# Turning the Tide: Curbing Deceptive Yelp Behaviors

Mahmudur Rahman
Florida Int'l University
mrahm004@cs.fiu.edu

Bogdan Carbunar
Florida Int'l University
carbunar@cs.fiu.edu

Jaime Ballesteros
Nokia Inc.
jaime.ballesteros@here.com

George Burri
Jive Software
george.burri@jivesoftware.com

Duen Horng (Polo) Chau
Georgia Tech
polo@gatech.edu

## Abstract

The popularity and influence of reviews, make sites like Yelp ideal targets for malicious behaviors. We present Marco, a novel system that exploits the unique combination of social, spatial and temporal signals gleaned from Yelp, to detect venues whose ratings are impacted by fraudulent reviews. Marco increases the cost and complexity of attacks, by imposing a tradeoff on fraudsters, between their ability to impact venue ratings and their ability to remain undetected. We contribute a new dataset to the community, which consists of both ground truth and gold standard data. We show that Marco significantly outperforms state-of-the-art approaches, by achieving 94% accuracy in classifying reviews as fraudulent or genuine, and 95.8% accuracy in classifying venues as deceptive or legitimate. Marco successfully flagged 244 deceptive venues from our large dataset with 7,435 venues, 270,121 reviews and 195,417 users. Among the San Francisco car repair and moving companies that we analyzed, almost 10% exhibit fraudulent behaviors.

## 1 Introduction and Motivation

Online reviews are central to numerous aspects of people's daily online and physical activities. Which Thai restaurant has good food? Which mover is reliable? Which mechanic is trustworthy? People rely on online reviews to make decisions on purchases, services and opinions, among others. People assume these reviews are written by real patrons of venues and services, who are sharing their honest opinions about what they have experienced. But, is that really the case? Unfortunately, no. Reviews are sometimes fake, written by fraudsters who collude to write glowing reviews for what might otherwise be mediocre services or venues [1, 2, 3, 4].

In this paper we focus on Yelp [5], a popular social networking and location based service that exploits crowdsourcing to collect a wealth of peer reviews concerning venues and services. Crowdsourcing has however exposed Yelp to significant malicious behaviors: Up to 25% of Yelp reviews may be fraudulent [6].

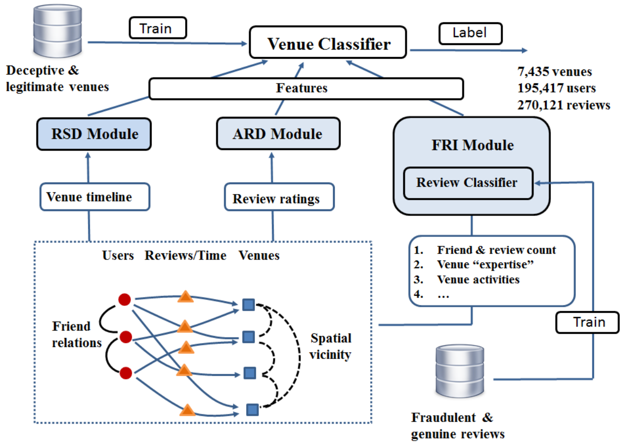


Figure 1: System overview of *Marco*. Marco relies on social, temporal and spatial signals gleaned from Yelp, to extract novel features. The features are used by the *venue classifier* module to label venues (deceptive vs. legitimate) based on collected *ground truth* and *gold standard* data. Section 4 describes Marco in detail.

While malicious behaviors may occasionally be performed by inexperienced fraudsters, they may also be professionally organized. For example, *search engine optimization* (SEO) companies tap into review writer markets [7, 8, 9] to offer *review campaigns* or "face lift" operations for business owners [10], to manipulate venues' ratings (1–5 star) through multiple, coordinated artificial reviews. For business owners, profit seems to be the main incentive to drive them to engage in deceptive activities. Studies have shown that an extra half-star rating on Yelp causes a restaurant to sell out 19% more often [11], and a one-star increase leads to a 5–9% increase in revenue [12].

