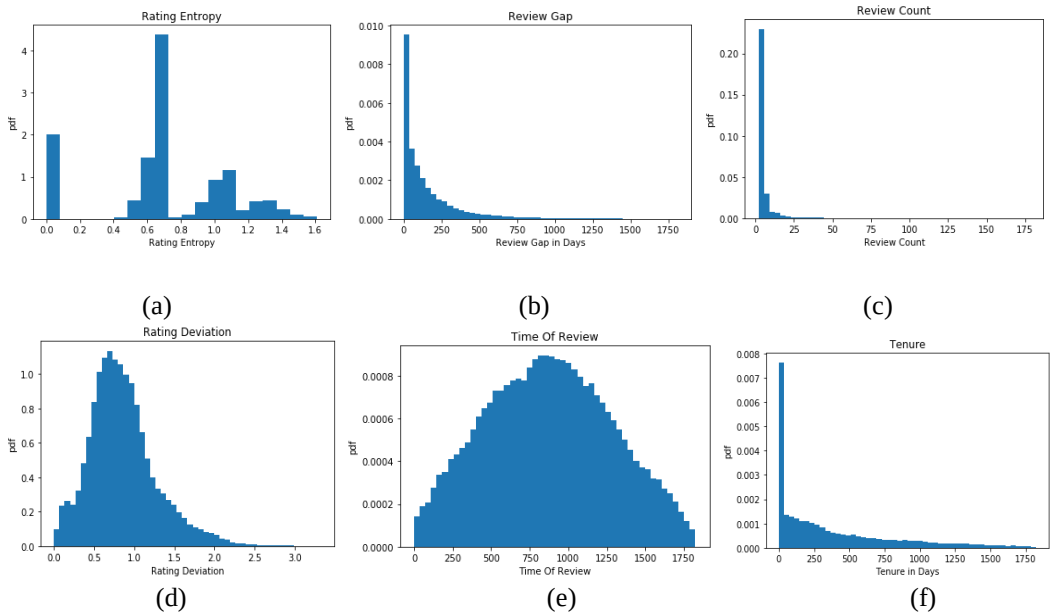


Figure 2. Empirical distributions of reviewer-centric univariate features

Review-centric features

(i) *Review length (number of words):* One observation that we considered is the amount of words put in the reviews. According to Jindal and Liu [23], fraudulent reviewers are likely to write less detailed reviews, whereas a genuine reviewer will write a more detailed review. Unigrams and bigrams are specifically considered in the review text data for counting the total number of unique words. The distribution of users across the average and standard deviations over word-counts is shown in Figure 3(a).

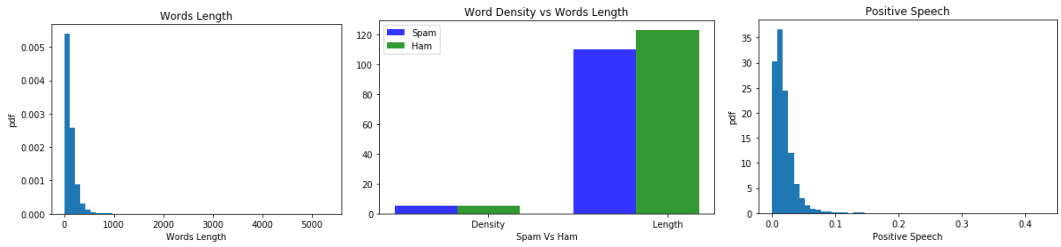
(ii) *Word density:* This important text feature denotes the average length of the words used in each review [23]. Word density is calculated by number of characters divided by number of words in each review. We set the calculation formula for "word density" as (char count)/(word count +1) to avoid division by zero, as some reviews might not have punctuations, and the result is illustrated in Figure 3(b).

(iii) *Part-of-speech ratio:* The main objective of part-of-speech ratio is to extract the linguistic characteristics of fraudulent reviews and identify the grammatical groups of given words. According to Zhang et al. [60], we can count the number of all of these characteristics individually and divide the sum by the word count to assign a corresponding score to the instances.

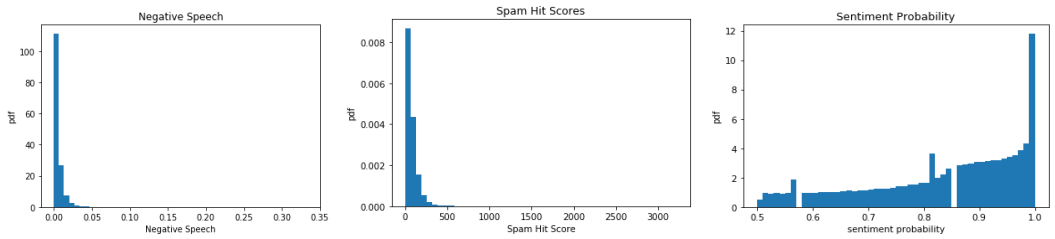
(iv) *Ratio of positive and negative words:* We use the sentiment dictionary to calculate the "ratio of positive words" and "ratio of negative words" and then divide them by the number of words in each review [55]. The highly skewed distribution of this feature can be seen in Figures 3(c) and 3(d).

(v) *SpamHitScore:* Jindal and Liu [23] showed that some companies hire professionals to regularly write fraudulent reviews. In such cases of repeated practice, fraudulent reviewers tend to use particular word or phrase patterns. Figure 3(e) shows the skewed distribution of this feature. To model this suspicious behavior, we use bigram sets with corresponding words & expressions. We use a dictionary to break each review into corresponding N-gram and calculate the score after checking for the presence or absence of a particular word or expression in the fraudulent or real review. The resulting score indicates how much a particular review is different or similar to the spam or fraudulent review.

(vi) *Sentiment probability:* Zhang et al. [60] showed the importance of this variable to predict the fraudulent reviewer' behavior in their research. To calculate the probability of a review (A) being positive or negative based on B (the length of the review), we can use the conditional probability P(A|B) = P(A and B)/P(B). We use the sentiment dictionary to teach the model how a positive/negative review looks based on the words it contains. When we run the classifier on each review, it returns the sentiment (positive/negative) along with a probability of how confident the classifier is while making this decision, the latter of which comprises the sentiment score. Figure 3(f) shows the skewed distribution of this feature.



(a) (b) (c)



(d) (e) (f)

Figure 3. Empirical distribution of review-centric univariate features

Tables 1 and 2 present the detailed summary statistics of reviewer- and review-centric pre-processed features, respectively

Table 1. Summary statistics of reviewer-centric features

Features	**Max**	**Min**	**Mean**	**SD**
Review Count	2	4.64	6.43	165
User Tenure	0	328.37	1613	1613
Rating Deviation	0	0.82	0.44	3.3
Review Gap	0	140.07	211.08	1594
Time of Review	0	745.7	372.14	1611
Rating Entropy	0.45	1.17	0.69	5.08

Table 2. Summary statistics of review-centric features

Features	**Max**	**Min**	**Mean**	**SD**
Review Length	1	117	108	5333
Word Density	0.50	5.40	2.91	810
Part-of-speech Ratio	0	0.08	0.04	0.50
Ratio of Positive words	0	0.02	0.04	0.36
Ratio of Negative words	0	0.01	0.03	0.30
Spam Hit Score	0	72.99	66.85	3224
Sentiment Score	0.50	0.83	0.14	1
Ratio of Nouns	0	0.25	0.07	0.92
Ratio of Verbs	0	0.17	0.05	0.75
Ratio of Adverb	0	0.08	0.04	0.67
Ratio of Adjectives	0	0.11	0.05	0.99

Feature engineering, M-SMOTE algorithm development, hypotheses testing, experimentation, and results

Machine learning often entails working with imbalanced, non-normal, and skewed datasets; however, sending raw or real-world data to the model without processing might result in errors and false results. Feature engineering transforms real-world data into a machine understandable format. Because our dataset is highly imbalanced and we have highly skewed features, we need to solve all data pre-processing challenges before developing the ML models. This work is presented step-by-step in the below subsections.

