Experience: Improving Opinion Spam Detection by Cumulative Relative Frequency Distribution

# Experience: Improving Opinion Spam Detection by Cumulative Relative Frequency Distribution

MICHELA FAZZOLARI, Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Italy
FRANCESCO BUCCAFURRI, DIIES, Università Mediterranea di Reggio Calabria, Italy
GIANLUCA LAX, DIIES, Università Mediterranea di Reggio Calabria, Italy
MARINELLA PETROCCHI, Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Italy

Over the last years, online reviews became very important since they can influence the purchase decision of consumers and the reputation of businesses, therefore, the practice of writing fake reviews can have severe consequences on customers and service providers. Various approaches have been proposed for detecting opinion spam in online reviews, especially based on supervised classifiers. In this contribution, we start from a set of effective features used for classifying opinion spam and we re-engineered them, by considering the Cumulative Relative Frequency Distribution of each feature. By an experimental evaluation carried out on real data from Yelp.com, we show that the use of the distributional features is able to improve the performances of classifiers.

CCS Concepts: • **Computing methodologies** → *Supervised learning by classification*; • **Networks** → Online social networks; • **Information systems** → *Information extraction*; **Data cleaning**.

Additional Key Words and Phrases: Trustworthiness in Social Media, Online Reviews Analysis, Supervised Classification Models, Yelp

**ACM Reference Format:**
Michela Fazzolari, Francesco Buccafurri, Gianluca Lax, and Marinella Petrocchi. 2020. Experience: Improving Opinion Spam Detection by Cumulative Relative Frequency Distribution. 1, 1 (November 2020), 16 pages. https://doi.org/xxx/yyy

## 1 INTRODUCTION

As illustrated in a recent survey on the history of Digital Spam [18], the Social Web has led not only to a 'participatory, interactive nature of the Web experience', but also to the proliferation of new and widespread forms of spam, among which the most notorious ones are fake news and spam reviews, *a.k.a.* opinion spam. This results in the diffusion of different kinds of disinformation and misinformation, where misinformation refers to inaccuracies that may even originate acting in good faith, while disinformation is false information deliberately spread to deceive [20].

Over the last years, online reviews became very important since they reflect the customers' experience with a product or service and, nowadays, they constitute the basis on which the reputation of an organization is built. Unfortunately,

Authors' addresses: Michela Fazzolari, Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Via G. Moruzzi, 1, Pisa, Italy, m.fazzolari@iit.cnr.it; Francesco Buccafurri, DIIES, Università Mediterranea di Reggio Calabria, Via Graziella, Località Feo di Vito, Reggio Calabria, Italy, bucca@unirc.it; Gianluca Lax, DIIES, Università Mediterranea di Reggio Calabria, Via Graziella, Località Feo di Vito, Reggio Calabria, Italy, lax@unirc.it; Marinella Petrocchi, Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Via G. Moruzzi, 1, Pisa, Italy, m.petrocchi@iit.cnr.it.


Manuscript submitted to ACM

the confidence in such reviews is often misplaced, due to the fact that spammers are tempted to write fake information in exchange for some reward or to mislead consumers for obtaining business advantages [28].

The practice of writing false reviews is not only morally deplorable, as it is misleading for customers and inconvenient for service providers, but it is also punishable by law. Considering both the longevity and the spread of the phenomenon, scholars for years have investigated various approaches to opinion spam detection, mainly based on supervised or unsupervised learning algorithms. Further approaches are based on Multi Criteria Decision Making [42].

Machine learning approaches rely on input data to build a mathematical model in order to make predictions or decisions. To this aim, data are usually represented by a set of features, which are structured and ideally fully representative of the phenomenon being modeled. An effective feature engineering process, i.e., the process through which an analyst uses the domain knowledge of the data under investigation to prepare appropriate features [11], is a critical and time-consuming task. However, if done correctly, feature engineering increases the predictive power of algorithms by facilitating the machine learning process.

In this paper, we do not aim to contribute by defining novel features suitable for fake reviews detection, rather, starting from features that have been proven to be very effective by Academia, we *re-engineered* them, by considering the distribution of the occurrence of the features values in the dataset under analysis. In particular, we focus on the Cumulative Relative Frequency Distribution of a set of the basic features already employed for the task of fake review detection. We compute this distribution for each feature and substitute each feature value with the corresponding value of the distribution. To demonstrate the effectiveness of the proposed approach, the *distributional (cumulative) features* and the *basic* ones have been exploited to train several supervised machine-learning classifiers and the obtained results have been compared. To the best of the authors' knowledge, this is the first time that Cumulative Relative Frequency Distribution of a set of features has been considered for the unveiling of fake reviews. The experimental results show that the distributional features improve the performances of the classifiers, at the mere cost of a small computational surplus in the feature engineering phase.

The rest of the paper is organized as follows. The next section revises related work in the area. Section 3 describes the process of feature engineering. In Section 4, we present the experimental setup, while Section 5 reports the results of the comparison among the classification algorithms. Moreover, in this section, we assess the importance of the distributional features and discuss about the benefits brought by their adoption. Finally, Section 6 concludes the paper.

## 2 RELATED WORK

Social Media represent the perfect means for everyone to "spread contents in the form of User-Generated Content (UGG), almost without any traditional form of trusted control" [41]. Since years, Academia, Industry, and Platform Administrators have been fighting for developing automatic solutions to raise the users' awareness about the credibility of the news they read online. One of the contexts in which the problem of credibility assessment is receiving the most interest is spam - or fake - reviews detection. The existence of spam reviews has been known since the early 2000s when e-commerce and e-advice sites began to be popular.

In his seminal work, Liu lists three approaches to automatically identify opinion spam: the supervised, unsupervised, and group approaches [28]. In a standard supervised approach, a ground truth of a priori known genuine and fake reviews is needed. Then, features about the labeled reviews, the reviewers, and the reviewed products are engineered. The performances of the first models built on such features achieved good results with common algorithms such as Naive Bayes and Support Vector Machines [35].

As usual, a supervised approach is particularly challenging since it requires the existence of labeled data, that is, in our scenario, a set of reviews with prior knowledge about their (un)trustworthiness. To overcome the frequent issue of lack of labeled data, in the very first phases of investigation in this field, the work done by Jindal et al. in [23] exploited the fact that a common practice of fraudulent reviewers was to post almost duplicate reviews: reviews with similar texts were collected as fake instances. As shown in [11], linguistic features have been proven to be valid for fake reviews detection, particularly in the early advent of this phenomenon. Indeed, pioneer fake reviewers exhibited precise stylistic features in their texts, such as a marked use of short terms and expressions of positive feelings. Anomaly detection was also been widely employed in this field: an analysis of anomalous practices with respect to the average behavior of a genuine reviewer led to good results. Anomalous behavior of the reviewer may be related to general and early rating deviation, as highlighted by Liu in [28], or temporal dynamics (see Xie et al. [55]).