We propose *Marco* (MAlicious Review Campaign Observer), a novel system that leverages the wealth of spatial, temporal and social information provided by Yelp, to detect venues that are targets of deceptive behaviors. Marco (see Figure 1) exploits fundamental fraudster limitations (see Section 4.1) to identify venues with (i) abnormal review spikes, (ii) series of dissenting reviews and (iii) impactful but suspicious reviews. Marco detects both venues that receive large numbers of fraudulent reviews, and venues that have insufficient genuine reviews to neutralize the effects of even small scale campaigns. Our major contributions include:

- We introduce a *lower bound* on the number of reviews required to launch a review campaign that impacts a target venue's rating, and prove that this bound renders such campaigns detectable. Our theoretical results force fraudsters to compromise between the impact and undetectability of their review campaigns. [Section 4]
- We present *Marco*, a system that leverages novel social, spatial and temporal features gleaned from Yelp to flag suspicious reviews and venues. Marco makes it much harder for fraudsters to hide their trails by making the tasks of posting fraudulent reviews much more costly and complex. [Section 4]
- We contribute a novel dataset of reviews and venues, which consists of both ground truth (i.e., objectively correct) and gold standard instances (i.e., selected based on best available strategies); and a large collection of 7,435 venues, 270,121 reviews and 195,417 reviewer profiles. [Section 3].
- We demonstrate that Marco is effective and fast; its classification accuracy is up to 94% for reviews, and 95.8% for venues. It flags 244 of the 7,435 venues analyzed as deceptive; manual inspection revealed that they were indeed suspicious. [Section 5]

Marco aims to complement legal actions against profitable, fraudulent review activities [10]. Organizations caught red-handed in setting up review campaigns have been shown to pay \$1-\$10 per fraudulent review. By making the cost of purchasing reviews approach the cost of products and services provided by hiring venues, Marco has the potential to act as an economic counter-incentive for rational venue owners.

## 2 Related Work, Background, and Our Differences

**2.1 Yelp's Review System.** For this work, we focus on Yelp [5], a review centric geosocial network that hosts information concerning users and venues. Subscribed users ("yelpers") have accounts and can write reviews, befriend other subscribers, report locations and search for venues of interest. Venues represent businesses or events with an associated location (e.g., restaurants, shops, offices, concerts).

Reviews have a star *rating*, an integer ranging from 1 to 5, with 5 being the highest mark. An *average rating* value is computed for each venue (rounded to the nearest half star), over the ratings of all the posted reviews. For a review $R$, let $R.\rho$ denote its rating and $R.\tau$ to denote the time when the review was posted. We say a review is "positive" if its rating is at least 4 stars and "negative" if its rating is 2 stars or fewer.

**2.2 Influential & Elite Yelpers.** Users can rate the reviews of others, by clicking on associated buttons (e.g., "useful", "funny" or "cool" buttons). They can upload photos taken at venues reviewed and perform "check-ins", to formally record their real-time presence at the venue. Yelp rewards "influential" reviewers (often peer-recommended) with a special, yearly "Elite" badge.

**2.3 Fraudulent Reviews & Deceptive Venues.** A review is *fraudulent* if it describes a fictitious experience. Otherwise, the review is *genuine*. We say a venue is *deceptive* if it has received a sufficient number of fraudulent reviews to impact its average rating. Otherwise, the venue is *legitimate*.

Yelp relies on proprietary algorithms to filter reviews it considers fraudulent. See [13] for an attempt to reverse engineer Yelp's filter. Furthermore, Yelp has launched a "Consumer Alert" process, posting "alert badges" on the pages of venues for which (i) people were caught red-handed buying fraudulent reviews, offering rewards or discounts for reviews or (ii) that have a large number of reviews submitted from the same IP address. The consumer alert badge is displayed for 90 days.

**2.4 Research in Detecting Fraudulent Reviews.** Jindal and Liu [2] introduce the problem of detecting opinion spam for Amazon reviews. They proposed solutions for detecting spam, duplicate or plagiarized reviews and outlier reviews. Jindal et al. [3] identify unusual, suspicious review patterns. In order to detect "review spam", Lim et al. [4] propose techniques that determine a user's deviation from the behavior of other users reviewing similar products. Mukherjee et al. [14] focus on fake reviewer groups; similar organized fraudulent activities were also found on online auction sites, such as eBay [15]. Mukherjee et al. [16] leverage the different behavioral distributions of review spammers to learn the population distributions of spammer and non-spammer clusters. Li et al. [17] exploit the reviews of reviews concept of Epinions to collect a review spam corpus, then propose a two view, semi-supervised method to classify reviews.

Ott et al. [18] integrate work from psychology and computational linguistics to develop and compare several text-based techniques for detecting deceptive TripAdvisor reviews. To address the lack of ground truth, they crowdsourced the job of writing fraudulent reviews for existing venues.

Unlike previous research, we focus on the problem of detecting *impactful* review campaigns. Our approach takes advantage of the unique combination of social, spatial and temporal dimensions of Yelp. Furthermore, we do not break Yelp's terms of service to collect ground truth data. Instead, we take advantage of unique Yelp features (i.e., spelp sites, consumer alerts) to collect a combination of ground truth and gold standard review and venue datasets.