Step 1: Creation of new variables (extracting new features)

One of the main contributions of this study is to build novel and univariate review- and reviewer-centric features and distributional aspects to detect fraudulent reviewers and their behavior. To illustrate the efficiency and robustness of our proposed approach, reviewers' features and their distributional characteristics are used to address the following questions: (1) Do features from skewed distributions that are directly used in machine learning reduce the accuracy of predictive models? (2) When characterizing overall spammer behavior, what is the specific contribution of each feature? We develop a framework in which univariate features are transformed according to their underlying probability distribution by generalizing the features in accordance with distribution transformations and the whole process is discussed in next sections.

Step 2: Treatment of missing values and outliers

Rather than immediately scaling the data, we first dealt with the outliers. Outliers significantly contribute to skewness; however, some outliers might actually contribute to the model's learning. Considering both the scenarios, we first visualized the outliers for each feature/column with the help of boxplots to glean distortions in the data and then applied log transformation on the complete dataset to help retain novel outliers and enhance the Gaussian distribution. Next, we obtained the interquartile ranges for outlier removal and then used the "RemoveWithValue" filter to removed 36,382 outliers and 10,353 extreme values from the dataset.

Step 3: Feature ranking and dimensionality reduction

In order to gain insights into the relative importance of individual features, we used the model-agnostic feature importance score to determine the univariate importance of each feature with respect to the target variable. As seen in Table 3, our ranking system resulted in reducing the initial 19 features to the top 12 (six review-centric and six reviewer-centric) features that produce the best performance results on predictive models. Next, we used the heat map illustrated in Figure 4 to analyze correlations and multicollinearity among features, whereby darker shades of blue (towards +1) indicate stronger correlations and darker shades of red denote weaker correlations (towards -1). Highly correlated features would require removal, as they might serve similar purposes in training the model. Apart from the main diagonal, we can observe that there are no strong positive or negative correlations among the features.

Figure 4. Heat map-showing correlations between features

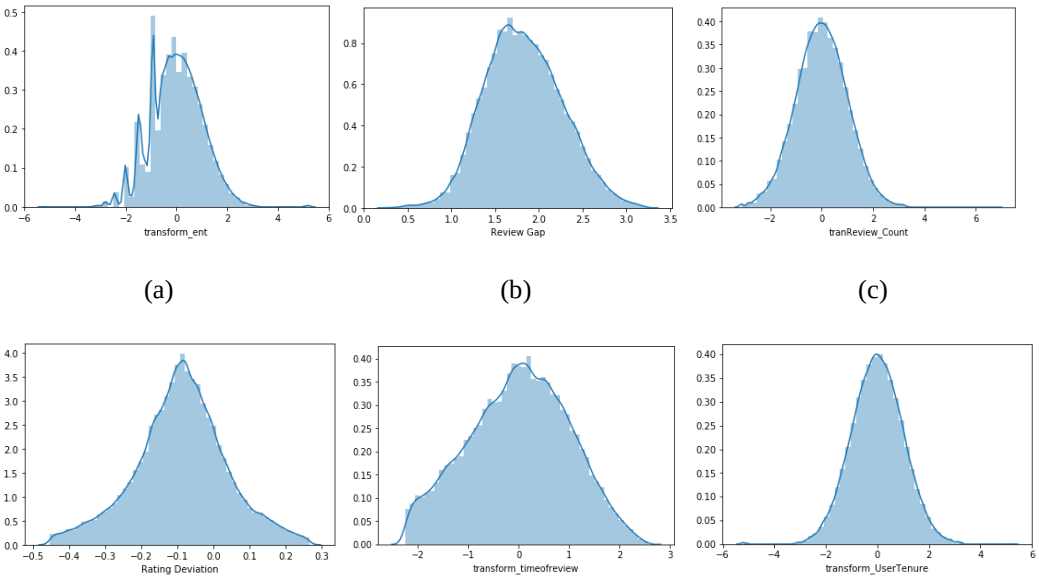
Table 3. Ranking of most important review-centric and reviewer-centric features

Rank	Features	Important Scores
1	Review Count	85.23
2	Review Gap	57.66
3	Review Count	46.55
4	SpamHitScore	42.45
5	Rating Deviation	37.12
6	Review Length	28.56
7	Rating Entropy	24.46
8	User Tenure	22.12
9	Time of Review	14.66

10	Word Density	14.23
11	Sentiment Score	13.49
12	Ratio of Positive words	13.01

Step 4: Operating features individually

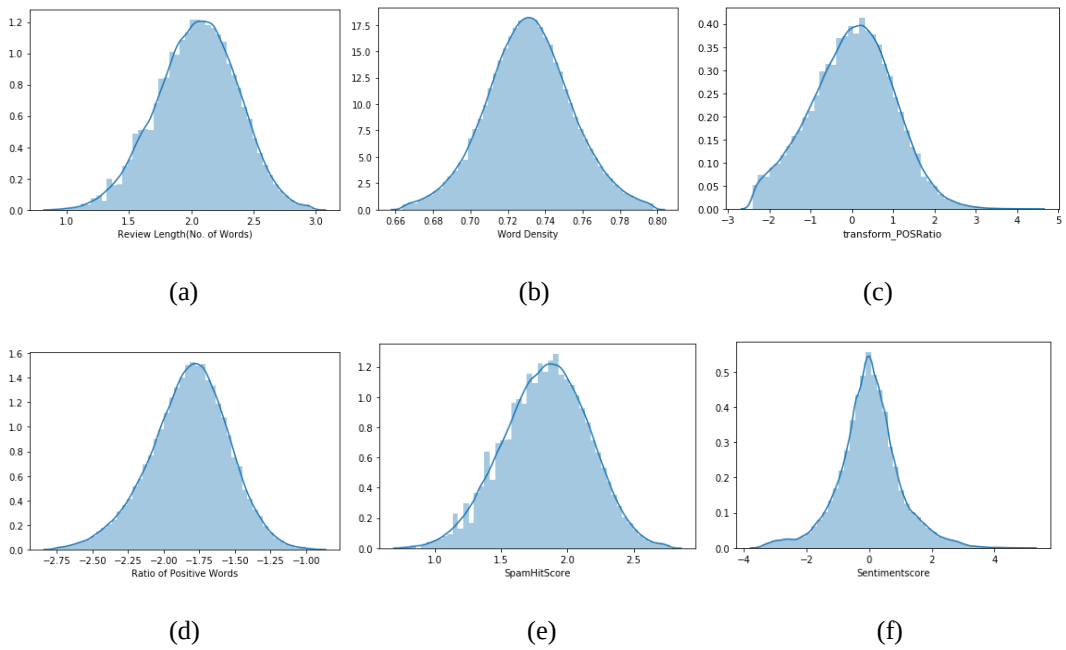
The goal of standardization or normalization is to bring variables to similar scale when comparing measurements across units. There is high chance of bias when variables are with different scales and hence do not contribute equally and fairly to model performance. When features have different ranges, the algorithm's learning rate is determined by the feature with the largest range; thus, scaling the data speeds up the algorithm's training time and improves the overall accuracy of the model. After performing previous steps, the data are normally distributed for half of the features; however, other techniques are needed to achieve bell curves for the remaining features. To achieve normalization, we used min-max scaling and later applied a "yeo-johnson" power transformation for negatively skewed data and a "box-cox" power for positively skewed data. In cases of rigidly skewed data, we applied a multi-step power transformation to bring it within normal distributed range (i.e., -0.5–0.5). Figures 5 and 6 illustrate the transformed reviewer- and review-centric features, respectively. We can see that all features now have a normal distribution, and they can now be used to develop our fraudulent review prediction models.