Going further with the useful methodologies, human annotators, possibly recruited from crowd-sourcing services like Amazon Mechanical Turk [3], have also been employed, both 1) to manually label reviews' sets to separate fake from non-fake reviews (e.g., see the very recent survey by Crawford et al. in [11]) and 2) to let them write intentionally false reviews, in order to test the accuracy of existing predictive models on such set of ad hoc crafted reviews, as nicely reproduced by Ott et al. in [40].

Recently, an interesting point of view has been offered by Cocarascu and Tonotti in [9]: deception is analysed based on contextual information derivable from review texts, but not in a standard way, e.g., considering linguistic features, but evaluating the influence and interactions that one text has on the others. The new feature, based on bipolar argumentation on the same review, has been shown to outperform more traditional features, when used in standard supervised classifiers, and even on small datasets.

Supervised learning algorithms usually need diverse examples - and the values of diverse features derived from such examples - for an accurate training phase. Wang et al. investigated the 'cold-start' problem [51]: the identification of a fake review when a new reviewer posts one review. Without enough data about the stylistic features of the review and the behavioral characteristics of the reviewer, the authors first find similarities between the review text under investigation and other review texts. Then, they consider similar behavior between the reviewer under investigation and the reviewers who posted the identified reviews. A model based on neural networks proves to be effective to approach the problem of lack of data in cold-start scenarios.

Although many years have passed and, as we will see briefly later, the problem has been addressed in many research works, with different techniques, automatically detecting a false review is an issue not completely solved yet, as stated in the recent survey of Wu et al. [54]. This inspiring work examines the phenomenon not only giving an overview of the various detection techniques used over time, but also proposing twenty future research questions. Notably, to help scholars find suitable datasets for a supervised classification task, this survey lists the currently available review datasets and their characteristics.

A similar work by Hussain et al [21], aimed at a comparison of different approaches, focuses on the performances obtained by different classification frameworks. Also, the authors carried on a relevance analysis of six different behavioral features of reviewers. Weighting the features with respect to their relevance, a classification over a baseline dataset obtains an 84.5% accuracy.

A quite novel work considers the unveiling of malicious reviewers by exploiting the notion of 'neighborhood of suspiciousness'. In [24], Kaghazgaran et al. proposed a system called TwoFace that, starting from identifiable reviewers paid by crowd-sourcing platforms to write fake reviews on well-known e-commerce platforms, such as Amazon, studies

the similarity between these and other reviewers, based, e.g., on the reviewed products, and shows how it is possible to spot organized fake reviews campaigns even when the reviewers alternate genuine and malicious behaviors.

Serra et al. developed a supervised approach where the task is to differentiate amongst different kinds of reviewers, from fraudulent, to uninformative, to reliable [48]. Leveraging a supervised classification approach based on a deep recurrent neural network, the system achieves notable performances over a real dataset where there is an a priori knowledge of the fraudulent reviewers.

The research work reminded so far lies in supervised learning. However, unsupervised techniques have been employed too, since they are very useful when no tagged data is available. As an example, the authors of [16] start from the same hypothesis as the above-cited [24] on the classification of the kind of reviewers. A reviewer may not always be considered either fraudulent or honest. Indeed, a behavioral analysis of the reviewer may leave a degree of uncertainty that can lead to classification errors. So, the work in [16] proposed an unsupervised learning approach based on fuzzy logic, developing a new deductive algorithm able to obtain about 80% accuracy on the classification of a group of reviewers, as belonging to one of seven categories of suspiciousness.

Lack of annotated datasets is not the only reason to resort to other than a supervised approach. In fact, the quality of available datasets can be questionable. The datasets need to be constantly updated, and it has been demonstrated that a human-operated annotation process is prone to error [40]. In this regard, the work by Rout et al. in [47] investigated semi-supervised approaches, i.e., approaches in which a small amount of labeled data is used in combination with a large amount of unlabeled data during training. The adoption of four well-known semi-supervised learning approaches shows a promising increase in the classification performances.

In recent years, a behavioral analysis of the target under investigation has been proven to be useful not only to discover individual fake users, but also to detect the coordinated and synchronized behavior that characterizes groups of malicious users. For some years now, acting in groups is a very common practice that characterizes social bots, i.e., automated social accounts programmed for unethical and often illicit online trafficking [12–14]. They have been massively employed to influence and alter the public opinion in major events, such as political elections and societal debates (e.g., see Badawi et al. about the 2016 US presidential elections [5] and Caldarelli et al. about the immigration from North Africa to Italy [7]). Also in the field of electronic word of mouth, some researchers have highlighted how it is possible to find reviewers, in this case real humans, paid to review the same product with predefined schemes and timing[1]. Fake reviewers' coordination can emerge by mining frequent behavioral patterns and ranking the most suspicious ones. A pioneer work by [36] first identifies groups of reviewers that reviewed the same set of products; then, the authors compute and aggregate an ensemble of anomaly scores (e.g., based on similarity amongst reviews and times at which the reviews have been posted): the scores are ultimately used to tag the reviewers as colluding or not. Another interesting approach for the analysis of colluding users is the one proposed by [49]: the authors check whether a given group of accounts (e.g., reviewers on *Yelp*) contains a subset of malicious accounts. The intuition behind this methodology is that the statistical distribution of reputation scores (e.g., number of friends and followers) of the accounts participating in a tampered computation significantly diverges from that of untampered ones.

We close this section by referring back to the division made by Liu in [28] about supervised, unsupervised, and group approaches to spot fake reviewers and/or reviews. As noted in [41], these are *data-driven* classification methods, mostly aiming at classifying in 'a binary or multiple way information items (i.e., credible vs non-credible)' with the evaluation of a series of credibility features extracted from the data. Notably, *model-driven* approaches, which are based on some

[1] 'Why I write fake online reviews', Online: https://www.bbc.com/news/uk-47952165

prior domain knowledge, are promising in providing a ranking of the information item (i.e., in our scenario, of the review) with respect to credibility. This is the case of recent work by Pasi et al. [42, 50], which exploits a Multi-Criteria Decision Making approach to assess the credibility of a review. In this context, a given review, seen as an alternative amongst others, is evaluated with respect to some credibility criteria. An overall credibility estimate of the review is then obtained by means of a suitable model-driven approach based on aggregation operators. This approach has also the advantage of assessing the contribution that single or interacting criteria/features have in the final ranking.

The techniques presented above have their pros and cons, and depending on the particular context, one approach can be preferred with respect to another. The most relevant contribution of our proposal with respect to the state of the art is to improve the effectiveness of the solution based on supervised classifiers, which, as seen above, is a well-known and widely-used approach in this context.

## 3 FEATURE ENGINEERING

In this section, we introduce a subset of features that have been adopted in past work to detect opinion spam and we propose how to modify them in order to improve the performances of classifiers. We emphasize that the listed features have been used effectively for this task by past researchers. We give below the rationale for their use in the context of unveiling fake reviews. Finally, it is worth noting that the list of selected features is not intended to be exhaustive.