Feng et al [19] seek to address the lack of ground truth data for detecting deceptive Yelp venues: They introduce three venue features and use them to collect gold standard sets of deceptive and legitimate venues. They show that an SVM classifier is able to classify these venues with an accuracy of up to 75%. In Section 5 we confirm their results on our datasets. We show that with an accuracy of 95.8%, Marco significantly outperforms the best strategy of Feng et al [19].

## 3 Collected Yelp Data.

In this section we describe the Yelp datasets we collected using the *YCrawl* crawler that we developed. Our data consists of: (i) 90 deceptive and 100 legitimate venues; (ii) 200 fraudulent and 202 genuine reviews; and (iii) a large collection of 7,435 venues and their 270,121 reviews from 195,417 reviewers, from San Francisco, New York City and Miami.

**3.1 YCrawl.** Written with 1820 lines of Python code, YCrawl fetches raw HTML pages of Yelp venue and user accounts. YCrawl uses a pool of servers, IP proxies [20], and DeathByCaptcha [21] to collect CAPTCHA-protected reviews filtered by Yelp.

We used YCrawl to collect a seed dataset of random venue and user accounts, using a breadth-first crawling strategy and stratified sampling [22]. This seed dataset initiated the collection of subsequent datasets. First, we collected 100 venues randomly selected from 10 major US cities (e.g., NY, San Francisco, LA, Chicago, Miami). Second, we used YCrawl to collect basic account information of the 10,031 Yelp users who reviewed those venues. Third, we randomly selected a subset of 16,199 venues from all the venues reviewed by those users.

**3.2 The Data.** We use the term "ground truth" set to denote data objectively known to be correct. We use the term "gold standard" to denote data selected according to the best available strategies. We collect such data following several stringent requirements, often validated by multiple third-parties.

**Ground truth deceptive venues.** We relied on Yelp's "Consumer Alert" feature to identify deceptive venues. We have used Yelp and Google to identify a snapshot of all the 90 venues that received consumer alerts during July and August, 2013.

**Gold standard legitimate venues.** We have used the collected list of 16,199 venues previously described to first selected a preliminary list of venues with well known consistent quality, e.g., the "Ritz-Carlton" hotel. We have then manually verified each review of each venue, including their filtered reviews. We have selected only venues with at most one tenth of their reviews filtered by Yelp and whose filtered reviews include a balanced amount of positive and negative ratings. While Yelp tends to filter reviews received from users with few friends and reviews, Feng et al. [19] showed that this strategy is not accurate. In total, we selected 100 legitimate venues.

**Gold standard fraudulent reviews.** We have used spelp (Spam + Yelp) sites (e.g., [23, 24]), forums where members, often "Elite" yelpers with ground truth knowledge, reveal and initiate the discussion on fraudulent Yelp reviews. While in theory such sites are ideal targets for fraudulent behavior, the high investment imposed on fraudsters, coupled with the low visibility of such sites, make them unappealing options. Nevertheless, we have identified spelp reviews that (i) were written from accounts with no user photo or with web plagiarized photos (identified through Google's image search), and that (ii) were short (less than 50 words). From this preliminary set, we have *manually* selected 200 generic reviews, that provide no venue specific information [25].

**Gold standard genuine reviews.** Given the seed user and venue datasets previously described, we have extracted a list of 202 genuine reviews satisfying a stringent test that consists of multiple checkpoints. In a first check we used Google (text and image search) to eliminate reviews with plagiarized text and reviewer account photos. In a second check we discarded short (less than 50 words), generic reviews, lacking references to the venue. Third, we gave preference to reviews written by users who

- Reached the "Elite" member status at least once.
- Participated in forums e.g. Yelp Talk.
- Garnered positive feedback on their reviews.
- Provided well thought out personal information on their profile.

**Large Yelp Data Set.** We have used YCrawl to collect the data of 7,435 car repair shops, beauty & spa centers and moving companies from San Francisco, New York City and Miami. The collection process took 3 weeks.

| Notation | Definition |
|---|---|
| $\mathcal{A}$ | Adversary |
| $V$ | Target venue |
| $H_V$, $\Delta T$ | $V$'s timeline and active interval |
| $\rho_V(T)$ | Rating of $V$ at time $T$ |
| $\delta r$ | Desired rating increase by $\mathcal{A}$ |
| $\delta t$ | Review campaign duration |
| $q$ | Number of fraudulent reviews by $\mathcal{A}$ |
| $R$, $R.\rho$, $R.\tau$ | Review, its rating and its posting time |
| $n$ | Number of genuine reviews of $V$ |
| $\sigma$ | Sum of ratings of all genuine reviews |
| $p$ | Number of genuine positive reviews |

Table 1: Table of Notations

Of the 7,345 venues, 1928 had no reviews posted. We have collected all their 270,121 reviews and the data of their 195,417 reviewers (one user can review more than 1 of these venues). Table 5 shows the number of venues collected for each venue type and city. Yelp limits the results for a search to the first 1000 matching venues. Entries with values less than 1000 correspond to cities with fewer than 1000 venues of the corresponding type.