(a) (b) (c)

(d) (e) (f)

Figure 5. Transformed reviewer-centric features



(a) (b) (c)

(d) (e) (f)

Figure 6. Transformed review-centric features

Step 5: Solving the class imbalance problem

Class imbalance occurs in supervised machine learning when samples or observations in one class are much higher than the other class. ML algorithms are biased towards majority class samples, which they accurately predict, and tend to ignore minority class samples, resulting in significant misclassifications. Three well-known methods to handle class imbalances are SMOTE, undersampling, and oversampling [27]. SMOTE creates "synthetic" samples in minority class rather than by over-sampling with replacement [8]. Undersampling balances class distribution by randomly removing majority class examples, whereas oversampling entails randomly adding minority class samples. With a ratio of 91:9 between majority and minority instances, our Yelp dataset was highly imbalanced. We developed a modified form of SMOTE called M-SMOTE, which gives better results compared with the above-described methods. After applying the M-SMOTE algorithm in the steps elucidated below, the imbalance ratio is reduced to 71:29 instances.

Figure 7 depicts a flowchart of the M-SMOTE algorithm. First, the initial imbalanced dataset is classified by the GaussianProcessClassifier algorithm, and the minority misclassified samples are grouped into a single sample set. The K nearest neighbor samples are generated for each of the misclassified samples. Second, following noise removal, we use the k-means

algorithm to find the center (c) of the misclassified samples. We then calculate the cosine similarity distance d (distance from data point "u" to center "c") from the center sample to each of the minority samples, and calculate the average d_mean of all distances. Next, we calculate the ratio between average d_mean (Average of all "d") and Euclidean distance. Fourth new instances are generated in the following process: (a) We count the values of the neighbor samples of the minority sample (u_s) and save it by number "n" of minority class in the neighbor sample. We can get an idea of the number of minority samples after calculating the value of "n." If the value of "n" is large, then we provide the smaller weight to generate the minority instances; if the value of "n" is small, then the function should generate more samples. (b) The most similar neighbors are again used in generating new instances in the formula shown in Equation (6)

$$t = u + \text{random}(0, M[i]*0.2)*(u\_s - u)$$

$$x\_new = u + \text{random}(0, M[i]*0.8)*(t - c) \quad (6)$$

Where, u = the minority data point, u_s = the most similar neighbor of u, c = the minority cluster center, M[i] = the d/d_mean for a specific data point at index "i," d = the distance from data point "u" to center "c," d_mean = the average of all "d," and x_new = the newly generated instance.

Next, steps 3 and 4 are repeated until we reach the difference of instances between minority and majority instances. We have to remove the newly generated boundary instances until the minority and majority class instances are balanced. Finally, we integrate the newly generated data with the previous dataset to create a final dataset.

LR-SMOTE uses very similar principles to generate a new minority class instance; however, it solves noise generation by first removing the noise from the original dataset and then performing weighted multiplication to oversample the minority class [37]; however, this method does not take care of the noise that it has generated itself while performing weighted multiplication. Our M-SMOTE works in a similar pattern as LR-SMOTE; however, it takes care to minimize the noise generated due to newly generated minority instances, thereby avoiding the production of redundant data. To improve the denoising technique used in LR-SMOTE, we changed the classification algorithm from SVM to GaussianProcessClassifier. SVM's less competent classification could bottleneck the de-noising algorithm and data generation, and GaussianProcessClassifier performs better even in complex and rich features

datasets. In addition, we use the Cosine similarity rather than Euclidean distance to find the most similar neighbors to use for generating new instances.

Unlike LR-SMOTE and SMOTE, in which the newly generated data points will always be on the same line connecting the data points used (u_i) and the center, in our case, the data points will not be on a same line; the most similar neighbor will contribute in the direction, which empowers the data distribution and guarantees a better overall data density that enables easy classification. The LR-SMOTE formula adds a small scaler value to the original point to generate a new point; however, this will only generate points in same direction of a line connecting the center and the original point. To overcome that problem, we consider both a vector value (which gives direction towards a similar looking point) and a scaler value.

For further verification that our M-SMOTE algorithm can be universally applicable on several imbalanced datasets, we tested six datasets from the UCI machine learning repository [54]. As shown in Table 4, our proposed M-SMOTE algorithm performs better than SMOTE according to the precision, recall, F-1, and AUC values. Whether we use the XGBoost or random forest predictive model, the AUC score obtained is higher in the modified dataset.