### 3.1 Basic Features

Following a supervised classification approach, the selection of the most appropriate features plays a crucial role, since they may considerably affect the performance of the machine learning models constructed starting from them [11].

Features can be review-centric or reviewer-centric [22]. The former are features that refer to the review, while the latter refer to the reviewer. In the literature, several reviewer-centric features have been investigated, such as the maximum number of reviews, the percentage of positive reviews, the average review length, the reviewer rating deviation [37]. According to the outcomes of several works proposed in the context of opinion spam detection [35, 37, 46], we focused on reviewer-centric features, which have been demonstrated to be more effective for the identification of fake reviews. Thus, we relied on a set of *basic features*, which have been already used proficiently in the literature for the detection of opinion spam in reviews. Specifically, we focused on the following reviewer-centric features:

- **Photo Count**: This metric measures the number of pictures uploaded by a reviewer and is directly retrieved from the reviewer profile. In [58], the authors demonstrated the effectiveness of using photo count, together with other non-verbal features, for detecting fake reviews.
- **Review Count**: It measures how many reviews have been posted by a reviewer on the platform. [35, 37] showed that spammers and non-spammers present different behavior regarding the number of reviews they post. In particular, spammers usually post more reviews, since they may get paid. This feature has also been investigated by [29] and [52].
- **Useful Votes**: The most popular online review platforms allow users to rank reviews as useful or not. This information can be retrieved from the reviewer profile, or computed by summing the total amount of useful votes received by a reviewer. This feature has already been exploited by [58] and it has been demonstrated to be effective for opinion spam detection.
- **Reviewer Expertise**: Past research in [35, 56] highlights that reviewers with acquired expertise on the platform are less prone to cheat. Particularly, Mukherjee et al. in [36] report that opinion spammers are usually not

longtime members of a site. Genuine reviewers, however, use their accounts from time to time to post reviews. Although this experimental evidence does not mean that no spammer can be a member of a review platform for a long time, the literature has considered useful to exploit the activity freshness of an account in cheating detection. The Reviewer Expertise has been defined by Zhang et al. in [58] as the number of days a reviewer has been a member of the platform (the original name was Membership Length).

- **Average Gap**: The review gap is the time elapsed between two consecutive reviews. This feature has been previously introduced in the seminal work by Mukherjee et al., under the name *Activity Window* [38], and successfully re-adopted for detecting both colluders (i.e., spammers acting with a coordinated strategy) [35, 37] and singleton reviewers (i.e., reviewers with just isolated behavioral posting) [17]. In the cited work, the Activity window feature as been proved highly discriminant for demarcate spammers and non-spammers. Quoting from [38], "fake reviewers are likely to review in short bursts and are usually not longtime active members". On a Yelp dataset where was a priori known the benign and malicious nature of reviewers, work in [38] proved that, by computing the difference of timestamps of the last and first reviews for all the reviewers, a majority (80%) of spammers were bounded by 2 months of activity, whereas the same percentage of non-spammers remain active for at least 10 months. We define the Average Gap feature as the average time, in days, elapsed between two consecutive reviews of the same reviewer and is defined as:

$$AG_i = \frac{1}{N_i - 1} \sum_{j=2}^{N_i} (T_{i,j} - T_{i,j-1})$$

  where $AG_i$ is the Average Gap for the $i$-th user, $N_i$ is the number of reviews written by the user, $T_{i,j}$ is the timestamp of the $j$-th reviews of the $i$-th user.

- **Average Rating Deviation**: The rating deviation measures how much a reviewer's rating is far from the average rating of a business. [26] observed that spammers are more prone to deviate from the average rating than genuine reviewers. However, a bad experience may induce a genuine reviewer to deviate from the mean rating. The Average Rating Deviation is defined as follows [17, 23, 26, 37]:

$$ARD_i = \frac{1}{N_i} \sum_{j=1}^{N_i} |R_{i,j} - R_{B(j)}|$$

  where $ARD_i$ is the Average Rating Deviation of the $i$-th user, $N_i$ is the number of reviews written by the user, $R_{i,j}$ is the rating given by the $i$-th user to her/his $j$-th reviews corresponding to the business $B(j)$, $R_{B(j)}$ is the average rating obtained by the business $B(j)$.

- **First Review**: Spammers are usually paid to write reviews when a new product is placed on the market. This is due to the fact that early reviews have a great impact on consumers' opinions and, in turn, impact the sales, as pointed out by [26] and [37]. We compute the time elapsed between each review of a reviewer and the first review, for the same business. Then, we average the results on all the reviews. Specifically, the First Review value for reviewer $i$ is given by:

$$FRT_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (T_{i,j} - F_{B(j)})$$

  where $FR_i$ is the First Review value of the $i$-th user, $N_i$ is the number of reviews written by the user, $T_{i,j}$ is the time the $i$-th user wrote the $j$-th review and $F_B(j)$ is the time the first review of the same business $B(j)$, corresponding to the one of the $j$-th review, has been posted.

- **Reviewer Activity**: Several works pointed out that the more active a user on the online platform, the more the user is likely genuine [19], in terms of contributing with knowledge sharing in a useful way. The usefulness of this feature has been demonstrated several years ago. Since the early 00s, surveys have been conducted on large communities of individuals, trying to understand what drives them to be active and useful on an online social platform, in terms of sharing content [53]. Results showed that people contribute their knowledge when they perceive that it enhances their reputations, when they have the experience to share, and when they are structurally embedded in the network.
  The Activity feature expresses the number of days a user has been active and it is computed as:

$$A_i = T_{i,L} - T_{i,0}$$

  where $A_i$ is the activity (expressed in days) of the $i$-th user, $T_{i,L}$ is the time of the last review of the $i$-th user and $T_{i,0}$ is the time of the first review of the $i$-th user.

### 3.2 From Basic to Cumulative Features

The features described so far have been used to train a machine learning algorithm to construct a classifier, in a supervised-learning fashion.

In this work, we propose to build on the basic features, with a proper feature engineering process, to possibly assess an improvement of the classification performances. The proposed feature engineering process is based on the concept of Cumulative Relative Frequency Distribution.

The Relative Frequency is a quantity that expresses how often an event occurs divided by all outcomes. It can be easily represented by a Relative Frequency table. The Relative Frequency table is constructed directly from the data by simply dividing the frequency of a value by the total number of values in the dataset.

The Cumulative Relative Frequency is then calculated by adding each frequency from the Relative Frequency table to the sum of its predecessors. In practice, the Cumulative Relative Frequency indicates the percentage of elements in the dataset that lies below the current value. In this work, we modify each feature by using its Cumulative Relative Frequency Distribution.

In the following, we show an example of how to compute the Cumulative Relative Frequency Distribution. Let us consider the *Photo Count* feature and assume that for each photo count value, the corresponding number of occurrences is the one reported in the second column of Table 1. Thus, the second column reports the number of reviews associated with a reviewer who uploaded a given number of photos: in our example, there are 7,944 reviews whose reviewers have no photo associated. The third column measures the Relative Frequency, which is computed by dividing the number of occurrences by the total number of reviews. Finally, the fourth column reports the Cumulative Relative Frequency values, which have been obtained by adding each Relative Frequency value to the sum of its predecessor.

In our proposal, the process described so far is carried out for each basic feature and the Cumulative Relative Frequency values are used to train the classifier instead of the simple values. This involves, in practice, to substitute each value of the first column with the corresponding value of the fourth column of Table 1.

## 4 EXPERIMENTAL SETUP

In this section, we describe the setting of the experiments conducted to evaluate the effectiveness of the proposed features. This is done by comparing the results obtained when using the basic features and the cumulative ones with the most widespread supervised machine learning algorithms.

| Photo Count | Frequency (#Reviews) | Relative.Freq. (%Reviews) | Cumulative Rel.Freq. |
|---|---|---|---|
| 0 | 7944 | 0.44 | 0.44 |
| 1 | 2301 | 0.13 | 0.57 |
| 2 | 1756 | 0.10 | 0.67 |
| 3 | 1401 | 0.08 | 0.75 |
| 4 | 822 | 0.04 | 0.79 |
| 5 | 1382 | 0.08 | 0.87 |
| 6 | 550 | 0.03 | 0.90 |
| 7 | 347 | 0.02 | 0.92 |
| 8 | 780 | 0.04 | 0.96 |
| 9 | 342 | 0.02 | 0.98 |
| 10 | 460 | 0.02 | 1.00 |

Table 1. An example of Frequencies, Relative Frequencies, and Cumulative Relative Frequencies values for the Photo Count feature.

### 4.1 Dataset Construction and Characteristics

The dataset used in this study is composed of 56,317 business reviews, 42,673 businesses (both restaurants and hotels), and 1,429 reviewers. This dataset has been obtained by repopulating the YelpCHI dataset [57]. The YelpCHI dataset included 67,395 reviews from 201 hotels and restaurants done by 38,063 reviewers and each review was tagged with a fake/non-fake label. To tag reviewers in the YelpCHI dataset and to obtain fresher data to work with, we operate as follows:

(1) *Reviewers tagging*. We assign a fake/non-fake label to each reviewer in the following way: if all the YelpCHI reviews of a single reviewer are tagged as fake (non-fake), then we consider the corresponding reviewer as fake (non-fake). Instead, reviewers who presented a mix of fake and non-fake reviews in YelpCHI are tagged as *mix*. At the end of this elaboration, we measure in YelpCHI 79.7% non-fake reviewers, 20% fake reviewers, and 0.3% *mix* reviewers, which are discarded due to their limited number.

(2) *Dataset repopulation*. For the fake and non-fake reviewers, we crawl again their reviews from the Yelp website. The updated dataset includes reviews from 2005 to September 2018. Moreover, for each reviewer, we retrieve additional information, namely: 1) the number of posted photos, 2) the number of received useful votes, and 3) the date of registration to the platform. Some of the reviewers were no longer available on the Yelp website and we could not update their profile, therefore they were discarded.

In Table 2 we report a summary with the statistics of the basic features for the repopulated dataset, whereas in Figure 1 we present the correlation matrix, which shows the correlation coefficients among variables.

| | photo count | review count | useful votes | reviewer expertise | avg gap | avg rating deviation | first review | reviewer activity |
|---|---|---|---|---|---|---|---|---|
| **mean** | 170.9 | 201.9 | 502.7 | 3664.7 | 55.2 | 0.01 | 13.9 | 2637.6 |
| **std_dev** | 911 | 298.2 | 2089.45 | 579.8 | 110.6 | 0.06 | 50.2 | 991.3 |

Table 2. Mean and standard deviation of the basic features for the repopulated dataset.

From the correlation matrix, we notice a slightly positive correlation between *photo count* and *review count*, between *reviewer activity* and *reviewer expertise*, between *first review* and *average rating deviation*, and between *useful votes* and *photo count*. A stronger positive correlation is highlighted between *useful votes* and *review count*.

### 4.2 Data Labeling

One limitation of supervised classification approaches is the possible lack of labeled data. To overcome this problem, [39] relied on Amazon Mechanical Turks to generate fake and genuine reviews. Nevertheless, [37] highlighted the limits

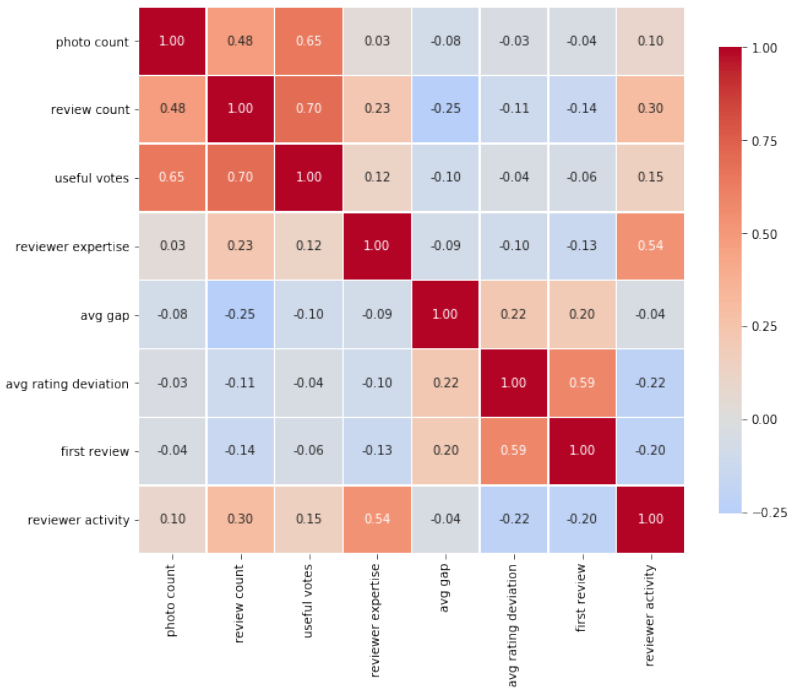


Fig. 1. Correlation matrix showing the correlation coefficients among the basic features.

of this approach, since workers could be not always effective in imitating the behavior of fake reviewers, and proposes to use the results of the Yelp classification algorithm for obtaining labeled data. The same approach has also been used by [45] and [59] and the literature has recognized its effectiveness in detecting fake reviews [29, 58]. According to this result, we use the same approach to build a dataset of labeled examples (as described in Section 4.1).

### 4.3 Experimental Framework

We experiment several supervised machine-learning algorithms, namely Logistic Regression (LogReg) [33], Linear Discriminant Analysis (LDA) [32], Support Vector Machine (SVM) [10], Decision Tree (DT) [6], Naive Bayes (NB) [30], K-Nearest Neighbors (k-NN) [2].