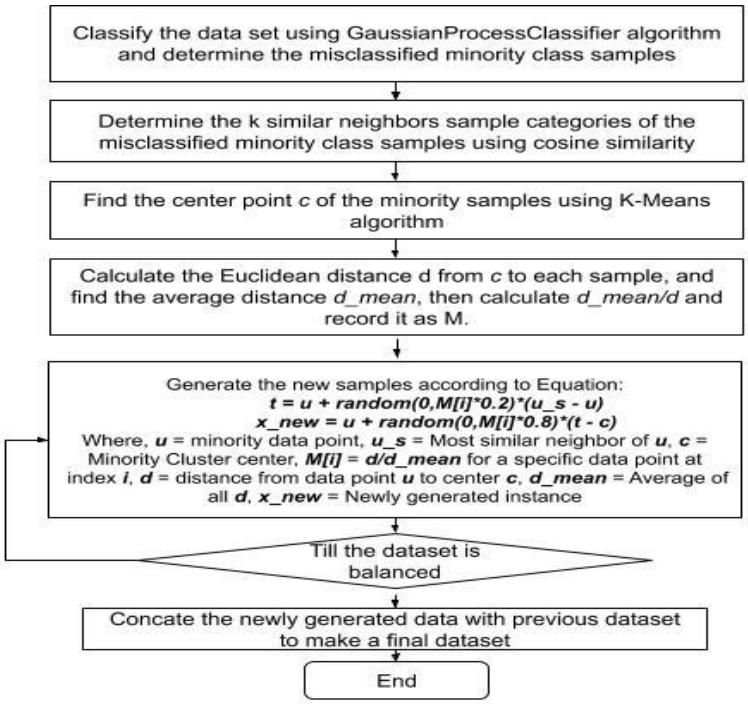


Figure 7. Flow chart of the main steps of M-SMOTE algorithm

Table 4. Value of each evaluation index on UCI, Amazon and Yelp datasets

Dataset	Methods	XGBoost				Random Forest			
		P[a]	R[a]	F-1	AUC	P	R	F-1	AUC
Haberman	None	0.624	0.638	0.695	0.928	0.792	0.807	0.763	0.796
	SMOTE	0.667	0.967	0.944	0.941	0.822	0.826	0.793	0.802
	M-SMOTE	0.811	0.992	0.965	0.961	0.878	0.877	0.881	0.831
Breast cancer	None	0.701	0.803	0.752	0.785	0.613	0.643	0.705	0.818
	SMOTE	0.867	0.804	0.791	0.822	0.674	0.688	0.725	0.868
	M-SMOTE	0.899	0.889	0.876	0.872	0.723	0.739	0.792	0.932
Telecom Churn	None	0.554	0.588	0.605	0.689	0.501	0.511	0.603	0.637
	SMOTE	0.662	0.677	0.694	0.722	0.563	0.578	0.624	0.642
	M-SMOTE	0.794	0.810	0.812	0.822	0.713	0.722	0.733	0.748
Abalone	None	0.954	0.338	0.315	0.958	0.902	0.432	0.395	0.931
	SMOTE	0.917	0.927	0.864	0.911	0.922	0.928	0.873	0.901
	M-SMOTE	0.991	0.982	0.995	0.992	0.962	0.971	0.963	0.944
Amazon	None	0.604	0.648	0.615	0.668	0.534	0.565	0.645	0.602
	SMOTE	0.657	0.687	0.664	0.701	0.598	0.613	0.684	0.665
	M-SMOTE	0.803	0.842	0.885	0.891	0.671	0.744	0.754	0.769
Yelp	None	0.732	0.707	0.721	0.796	0.611	0.572	0.599	0.644
	SMOTE	0.742	0.716	0.733	0.802	0.623	0.593	0.622	0.664
	M-SMOTE	0.772	0.747	0.781	0.838	0.655	0.670	0.690	0.712

a. P = precision; R = recall

Step 6: Model Development

To evaluate the performance of several ML classifiers with and without data pre-processing and feature engineering solutions, we apply the 5-fold (k=5) cross validation procedure, whereby the algorithm randomly chooses four partitions for training purposes whereas the

fifth is used for testing. The algorithm repeats this procedure five times in order to use each partition. We then take the average performance of the particular ML model on each partition and use it to measure the average F-1 score and AUC with the best threshold or hyper parameter for all runs. Below, we analyze the empirical results of applying several data pre-processing techniques and feature engineering tasks in three experimental scenarios.

First scenario- no pre-processing

Table 5 shows the results of performing classification with and without using any feature engineering on the dataset. Compared with the first baseline algorithm (P: 73.2%, R: 70.7%, F-1: 72.1%, and AUC: 79.6%), which used XGBoost with all 12 features extracted from the review text and without any data pre-processing and feature engineering, the performance of the same XGBoost model is significantly better across all four measures after solving all data pre-processing challenges and feature engineering tasks. This result supports our first two hypotheses that feature engineering improves the accuracy of ML classifiers and that features from skewed distributions that are directly used in ML reduce the accuracy of predictive models. The results also support our third hypothesis that data pre-processing and feature-engineering steps improve the accuracy of all ML classifiers.

Table 5. Performance comparison of ML classifiers with and without data pre-processing challenges

	Without pre-processing				**After solving all pre-processing challenges including data imbalance by M-SMOTE algorithm**			
Algorithms	**P**	**R**	**F-1**	**AUC**	**P**	**R**	**F-1**	**AUC**
XGBoost	0.732	0.707	0.721	0.796	0.772	0.747	0.781	0.838
Long short-term memory (LSTM)	0.710	0.698	0.702	0.765	0.740	0.748	0.743	0.809
Light Gradient Boosting Method (GBM)	0.698	0.687	0.666	0.743	0.723	0.710	0.766	0.797
Artificial Neural Network (ANN)	0.677	0.645	0.688	0.736	0.695	0.703	0.754	0.788
Recurrent Neural Network (RNN)	0.644	0.637	0.654	0.731	0.667	0.707	0.733	0.780
SVM (Radial basis function)	0.601	0.572	0.590	0.633	0.678	0.699	0.727	0.793
Logistic Regression	0.554	0.713	0.642	0.727	0.722	0.701	0.723	0.775
k-NN (k=20)	0.546	0.704	0.624	0.704	0.642	0.714	0.721	0.754
Naïve Bayes	0.633	0.587	0.609	0.664	0.682	0.607	0.719	0.732
Random Forest	0.611	0.572	0.599	0.644	0.655	0.670	0.690	0.712

Second Scenario: outliers & extreme value treatment + standardization/scaling+ normalization+ class imbalance treatment → model development using M-SMOTE algorithm

The first part of Table 6 shows that the performance of the same XGBoost model (is significantly increased across all four measures after solving the class imbalance problem using the M-SMOTE technique. The results support the fourth hypothesis that if we solve the class imbalance problem, we can improve the classifier accuracy and model interpretability and decrease the misclassification rate of all ML models.

Table 6. Performance comparison of top three ML classifiers with and without solving class imbalance problem

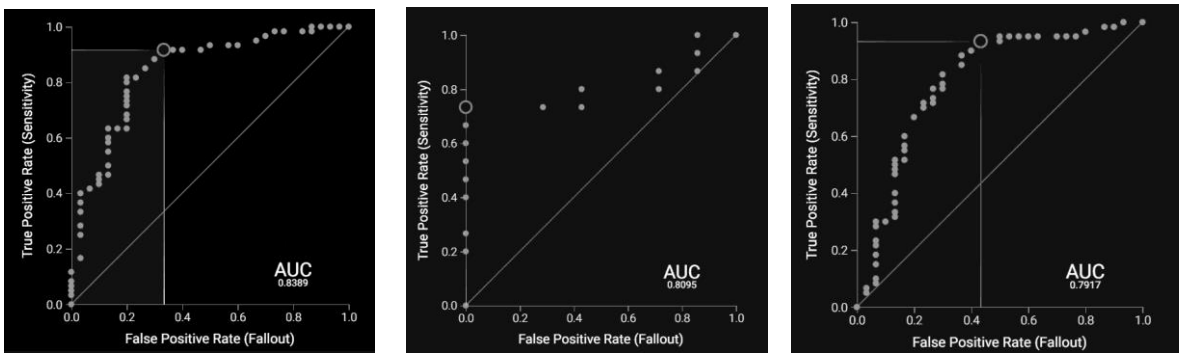
	Without solving class imbalance problem				**After solving all pre-processing challenges including data imbalance by M-SMOTE algorithm**			
Algorithms	**P**	**R**	**F-1**	**AUC**	**P**	**R**	**F-1**	**AUC**
XGBoost	0.741	0.725	0.741	0.810	0.772	0.747	0.781	0.838
LSTM	0.723	0.698	0.702	0.765	0.740	0.748	0.743	0.809
Light GBM	0.708	0.707	0.683	0.761	0.723	0.710	0.766	0.797

We next present the result of classifying fraudulent reviewers using our proposed model that applies feature engineering and majority voting (Table 7). In this stage, we develop a number of machine learning models for identifying the fraudulent reviewers from the dataset after solving all data pre-processing challenges & feature engineering tasks and then use the case of majority voting on the best three machine learning classifiers' performance. Table 5 shows that score of XGBoost, LSTM and Light GBM models provide the best results on the Yelp dataset, so we select these, apply the majority voting rule on them, and calculate the accuracy of the final predictive model. We apply a simple rule of hard voting for predicting the class label Y via majority voting of three best classifiers [Y = mode{M1(X),M2(X),M3(X)}]. For example, if our three best machine learning classifiers give the prediction results as follows: if Classifier1 = Fraudulent (1), Classifier2 = True (0) and Classifier3 = True (1) then we would use the majority vote concept and the final prediction result would be Y= mode {1, 0, 1} =1. This would be the fourth prediction result, and we would consider it final output of our ensemble vote classifier in our proposed framework for detecting the fraudulent reviews. Figures 8(a)-(c) show the three models' receiver operating characteristics (ROC) curves. The ROC curve tells us to choose a threshold value that balances sensitivity, or the true positive rate (TPR), and specificity, or the true negative tare (TNR), in a way that makes sense for our

particular case. The area under curve (AUC) measures the predictive models' capacity to separate fraudulent and real reviewer classes. A perfect model has an AUC of one. The left side of Table 7 shows that the XGBoost has the best chance of being able to distinguish between fraudulent review and real review class on the Yelp dataset, followed by the LSTM and Light GBM models, respectively. The right side of Table 7 shows the results of the last majority vote step of our proposed fraudulent review detection model. The overall results show an improvement of 2-6 % with our proposed model compared to the baseline approaches.

Table 7. Final model results

	After solving all pre-processing challenges (including data imbalance)				**After applying majority vote rule Y= mode {0, 1, 1} =1**			
Algorithms	**P**	**R**	**F-1**	**AUC**	**P**	**R**	**F-1**	**AUC**
XGBoost	0.772	0.747	0.781	0.838	0.794	0.835	0.853	0.874
LSTM	0.720	0.738	0.733	0.809				
Light GBM	0.723	0.710	0.786	0.797				



(a) (b) (c)

Figure 8. ROC curve of best 3 models (XGboost, LSTM and light GBM)

To further validate our hypotheses (H1 and H2) concerning the relative importance of reviewer-centric features vs review-centric features and the benefits of combining the features, we test the combined effect of all features for improving the accuracy of all machine learning classifiers, we repeated the same set of data pre-processing and feature engineering tasks on both datasets. Table 8 shows that the results of reviewer-centric features are

significantly better than those of the review-centric features across all four measures after separately applying the same machine learning algorithms on both datasets. Thus, H1 is supported across all statistical measures.

Table 8. Performance of ML models with review-centric vs. reviewer-centric features

	Reviewer-centric Features				**Review-centric Features**			
Algorithms	**Precision**	**Recall**	**F-1**	**AUC**	**Precision**	**Recall**	**F-1**	**AUC**
XGBoost	0.704	0.721	0.741	0.806	0.675	0.647	0.702	0.733
LSTM	0.730	0.702	0.762	0.782	0.623	0.658	0.673	0.708
Light GBM	0.705	0.694	0.766	0.765	0.626	0.660	0.657	0.681
Random Forest	0.682	0.667	0.698	0.743	0.593	0.623	0.650	0.664
Logistic Regression	0.652	0.643	0.692	0.720	0.564	0.677	0.637	0.643

To further validate our next hypothesis (H2), Table 8 results show the performance of top five ML models developed using all features (review- and reviewer-centric) and review-centric features, and Table 10 shows the t-test results comparing the accuracy of the top five ML classifiers when combining review- and reviewer-centric features over reviewer-centric features alone. Both tables demonstrate that incorporating both review- and reviewer's behavior features improved all ML classifiers performance in terms of precision, recall, AUC score and F-1 scores compared with using review-centric features alone in the same logistic regression model. We can clearly see that the performance of the same logistic regression model is significantly increased across all four measures after incorporating review-centric features with reviewer-centric features in developing the fraudulent review detection models. Therefore, H2 is supported across all statistical measures.

Table 9. Performance of ML models with all features vs. reviewer-centric features

	Reviewer-centric + Review-centric Features (%)	**Reviewer-centric Features (%)**

Algorithms	Precision	Recall	F-1	AUC	Precision	Recall	F-1	AUC
XGBoost	0.772	0.747	0.781	0.836	0.704	0.721	0.741	0.806
LSTM	0.730	0.738	0.733	0.804	0.726	0.702	0.762	0.782
Light GBM	0.723	0.710	0.786	0.794	0.705	0.694	0.766	0.765
Random Forest	0.655	0.670	0.690	0.712	0.682	0.667	0.698	0.743
Logistic Regression	0.722	0.701	0.723	0.775	0.652	0.643	0.692	0.720

Table 10. Performance improvement after incorporating reviewer-centric features (mean difference)

Algorithms	Precision (%)	Recall (%)	F-1 (%)	AUC (%)
XGBoost	+6.8***	+2,6***	+4%***	+3.0***
LSTM	+0.4	+3.6	-2.9	+2.2
Light GBM	1.8	+1.6	+2.0	+2.9
Logistic Regression	+7.0	+5.8	+3.1	+5.5
*** denote the significance at .001 level.				

In summary, the analysis leads to the following conclusions:

We can verify that positive reviews (5-star ratings) tend to be lengthier and more detailed, and exhibit a high rate of misspelled words and incorrectly used apostrophes. Negative reviews (1-star ratings) are typically shorter and often do not include any punctuation. Furthermore, fraudulent reviewers use more content and more verbs, adjectives, and filler words than real reviewers do, whereas real reviews contain more nouns and pronouns (Figure 9). We have also identified positive reviews generally contain longer and more error-filled prose than negative reviews (Figure 10)