The performances of the learning algorithms are evaluated by means of commonly used metrics. Specifically, we compute the Accuracy on train (TraAcc) and test (TstAcc), the Precision (Pre), Recall (Rec) and F1-score (F1) for both classes. In addition, we compute the Matthews Correlation Coefficient (MCC) [31] and the Area Under Curve (AUC) [27], which are two evaluation metrics used to measure the quality of binary classifications and are useful to compare classification models' performances.

To ensure higher reliability of results, we apply a Stratified k-Fold Cross Validation approach [15, 25], with $k = 5$. The cross-validation involves partitioning the dataset into $k$ sets, then a model is trained using $k - 1$ folds (called training set) and validated on the remaining part of the data (validation set). To reduce variability, this process is repeated $k$ times, using only once each partition for the validation. The performance measures are obtained by averaging the validation results on all runs. The Stratified approach ensures the preservation of the frequencies of the classes in each

training and validation fold. The experimental framework described so far is depicted in Figure 2. All the experiments have been developed in Python, with the support of the Scikit-learn library [43].

## 5 EVALUATION AND DISCUSSION

In this section, we report the results obtained by applying the aforementioned algorithms for the detection of fake and genuine reviews, when using the basic or the cumulative features. Due to the imbalanced nature of the dataset, we report the performance metrics both for positive (fake-0) and negative (non fake-1) classes.

Table 3 and Table 4 report the results of the learning algorithms, obtained by using the basic and cumulative features, respectively. The performance values are computed by averaging the results over 5 runs of the algorithms, according to the employed cross-validation scheme.

| Algorithm | TraAcc | TstAcc | Prec-0 | Rec-0 | F1-0 | Prec-1 | Rec-1 | F1-1 | MCC | AUC |
|---|---|---|---|---|---|---|---|---|---|---|
| DT(Dep=10) | **0.99** | **0.97** | 0.77 | **1.00** | **0.87** | **1.00** | 0.96 | **0.98** | **0.86** | **0.98** |
| DT(Dep=5) | 0.97 | 0.94 | 0.67 | 1.00 | 0.80 | 1.00 | 0.94 | 0.97 | 0.79 | 0.97 |
| k-NN(k=5) | 0.93 | 0.92 | 0.71 | 0.79 | 0.75 | 0.96 | 0.94 | 0.95 | 0.70 | 0.95 |
| k-NN(k=10) | 0.90 | 0.89 | 0.70 | 0.64 | 0.67 | 0.92 | 0.94 | 0.93 | 0.61 | 0.93 |
| SVM(rbf) | 0.90 | 0.88 | **0.82** | 0.47 | 0.60 | 0.88 | **0.97** | 0.93 | 0.56 | 0.92 |
| LogReg | 0.87 | 0.85 | 0.43 | 0.89 | 0.58 | 0.99 | 0.85 | 0.91 | 0.55 | 0.87 |
| LDA | 0.84 | 0.81 | 0.36 | 0.87 | 0.51 | 0.98 | 0.80 | 0.88 | 0.48 | 0.84 |
| Naive Bayes | 0.85 | 0.78 | 0.34 | 0.96 | 0.50 | 0.99 | 0.75 | 0.86 | 0.49 | 0.85 |

Table 3. Algorithms performances obtained using the Basic Features.

Table 3 highlights that all the algorithms obtain very good precisions on the *non-fake* class (tagged with 1), while the performances are worst when dealing with the *fake* class. This is due to the fact that the dataset is imbalanced, with a ratio between classes of about 0.13. Among the experimented methods, the one that obtained the best precision on the minority class is the Support Vector Machine Classifier (precision = 0.97). However, the best global performances are obtained by the Decision Tree classifier (max depth=10), since in this case the MCC and the AUC reach the highest values (0.86 and 0.98, respectively). This occurs because the recall values obtained by the Decision Tree classifier are almost equal (majority class) or better (minority class) with respect to the ones obtained by the Support Vector Machine. We also experiment with a Decision Tree classifier with *max depth=5*, to obtain a less complex model, but we notice that the performances on the minority class drop dramatically. Still regarding the precision on the minority class, all the remaining algorithms but the SVM are outperformed by Decision Tree (max depth=10). However, all of them, except the SVM, perform quite well with respect to the recall. This implies that the algorithms work well in finding the fake instances, but there are also many false positives, i.e., genuine reviews erroneously classified as fake.

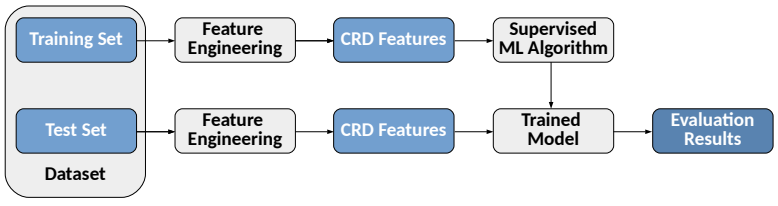


Fig. 2. Experimental framework scheme.

| Algorithm | TraAcc | TstAcc | Prec-0 | Rec-0 | F1-0 | Prec-1 | Rec-1 | F1-1 | MCC | AUC |
|---|---|---|---|---|---|---|---|---|---|---|
| DT(Dep=10) | **0.99** | **0.98** | **0.83** | **1.00** | **0.91** | **1.00** | **0.97** | **0.99** | **0.90** | **0.99** |
| DT(Dep=5) | 0.96 | 0.93 | 0.64 | 1.00 | 0.78 | 1.00 | 0.92 | 0.96 | 0.77 | 0.96 |
| k-NN(k=5) | 0.97 | 0.94 | 0.64 | 1.00 | 0.78 | 1.00 | 0.93 | 0.96 | 0.77 | 0.96 |
| k-NN(k=10) | 0.95 | 0.91 | 0.56 | 1.00 | 0.72 | 1.00 | 0.90 | 0.95 | 0.71 | 0.95 |
| SVM(rbf) | 0.92 | 0.90 | 0.53 | 0.94 | 0.68 | 0.99 | 0.89 | 0.94 | 0.66 | 0.92 |
| LogReg | 0.91 | 0.89 | 0.45 | 0.93 | 0.66 | 0.99 | 0.89 | 0.93 | 0.64 | 0.91 |
| LDA | 0.90 | 0.88 | 0.50 | 0.93 | 0.65 | 0.99 | 0.88 | 0.93 | 0.62 | 0.90 |
| Naive Bayes | 0.86 | 0.86 | 0.44 | 0.86 | 0.58 | 0.98 | 0.86 | 0.91 | 0.55 | 0.86 |

Table 4. Algorithms performances obtained using the Cumulative Relative Frequency distributions.

The results described so far and presented in Table 3 are generally worse with respect to those shown in Table 4, where we reported the results obtained by using the Cumulative Relative Frequency distributions. Also when using the cumulative features, the best approach is the Decision Tree (max depth=10), which outperforms the other algorithms. By comparing the two tables, we notice that the results obtained by k-NN classifiers, using the basic features, reach higher values in the precision of the minority class, but this improvement in precision is balanced by a lower recall. In practice, the fake reviews detected by the classifiers are indeed fake, but they also miss a lot of actual fakes.