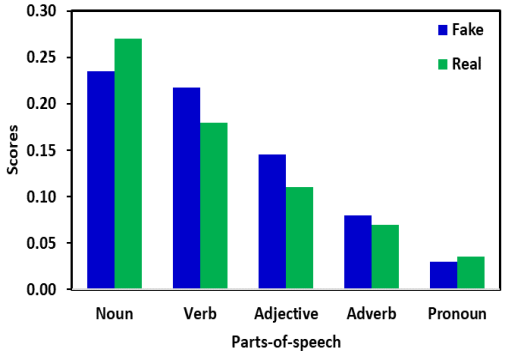


Figure 9. Part-of-Speech Ratios

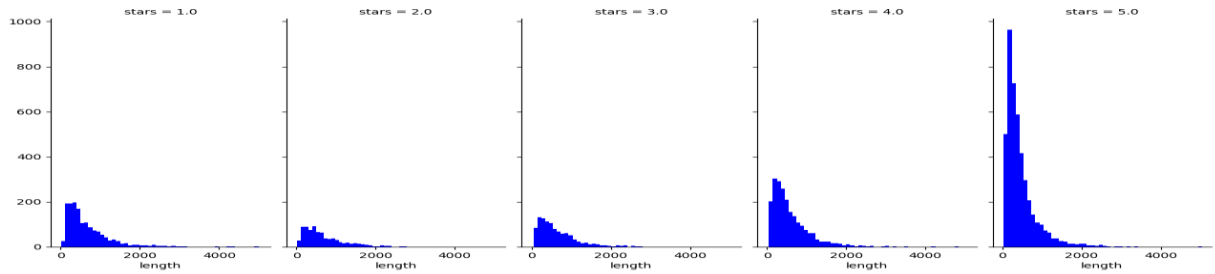
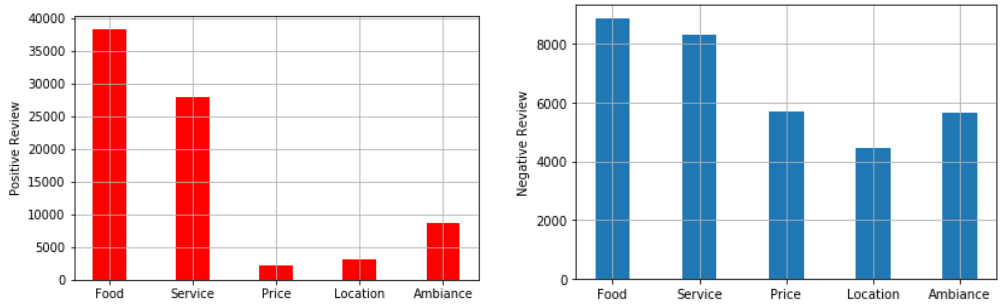


Figure 10. Star rating vs character length

Figure 11 shows that “food” and “service” are among the top words across both positive and negative reviews, which indicates that these are the two most important dimensions that reviewers consider when they visit and evaluate a restaurant.



(a) (b)

Figure 11. Predominant categories in positive and negative reviews

We can also see that reviewers tend to most frequently express positive views, as the average of positive words triple that of negative words across all reviews. Initially, we assumed that one- and five-star ratings would have the longest reviews because when a reviewer is very disappointed or very satisfied with a particular restaurant, they might write a relatively long review to express their strong emotions. However, 5-star reviews are in fact lengthier than 1-star reviews, although the latter do tend to be longer that 2- and 3-star reviews. Otherwise, as the star rating decreases, so do the average number of words in the review.

Our assumption that directly using skewed features in ML models would reduce their predictive accuracy was verified by our analysis, as was our hypotheses that reviewer-centric features of users are more important that review-centric features for detecting fraudulent reviewer’s behavior and that combining reviewer-centric with review-centric features

improves the accuracy of all ML detection models in comparison with using review-centric features alone. Finally, our analysis verifies that if we want to introduce new features that are not correlated with each other, then the adjusted R-squared will increase only if the introduced feature improves the model.

Methodology validation and performance comparison with prior studies

We can now validate our methodology on other three datasets from Yelp and Amazon website. We performed and repeated the same set of feature engineering and data pre-processing steps on two different Amazon datasets [12, 23] as well as the Yelp Open Dataset [57]. Table 11 shows the performance of the top four ML models with and without feature engineering on the Amazon dataset, all of which are largely consistent with those based on our original Yelp review dataset. After solving all data pre-processing and feature engineering tasks and applying our proposed M-SMOTE model, all of the classifiers perform better than the raw data on all four datasets. We can clearly see that the performance of the same random forest model is significantly better and increased across all four measures after solving all data pre-processing challenges and feature engineering tasks before feeding it into our proposed model. Therefore, these predictive results confirm the findings from the original Yelp restaurant review results.

Table 10. Comparison of ML classifiers' performance on the validation dataset

	After solving all pre-processing challenges (including data imbalance)				**After applying majority vote rule Y= mode {0, 1, 1} =1**			
Model	**Precision**	**Recall**	**F-1**	**AUC**	**Precision**	**Recall**	**F-1**	**AUC**
XGBoost	0.803(+3.3 %)	0.842(+4. 6%)	0.885(+ 5.2%)	0.891(+7 .8%)	0.812	0.869	0.902	0.937
LSTM	0.692(+9.3 %)	0.775(+6. 8%)	0.794(+ 6.2%)	0.821(+1 1.3%)				
ANN	0.683+(+4. 7%)	0.751(+5. 3%)	0.782(+ 6.2%)	0.776(+6 .6%)				
Random Forest	0.671+(+1 1.3%)	0.744(+9. 1%)	0.754(+ 7.2%)	0.769(+1 1.8%)				

We also compare the results of our proposed model with several other notable preevious studies conducted on the Amazon and Yelp datasets. Jindal and Liu [23] used the Amazon

dataset to train their predictive models and showed an AUC score of 78%. They did not used any feature engineering process on the dataset and used the raw data directly to develop the machine learning models. Feng et al [16] derived probabilistic context- free grammars from Yelp and TripAdvisor datasets to develop the fraudulent review detection models; however, both datasets were quite small (around 800-900 reviews). ML classifier's performance are strongly affected by data-set size and the noise contained in the training set, which is likely why they achieved only 64.3 % accuracy on Yelp dataset. Zhang et al [60] used verbal and non-verbal features on an Amazon dataset but did not improve ML classifier accuracy in any way or perform any data pre-processing and feature engineering tasks on the dataset before feeding it into machine learning. The only feature-engineering task that Kumar et al. [28] worked on was the "skewed distribution" of features before developing the supervised ML models on the Yelp dataset, and they obtained an 81.7% AUC score in the logistic regression model. Rayana and Akoglu [51] and Kumar et al. [29] used the Yelp dataset and combined metadata and text features to develop the unsupervised ML models. As previously described, Rayana and Akoglu [51] performed traditional text pre-processing solutions before developing their unsupervised ML models and obtained a 68.28% AUC score. Kumar et al [29] incorporated two feature engineering tasks, namely "skewed distributions of features" and "outlier detection and removal" and used a mixture model for detecting the fraudulent reviews, ultimately obtaining a 70% AUC score.

In comparison with the above, our proposed model achieves a greater than 80% AUC score on all four datasets, and on the main Yelp dataset we achieve particularly high scores across all four measures (P: 79.4%, R: 83.5%, F-1: 85.3%, and AUC: 87.4%). Clearly, our proposed fraudulent review prediction model outperforms the above-mentioned methods.

Discussions, conclusion, limitations and future work

In this paper, we have proposed an M-SMOTE-based ML framework to develop a fraudulent review detection model that can assist marketing managers and consumers to detect the opinion spammers and their suspicious behavior in the decision-making process. Although several other models to detect fraudulent reviewers have been developed, none of these fully exploits feature engineering, data pre-processing, or the underlying distributional characteristics of reviewers' behavior. Our model combines various heterogeneous distributions, pre-processing tasks, and feature engineering on a balanced dataset in order to holistically explain different characteristics of fraudulent reviewers' behavior.

Our analysis of the correlation between ratings and the results of fraudulent reviewers' sentiments revealed that the most positive reviews tend to be longer and more detailed and have a higher rate of misspelled words and incorrectly used apostrophes, whereas the most negative reviews are typically shorter in length and often do not include any punctuation. We have also identified that positive reviews contain longer and more error-filled prose than negative reviews. Moreover, whereas fraudulent reviews contain more content, verbs, adjectives, and filler words than real reviews, the latter contain more nouns and adjectives. During several experiments, we have: (1) derived univariate distributions of features that are commonly used to characterize reviewer's behavior; (2) performed feature engineering to transform the features and delete unnecessary entries from the dataset; (3) developed an M-SMOTE algorithm to solve the class imbalance problem; and (4) incorporated univariate distributions into a machine-learning model to detect fraudulent reviewers and their behavior. We have used both synthetic and real-world Yelp restaurant review data to compare our methodology with the traditional ML algorithms and other state-of-the-art supervised methods for detecting fraudulent reviewers and demonstrated that our method outperforms other approaches.