To summarize the performance differences, we report in Table 5 a comparison between the MCC and the AUC values. In particular, for each metric, we reported the corresponding value obtained by using the basic features, the cumulative features, and the improvement percentage. Table 5 highlights that the MCC and the AUC values are always higher or equal when using cumulative features, for all the algorithms except for the Decision Tree (max depth =5), since in this case the basic features lead to better MCC and AUC values.

| Algorithm | MCC | | | AUC | | |
|---|---|---|---|---|---|---|
| | **Basic** | **Cumulative** | **Improv.(%)** | **Basic** | **Cumulative** | **Improv.(%)** |
| DT(Dep=10) | 0.86 | **0.90** | 4.7 | 0.98 | **0.99** | 1.0 |
| DT(Dep=5) | **0.79** | 0.77 | -2.5 | **0.97** | 0.96 | -1.0 |
| k-NN(k=5) | 0.70 | **0.77** | 10.0 | 0.95 | **0.96** | 1.1 |
| k-NN(k=10) | 0.61 | **0.71** | 16.4 | 0.93 | **0.95** | 2.2 |
| SVM(rbf) | 0.56 | **0.66** | 17.9 | **0.92** | **0.92** | 0.0 |
| LogReg | 0.55 | **0.64** | 16.4 | 0.87 | **0.91** | 4.6 |
| LDA | 0.48 | **0.62** | 29.2 | 0.84 | **0.90** | 7.1 |
| Naive Bayes | 0.49 | **0.55** | 12.2 | 0.85 | **0.86** | 1.2 |

Table 5. Performances comparison between Basic and Cumulative Features.

We finally remark that the process described in Section 3.2 represents a pre-processing step and it can be performed only once before applying several machine learning algorithms. We computed the time required to build the Cumulative Relative Frequency for all the features in the pre-processing of the cross-validation process and we obtained an average value of 155 ms, pointing out that the impact of the calculation time for the proposed features is negligible.

### 5.1 Features Importance

The importance of features plays a significant role in a predictive model, since it provides insights into the data and into the model itself. Moreover, it poses the basis for dimensionality reduction and feature selection, which can sometimes improve the efficiency and the effectiveness of a predictive model. From the experiments carried on, the best algorithm resulted to be a Decision Tree model. Decision Tree algorithms offer importance scores based on the reduction in the criterion used to select split points in the tree, such as Gini or entropy. In this case, the importance of a feature has been

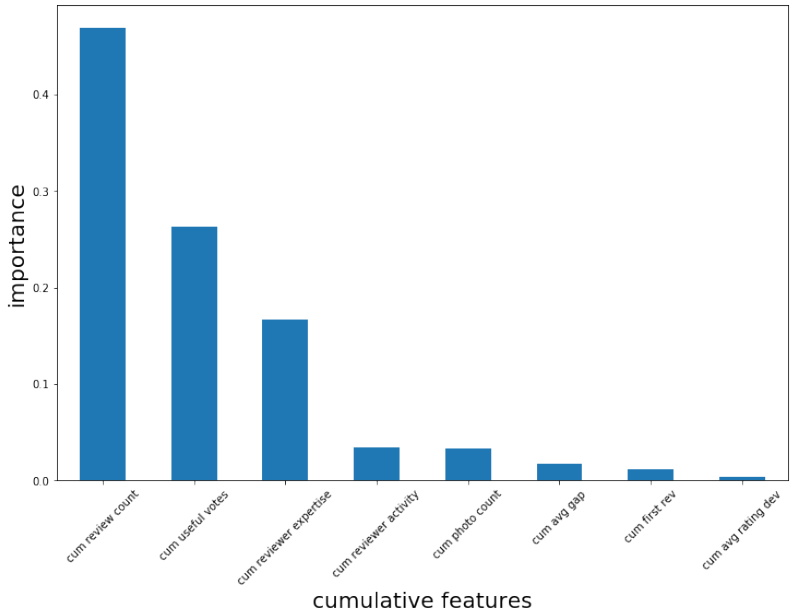


Fig. 3. Feature importance of cumulative features computed according to Gini index, for the Decision Tree model (max depth = 10).

computed following the Gini importance [6], which counts the times a feature is used to split a node, weighted by the number of samples it splits. We computed the Gini importance in each fold of the cross-validation and we averaged the results obtained. In Figure 3, we report a graph showing the importance of each cumulative feature, from the most important to the least one. The first three features, namely the number of reviews posted on the platform (*cum review count*), the number of useful votes received (*cum useful votes*), and the membership length of a reviewer (*cum reviewer expertise*) are probably the most useful for this predictive model.

To confirm this intuition, we perform a feature selection process, by selecting the first three most important features. A new Decision Tree model has been trained and tested, and the results (averaged on 5-folds) are reported in Table 6. The features selected are the same for both models, although they appear in different order of importance. For the sake of clarity, we reported in Table 6 the results already presented in Table 4, related to the Decision Tree models without feature selection. From their comparison, we notice that the feature selection process actually contributes to slightly improving the performances. In particular, for the Decision Tree model with *max depth=5*, this improvement is more evident since all the metrics reach higher values when the feature selection is applied. On the other hand, for the Decision Tree model with *max depth=10*, there is a small improvement in the precision and the F1-score of the positive (fake-0) class, whereas the recall for the negative (non fake-1) class is a bit lower with the feature selection applied. Nevertheless, the global performances are better, since the MCC value is higher.

### 5.2 Discussion on Feature Enhancement

In this section, we explain at a conceptual level why using the cumulative relative frequencies works better than the simple values. In data pre-processing, the normalization task changes the feature values to a common scale to avoid that differences in the data ranges can distort the results. Min-max normalization is one of the most used methods and

| Algorithm | Selected Features | TraAcc | TstAcc | Prec-0 | Rec-0 | F1-0 | Prec-1 | Rec-1 | F1-1 | MCC | AUC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DT(Dep=10) | c.rev.count, c.useful votes, c.rev.exp. | **0.99** | **0.98** | **0.85** | **1.00** | **0.92** | **1.00** | **0.98** | **0.99** | **0.92** | **0.99** |
| DT(Dep=5) | c.rev.count, c.rev.exp., c.useful votes | 0.97 | 0.95 | 0.70 | 1.00 | 0.82 | 1.00 | 0.94 | 0.97 | 0.81 | 0.97 |
| DT(Dep=10) | all | 0.99 | 0.98 | 0.83 | 1.00 | 0.91 | 1.00 | 0.97 | 0.99 | 0.90 | 0.99 |
| DT(Dep=5) | all | 0.96 | 0.93 | 0.64 | 1.00 | 0.78 | 1.00 | 0.92 | 0.96 | 0.77 | 0.96 |

Table 6. Algorithms performances obtained with and without feature selection for the Decision Tree models.

consists in re-scaling the values of features in the range [0, 1]. However, min-max normalization suffers from outliers, because it is sufficient one relevant outlier to flatten all the input values. Another normalization method that transforms the feature values in such a way that they have zero-mean is Z-score. This method avoids the outlier issue but does not produce data on the same scale.

Several probabilities distributions are affected by the presence of outliers. One of them is the Zipfian distribution, which is a discrete power-law probability distribution [60] in which the frequency of input is inversely proportional to its rank in the frequency table. That is, the most frequent inputs occur twice as often as the second most frequent ones, three times as often as the third most frequent inputs, and so on. For example, consider the number of friends in a social network: normally, most users have very few friends (usually, 0 is the most frequent value), whereas very few users have a very high number of friends. In these cases, the performance of min-max normalization gets worse due to outliers, whereas Z-score does not well scale the input data.

Our approach allows us to scale the value of a feature in the desired range [0, 1] using its rank in the frequency table, in a similar way as the Zipfian distribution. This transformation does not suffer from outliers: indeed, the normalized value of a feature does not depend on the magnitude of an outlier. Moreover, the normalized values are better distributed.

Consider again the example of the number of friends in a social network presented above: the normalized feature $v$ of a user with no friend is equal to zero using any standard normalization method. In our approach, $v = \frac{n_0}{n}$, which is the fraction of users with no friends.

By examining the features used in our experiments, we found that the most important ones follow a Zipfian distribution. This result confirms the findings of other research works, such as [1, 4, 34, 44], that observed a power-law distribution in many social network dimensions.

We expect that our strategy performs well in scenarios in which input data are distributed according to a power-law probability distribution because our feature normalization is strongly derived from the power-law distribution. Thus, as a practical recommendation in a classification task, we suggest to identify the probability distribution of each feature: in the case such a distribution follows the power law, then re-engineering this feature by considering the Cumulative Relative Frequency Distribution should improve the overall accuracy of the classifier performances. By considering that, in many fields, such as physical and social sciences [8], data follow a Zipfian distribution, we conclude that our normalization strategy can be effective in many contexts.

## 6 CONCLUSIONS

User opinions are an important information source, which can help a customer and a vendor to evaluate pros and cons of the buying/selling when they interact. For the importance of opinion role, there is the possibility to have unfair opinions used to promote own products or to disparage products of competitors. The important challenge of detecting unfair opinions has attracted and attracts the scientific community and one of the most promising approaches to address this problem is based on the use of supervised classifiers, which have been proven to be highly effective.

In this paper, we tried to further improve their effectiveness, not by proposing some change in the well-tested state-of-the-art algorithms, but only by modifying the input used for the training phase to construct supervised classifiers. Specifically, we considered eight features widely used to detect opinion spam and pre-processed them by considering the cumulative relative frequency distribution. To demonstrate the effectiveness of our proposal, we extracted a data set from Yelp.com and measured the performances of the six most used classifiers in detecting opinion spam, both in their standard use and when our proposal is adopted. The results of this comparison show that the use of the cumulative relative frequency distribution improves the performance of the state-of-the-art classifiers.

As future work, we intend to extend our proposal to detect not only individual spammers, but also groups of users who, acting in a coordinated and synchronized way, aim to give credit or discredit a product (or a service). The idea is that, once an ensemble of malicious reviewers is detected, an overlapping between the products that malicious reviewers have evaluated is searched. Groups of users with large overlap (i.e., who revised the same products) could be colluders.

## ACKNOWLEDGMENTS

Partially supported by the European Union's Horizon 2020 programme (grant agreement No. 830892, SPARTA); by the Integrated Activity Project TOFFEe (TOols for Fighting FakEs), funded by IMT Scuola Alti Studi Lucca; and by the IIT-CNR project DESIRE (DissEmination of ScIentific REsults).

## REFERENCES

[1] Lada A Adamic, Rajan M Lukose, Amit R Puniyani, and Bernardo A Huberman. 2001. Search in power-law networks. *Physical review E* 64, 4 (2001), 046135.
[2] N.S. Altman. 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician* 46, 3 (1992), 175–185.
[3] Amazon. 2019. Amazon Mechanical Turk. https://www.mturk.com/.
[4] Alex Arenas, Leon Danon, Albert Diaz-Guilera, Pablo M Gleiser, and Roger Guimera. 2004. Community analysis in social networks. *The European Physical Journal B* 38, 2 (2004), 373–380.
[5] Adam Badawy, Aseel Addawood, Kristina Lerman, and Emilio Ferrara. 2019. Characterizing the 2016 Russian IRA influence campaign. *Social Netw. Analys. Mining* 9, 1 (2019), 31:1–31:11.
[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. 2002. *Applied Logistic Regression Analysis.* Chapman and Hall.
[7] Guido Caldarelli, Rocco De Nicola, Fabio Del Vigna, Marinella Petrocchi, and Fabio Saracco. 2020. The role of bot squads in the political propaganda on Twitter. *Communications Physics* 3, 1 (2020), 1–15.
[8] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.
[9] Oana Cocarascu and Francesca Toni. 2018. Combining deep learning and argumentative reasoning for the analysis of social media textual content using small data sets. *Computational Linguistics* 44, 4 (2018), 833–858.
[10] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Mach. Learn.* 20, 3 (Sept. 1995), 273–297.
[11] Michael Crawford, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron N. Richter, and Hamzah Al Najada. 2015. Survey of review spam detection using machine learning techniques. *J. Big Data* 2 (2015), 23.
[12] Stefano Cresci. 2020. A decade of social bot detection. *Commun. ACM* 63, 10 (2020), 72–83.
[13] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2019. Better Safe Than Sorry: An Adversarial Approach to Improve Social Bot Detection. In *Proceedings of the 11th ACM Conference on Web Science, WebSci 2019, Boston, MA, USA, June 30 - July 03, 2019.* ACM, 47–56.
[14] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2019. On the capability of evolved spambots to evade detection via genetic engineering. *OSNEM* 9 (2019), 1–16.
[15] Pierre A. Devijver and Josef Kittler. 1982. *Pattern recognition: a statistical approach.* Prentice-Hall.
[16] Komal Dhingra and Sumit Kr Yadav. 2019. Spam analysis of big reviews dataset using Fuzzy Ranking Evaluation Algorithm and Hadoop. *International Journal of Machine Learning and Cybernetics* 10 (2019), 2143–2162.
[17] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. 2013. Exploiting burstiness in reviews for review spammer detection. In *7th International Conference on Weblogs and Social Media, ICWSM.* AAAI Press, 175–184.
[18] Emilio Ferrara. 2019. The history of digital spam. *Commun. ACM* 62, 8 (2019), 82–91.
[19] Paulo B Goes, Mingfeng Lin, and Ching man Au Yeung. 2014. "Popularity effect" in user-generated content: Evidence from online product reviews. *Information Systems Research* 25, 2 (2014), 222–238.

[20] Peter Hernon. 1995. Disinformation and misinformation through the internet: Findings of an exploratory study. *Government Information Quarterly* 12, 2 (1995), 133 – 139.