The present findings confirm that combining reviewer-centric features with review-centric features can significantly improve the performance of fraudulent review detection models. Importantly, our results provide evidence that reviewer-centric features can be more effective than review-centric features for detecting the opinion spammers, as reviewer-centric features are complementary to reviewers' behaviors, which deserve more attention when developing predictive models. We believe that the strong effect of reviewer-centric features is due to the likelihood that it is more difficult and costly to manipulate reviewers' behavior, which makes it a more robust and reliable cue to detect opinion spamming.

Our research model has several additional implications for online e-commerce. As social media platforms have become major conduits of fraudulent information, automated social media accounts are increasingly likely to increase the spread of fraudulent news and reviews [33]. Our proposed solution handles biases that make social media websites vulnerable to misinformation by offering an M-SMOTE fraudulent detector tool. Our proposed novel supervised ML method can be extended to categorize the fraudulent accounts of social bots, and ecommerce companies can use it to redesign their existing predictive model and combat the automated spread of fraudulent news content. Our findings suggest that online e-commerce platforms should encourage social interactions between customers and reviewers

in order to supply rich evidence of reviewers' behavior to enhance the detection of opinion spamming.

This research has some limitations that might restrict the generalization of the findings and provide potential opportunities for further research. First, the proposed model has only been tested on three different datasets from two platforms. Although Yelp and Amazon are very popular online review platforms and contain rich collections of the social activities of individual reviewers, different results might have been obtained if the model had been tested on other platforms such as Walmart, Alibaba, eBay, or TripAdvisor. The findings might not be directly applicable on other datasets because some novel features included in our predictive models may not be available. To overcome these limitations, further research should certainly examine the role of additional features of fraudulent reviewers' behavior, and the framework should be tested on multiple datasets collected from additional online platforms. ML models can produce false signals when fraudulent reviewers dynamically change their behavior; thus, if we apply the same features on new datasets, there is a high risk of the model losing credibility. However, we suggest that the features developed in this study could be generalized to other online e-commerce platforms, which is worthy of further investigation and future research. Moreover, we plan to more deeply analyze review-centric information in order to extract additional novel features that could contribute to improve the accuracy and interpretability of machine learning models.

References

1. Abbasi, A., Zahedi, F.M., Zeng, D., Chen, Y., Chen, H.and Nunamaker J.F. Enhancing predictive analytics for anti-phishing by exploiting website genre information. *Journal of Management Information Systems*, 31, 4 (2015), 109–157.

2. Abbasi A, Zhang, Z. Zimbra, D., Chen, H. and Nunamaker, J.F. Detecting fraudulent websites: the contribution of statistical learning theory. *MIS Quarterly*, 34, 3 (2010) 435–461.

3. Akoglu, L., Chandy, R., and Faloutsos, C. Opinion fraud detection in online reviews by network effects. *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 7, (2013), 2–11.

4. Arazy, O. and Woo, C. Enhancing information retrieval through statistical natural language processing: a study of collocation indexing. *MIS Quarterly*, 31, 3 (2007), 525-546.

5. Ball, L. and Elworthy, J. Fraudulent or real? The computational detection of online deceptive text. *Journal of Marketing Analytics*, 2 (2014), 187–201.

6. Banerjee S. and Chua, A.Y. A theoretical framework to identify authentic online reviews. *Online Information Review*, 38, 5 (2014) 634–649.

7. Bhutada, S., Balaram, V.V.S.S.S. and Sree, C.A.Deep learning framework to detect the false analysis of a product given by robots and malicious users. *International Journal of Innovative Technology and Exploring Engineering*, 8, 6 (2019), 689–692.

8. Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16 (2002), 321–357.

9. Chen, L.Y. Alibaba lawsuit throws the spotlight on fraudulent reviews. *Bloomberg News*, December 19, 2016. Available from: https://www.bloomberg.com/news/articles/2016-12-19/alibaba-lawsuit-throws-spotlight-on-brushers-gaming-rankings

10. Conner, C. Amazon sues 1,114 fraudulent reviewers on Fiverr. *Forbes*, October 8, 2015. Available from: https://www.forbes.com/sites/cherylsnappconner/2015/10/18/amazon-sues-1114-fraudulent-reviewers-on-fiverr-com/#5d06fc3637d1

11. Dalvi, N.N., Kumar, R. and Pang, B. (2013). Para 'normal' activity: on the distribution of average ratings. *Proceeding of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM-13).* Palo Alto: The AAAI press, pp. 110-119

12. Deception-Detection-on-Amazon-reviews-dataset. Available from: https://github.com/aayush210789/Deception-Detection-on-Amazon-reviews-dataset

13. Deng, S. Deceptive reviews detection of industrial product. *International Journal of Services Operations and Informatics*, 8, 2 (2016), 122.

14. Dong, L., Ji, S., Zhang, C., Zhang, Q., Chiu, D.W., Qiu, L. and Li, D. An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Systems with Applications*, 114 (2018), 210–223.

15. Federal Trade Commission. FTC brings first case challenging fraudulent paid reviews on an independent retail website. February 26, 2019. Available at https://www.ftc.gov/news-events/press-releases/2019/02/ftc-brings-first-case-challenging-fraudulent-paid-reviews-independent

16. Feng, S., Banerjee, R. and Choi, Y. Syntactic stylometry for deception detection. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2012, pp. 171–175.

17. Goes, P.B.; Lin, M.; and Au Yeung, C.M. Popularity effect" in user-generated content: Evidence from online product reviews. *Information Systems Research*, 25, 2 (2014), 222–238.

18. Hazim, M., Anuar, N.B., Ab Razak, M.F. and Abdullah, N.A. Detecting opinion spams through supervised boosting approach. *PLoS One*, 13, 6 (2018), e0198884.

19. Heredia, B., Khoshgoftaar, T.M., Prusa, J.D. and Crawford, M. Improving detection of untrustworthy online reviews using ensemble learners combined with feature selection. *Social Network Analysis and Mining*, 7, 37 (2017), 1–18.

20. Ho, S.M., Hancock, J.T., Booth, C., and Liu, X. Computer-mediated deception: Strategies revealed by language-action cues in spontaneous communication. *Journal of Management Information Systems*, 33, 2 (2016), 393–420.

21. Ivanova, O. and Scholz, M. How can online marketplaces reduce rating manipulation? A new approach on dynamic aggregation of online ratings. *Decision Support Systems*, 104, 4 (2017), 64–78.

22. Jalther, D. and Priya, G. Reputation reporting system using text based classification, *International Journal of Innovative Technology and Exploring Engineering*, 8 (8) (2019) 1555–1558.

23. Jindal, N. and Liu, B. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Data Mining*. New York, NY: ACM, 2008, pp. 219–230.

24. Kamalesh, M.D. and Diwedi, H.K. Extracting product features from consumer reviews and its applications. *International Journal of Applied Engineering Research*, 10, 2 (2015), 2345–2350.

25. Khansa, L., Ma, X., Liginlal, D. and. Kim, S.S. Understanding members' active participation in online question-and-answer communities: a theory and empirical analysis. *Journal of Management Information Systems*, 32,2 (2015), 162-203.

26. Khurshid F., Zhu, Y. Xu, Z. Ahmad, M. and Ahmad, M. Enactment of ensemble learning for review spam detection on selected features. *International Journal of Computational Intelligence Systems*, 12 (1) (2019) 387–394.

27. Kumar A., Shankar, R., Choudhary, A.K. and Thakur, L.S. A big data MapReduce framework for fault diagnosis in cloud-based manufacturing. *International Journal of Production Research*, 54 (2016) 7060–7073.

28. Kumar, N., Venugopal, D., Qiu, L., and Kumar, S. Detecting review manipulation on online platforms with hierarchical supervised learning. *Journal of Management Information Systems*, 35, 1 (2018), 350–380.

29. Kumar, N., Venugopal, D., Qiu, L., and Kumar, S. Detecting anomalous online reviewers: an unsupervised approach using mixture models. *Journal of Management Information Systems*, 36, 4 (2019), 1313–1346.

30. Lappas, T., Sabnis, G. and Valkanas, G. The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry. *Information Systems Research*, 27, 4, (2016), 940-961.

31. Lash, M.T. and Zhao, K. Early predictions of movie success: The who, what, and when of profitability. *Journal of Management Information Systems*, 33, 3 (2016), 874–903.

32. Lau, R.Y.K., Liao, S.Y., Kwok, R.C.W., Xu, K., Xia, Y. and Li, Y. Text mining and probabilistic language modeling for online review spam detection. *Transactions on Management Information Systems*, 2, 4 (2011), 1–30.

33. Lazer, D.M., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F. and Schudson, M. The science of fraudulent news. *Science*, 359, 6380 (2018), 1094–1096.

34. Li, F., Huang, M., Yang, Y., and Zhu, X. Learning to identify review spam. *Proceedings of the International Joint Conference on Artificial Intelligence*, 22, 3 (2011), 2488–2493.

35. Li, H., Liu, B., Mukherjee, A. and Shao, J. Spotting fraudulent reviews using positive-unlabeled learning. *Computations y Sistemas*, 18, 3 (2014), 467–475.

36. Li, L., Qin, B., Ren, W. and Liu, T. Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, 254 (2017), 33–41.

37. Liang, X., Jiang, A., Li, T., Xue, Y. and Wang, G. LR-SMOTE—An improved unbalanced dataset oversampling based on K-means and SVM. *Knowledge Based Systems* 196 (2020), 105845.

38. Lim, E. Nguyen, V. A., Jindal, N., Liu, B., and Lauw, H. Detecting product review spammers using rating behavior. *Proceedings of the 19th ACM international conference on Information and knowledge management*. Lyon: ACM, 2010, pp. 939–948.

39. Lin, Y., Zhu, T., Wu, H., Zhang, J., Wang, X., and Zhou, A. Towards online anti-opinion spam: spotting fraudulent reviews from the review sequence. *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. New York: Association for Computing Machinery, 2014, pp. 261–264.

40. Liu, Y. and Pang, B. A unified framework for detecting author spamicity by modeling review deviation. *Expert Systems With Applications*, 112 (2018), 148–155.

41. Liu Y., Pang, B., and Wang, X. Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. *Neurocomputing*, 366 (2019) 276–283

42. Luca, M. and Zervas, G. Fake it till you make it: reputation, competition, and yelp review fraud. *Management Science*, 62,12 (2016), 3412–3427.

43. Ludwig, S., Van Laer, T., De Ruyter, K. and Friedman, M. Untangling a web of lies: Exploring automated detection of deception in computer-mediated communication. *Journal of Management Information Systems*, 33, 2 (2016), 511–541.

44. Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M. and Ghosh, R. Spotting opinion spammers using behavioral footprints. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York: ACM, 2013, pp. 632-643.

45. Mukherjee, A., Liu, B. and Glance, N. Spotting fake reviewer groups in consumer reviews. *Proceedings of the 21st International World Wide Web Conference* (IW3C). Lyon: ACM, 2012, pp. 191-200.

46. Murphy, R. Local consumer review survey. December 2019. Available from: https://www.brightlocal.com/research/local-consumer-review-survey/

47. Ott, M., Choi, Y., Cardie, C. and Hancock, J. T. (2011). 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*. Stroudsburg: Association for Computational Linguistics, pp. 309–319.

48. Proudfoot, J.G.; Jenkins, J.L.; Burgoon, J.K.; and Nunamaker, J.F. Jr. More than meets the eye: how oculometric behaviors evolve over the course of automated deception detection interactions. Journal of Management Information Systems, 33, 2 (2016), 332–360.

49. Rajamohana, S. and Umamaheswari, K. Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection. *Computers & Electrical Engineering*, 67 (2018), 497–508.

50. Rajamohana, S., Umamaheswari, K. and Abirami, B. Performance analysis of iBPSO and BFPA based feature selection techniques for improving classification accuracy in review spam detection. *Applied Mathematics & Information Sciences*, 11, 4 (2017) 1149–1153.

51. Rayana, S. and Akoglu, L. Collective opinion spam detection: bridging review networks and metadata. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 21, (2015), 985–994.

52. Ren, Y. and Ji, D. Neural networks for deceptive opinion spam detection: an empirical study, *Information Sciences*, 385 (2017), 213–224.

53. Siering, M., Koch, J.A. and Deokar, A.V. Detecting fraudulent behavior on crowd platforms: The role of linguistic and content-based cues in static and dynamic contexts. *Journal of Management Information Systems*, 33, 2 (2016), 421–455.

54. UCI machine learning repository. Available from: https://archive.ics.uci.edu/ml/datasets.php

55. Vanta, T. and Aono, M. Fraudulent review detection focusing on emotional expressions and extreme rating. *Proceedings of the 25th Annual Conference of the Language Processing Society (NLP2019)*. Nagoya, Japan: The Association for Natural Language Processing, 2019, pp. 1-30.

56. Xu, Y., Yang, Y., Han, J., Wang, E., Ming, J. and Xiong, H. Slanderous user detection with modified recurrent neural networks in recommender system. *Information Sciences*, 505 (2019), 265–281.

57. Yelp open dataset. Available from https://www.yelp.com/dataset/

58. Yu, C., Zuo, Y., Feng, B., and Chen, B. An individual-group-merchant relation model for identifying fraudulent online reviews: an empirical study on a Chinese e-commerce platform. *Information Technology & Management*, 20 (2019) 123–138.

59. Yuan, S., Wu, X. and Xiang, Y. Task-specific word identification from short texts using a convolutional neural network. *Intelligent Data Analysis*, 22, 3 (2018), 533–550.

60. Zhang, D.S., Zhou, L.N., Kehoe, J.L. and Kilic, I.Y. What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fraudulent online reviews. *Journal of Management Information Systems* 33, 2 (2016), 456–481.

61. Zhang, L., Wu, Z. and Cao, J. Detecting spammer groups from product reviews: a partially supervised learning model. *IEEE Access*, 6 (2018), 2559–2568.

62. Zhang, W., Du, Y., Yoshida, T. and Wang, Q. DRI-RCNN: an approach to deceptive review identification using recurrent convolutional neural network. *Information Processing and Management*, 54, 4 (2018), 576–592.