[21] Naveed Hussain, Hamid Turab Mirza, and Ibrar Hussain. 2019. Detecting Spam Review through Spammer's Behavior Analysis. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* 8, 2 (2019), 61–71.

[22] N. Jindal and B. Liu. 2007. Analyzing and Detecting Review Spam. In *Data Mining*. IEEE, 547–552.

[23] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Web Search and Data Mining* (Palo Alto, California, USA) *(WSDM)*. ACM, 219–230.

[24] Parisa Kaghazgaran, James Caverlee, and Anna Squicciarini. 2018. Combating Crowdsourced Review Manipulators: A Neighborhood-Based Approach. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 306–314.

[25] Ron Kohavi. 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (Montreal, Quebec, Canada) *(IJCAI'95)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1137–1143.

[26] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting Product Review Spammers Using Rating Behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. ACM, New York, NY, USA, 939–948.

[27] Charles X. Ling, Jin Huang, and Harry Zhang. 2003. AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In *Advances in Artificial Intelligence*, Yang Xiang and Brahim Chaib-draa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 329–341.

[28] Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

[29] Michael Luca and Georgios Zervas. 2013. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. *SSRN Electronic Journal* 62 (01 2013). Issue 12.

[30] M. E. Maron. 1961. Automatic Indexing: An Experimental Inquiry. *J. ACM* 8, 3 (July 1961), 404–417.

[31] Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *BBA-Protein Structure* 405, 2 (1975), 442–451.

[32] Geoffrey J. McLachlan. 2004. *Discriminant analysis and statistical pattern recognition*. John Wiley and Sons.

[33] Scott Menard. 1984. *Classification and Regression Trees*. SAGE Publications.

[34] Lev Muchnik, Sen Pei, Lucas C Parra, Saulo DS Reis, José S Andrade Jr, Shlomo Havlin, and Hernán A Makse. 2013. Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Scientific Reports* 3, 1 (2013), 1–8.

[35] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Spotting Opinion Spammers Using Behavioral Footprints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, New York, NY, USA, 632–640.

[36] Arjun Mukherjee, Bing Liu, and Natalie S. Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*. ACM, 191–200.

[37] Animesh Mukherjee, V. Venkataraman, B. Liu, and N. Glance. 2013. What Yelp fake review filter might be doing?. In *Conference on Weblogs and Social Media*. AAAI Press, 409–418.

[38] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, Natalie Glance, et al. 2013. Fake review detection: Classification and analysis of real and pseudo reviews. Tech Rep UIC-CS-2013–03, University of Illinois at Chicago.

[39] Myle Ott, Claire Cardie, and Jeff Hancock. 2012. Estimating the Prevalence of Deception in Online Review Communities. In *Proceedings of the 21st International Conference on World Wide Web* (Lyon, France) *(WWW '12)*. ACM, New York, NY, USA, 201–210.

[40] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Portland, Oregon) *(HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 309–319.

[41] Gabriella Pasi and Marco Viviani. 2020. Information Credibility in the Social Web: Contexts, Approaches, and Open Issues. *CoRR* abs/2001.09473 (2020).

[42] Gabriella Pasi, Marco Viviani, and Alexandre Carton. 2019. A Multi-Criteria Decision Making approach based on the Choquet integral for assessing the credibility of User-Generated Content. *Inf. Sci.* 503 (2019), 574–588.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[44] Daphne R Raban and Eyal Rabin. 2009. Statistical inference from power law distributed web-based social interactions. *Internet Research* 19, 3 (2009), 266–278.

[45] Mahmudur Rahman, Bogdan Carbunar, Jaime Ballesteros, and Duen Horng Chau. 2015. To catch a fake: Curbing deceptive Yelp ratings and venues. *Statistical Analysis and Data Mining* 8 (2015), 147–161.

[46] Shebuti Rayana and Leman Akoglu. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. ACM, New York, NY, USA, 985–994.

[47] J. K. Rout, A. Dalmia, K. R. Choo, S. Bakshi, and S. K. Jena. 2017. Revisiting Semi-Supervised Learning for Online Deceptive Review Detection. *IEEE Access* 5 (2017), 1319–1327.

[48] Edoardo Serra, Anu Shrestha, Francesca Spezzano, and Anna Cinzia Squicciarini. 2020. DeepTrust: An Automatic Framework to Detect Trustworthy Users in Opinion-based Systems. In *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, March 16-18, 2020*. ACM, 29–38.

[49] Bimal Viswanath, Muhammad Ahmad Bashir, Muhammad Bilal Zafar, Simon Bouget, Saikat Guha, Krishna P Gummadi, Aniket Kate, and Alan Mislove. 2015. Strength in Numbers: Robust Tamper Detection in Crowd Computations. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*. ACM, 113–124.

[50] Marco Viviani and Gabriella Pasi. 2017. Quantifier Guided Aggregation for the Veracity Assessment of Online Reviews. *Int. J. Intell. Syst.* 32, 5 (2017), 481–501.

[51] Xuepeng Wang, Kang Liu, and Jun Zhao. 2017. Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 366–376.

[52] Yuanyuan Wang, Stephen Chi Fai Chan, Grace Ngai, and Hong-Va Leong. 2013. Quantifying Reviewer Credibility in Online Tourism. In *Database and Expert Systems Applications*, Hendrik Decker, Lenka Lhotská, Sebastian Link, Josef Basl, and A. Min Tjoa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 381–395.

[53] Molly McLure Wasko and Samer Faraj. 2005. Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice. *MIS Quarterly* 29, 1 (2005), 35–57.

[54] Yuanyuan Wu, Eric W.T. Ngai, Pengkun Wu, and Chong Wu. 2020. Fake online reviews: Literature review, synthesis, and directions for future research. *Decision Support Systems* 132 (2020), 113280.

[55] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. 2012. Review Spam Detection via Temporal Pattern Discovery. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. ACM, 823–831.

[56] Chang Xu. 2013. Detecting Collusive Spammers in Online Review Communities. In *Proceedings of the Sixth Workshop on Ph.D. Students in Information and Knowledge Management* (San Francisco, California, USA) *(PIKM '13)*. ACM, New York, NY, USA, 33–40.

[57] Yelp. 2019. YelpCHI dataset. http://odds.cs.stonybrook.edu/yelpchi-dataset/.

[58] Dongsong Zhang, Lina Zhou, Juan Luo Kehoe, and Isil Yakut Kilic. 2016. What Online Reviewer Behaviors Really Matter? Effects of Verbal and Nonverbal Behaviors on Detection of Fake Online Reviews. *Journal of Management Information Systems* 33, 2 (2016), 456–481.

[59] Wenqi Zhou and Wenjing Duan. 2016. Do Professional Reviews Affect Online User Choices Through User Reviews? An Empirical Study. *Journal of Management Information Systems* 33, 1 (2016), 202–228.

[60] George Kingsley Zipf. 1949. *Human behaviour and the principle of least-effort.* Addison-Wesley Press.