## 4 Marco: Proposed Methods

We present Marco, a system for automatic detection of fraudulent reviews, deceptive venues and impactful review campaigns. We begin with a description of the adversary and his capabilities.

**4.1 Adversarial Model.** We model the attacker following the corrupt SEO (Search Engine Optimization) model mentioned in the introduction. The attacker $\mathcal{A}$ receives a contract concerning a target venue $V$. $\mathcal{A}$ receives a finite budget, and needs to "adjust" the rating of $V$, i.e., either increase or decrease it by at least half a star.

We assume $\mathcal{A}$ controls a set of unique (IP address, Yelp Sybil account) pairs and has access to a market of review writers. Sybil accounts [26] are different Yelp identities controlled by $\mathcal{A}$. $\mathcal{A}$ uses these resources to launch a "review campaign" to bias the rating of $V$: post one review from each controlled (IP address, Yelp Sybil account) pair and/or hire (remote) review writers, with valid Yelp accounts, to do it.

The number of reviews $\mathcal{A}$ can post is limited by the number of unique (IP address, Yelp Sybil account) pairs it controls as well as by the budget received in the contract (minus $\mathcal{A}$'s fee) divided by the average cost of hiring a review writer.

**4.2 Overview of Marco.** Marco, whose functionality is illustrated in Figure 1, consists of 3 primary modules. The Review Spike Detection (RSD) module relies on temporal, inter-review relations to identify venues receiving suspiciously high numbers of positive (or negative) reviews. The Aggregate Review Disparity (ARD) module uses relations between review ratings and the aggregate rating of their venue, at the time of their posting, to identify venues that exhibit a "bipolar" review behavior. The Fraudulent Review Impact (FRI) module first classifies reviews as fraudulent or genuine based on their social, spatial and temporal features. It then identifies venues whose aggregate rating is significantly impacted by reviews classified as fraudulent. Each module produces several features that feed into a venue classifier, trained on the datasets of Section 3.2. Table 1 shows the notations used by Marco.

**4.3 Review Spike Detection (RSD) Module.** A review campaign needs to adjust (e.g., increase) the rating of its target venue, by posting (fraudulent) reviews that compensate the negative ratings of other reviews. The RSD module detects this behavior by identifying venues that receive higher numbers of positive (or negative) reviews than normal.

In the following, our first goal is to prove that review campaigns that impact the ratings of their target venues are detectable. For this, let $q$ denote the total number of fraudulent reviews that $\mathcal{A}$ posts for the target venue $V$. We focus on the typical scenario where an attacker attempts to increase the rating of $V$ (ballot stuffing). Attempts to reduce the rating of $V$ (bad mouthing) are similar and omitted here for brevity.

Let $T_s$ and $T_e$ denote the start and end times of the campaign, the times when the first and last fraudulent reviews initiated by $\mathcal{A}$ are posted. $\delta t = T_e - T_s$ is the campaign duration interval. Let $n$ denote the number of genuine reviews $V$ has at the completion of the campaign (time $T_e$). We prove the following lower bound on the number of reviews that $\mathcal{A}$ needs to write in order to impact the rating of $V$.

**Claim 1.** *The minimum number of reviews $\mathcal{A}$ needs to post in order to (fraudulently) increase the rating of $V$ by half a star is $q = n/7$.*

*Proof.* Let $R_1, R_2, .., R_n$ denote the $n$ genuine reviews of $V$. Let $\sigma = \sum_{i=1}^{n} R_i.\rho$. According to Yelp semantics, $R_i.\rho \in [1, 5]$, thus $\sigma \in [n, 5n]$. The "genuine" rating of $V$ is $\rho_V^g = \frac{\sigma}{n}$. In order to minimize $q$, $\mathcal{A}$ has to write only 5 star reviews. Let $\delta r$ be the increase in the rating of $V$ generated by $\mathcal{A}$'s review campaign. Note that $\delta r \in [0.5, 4)$. Furthermore, $\frac{\sigma}{n} + \delta \leq 5$, as the final rating of $V$ cannot exceed 5. Hence,

$$\frac{\sigma + 5q}{n + q} = \frac{\sigma}{n} + \delta r,$$

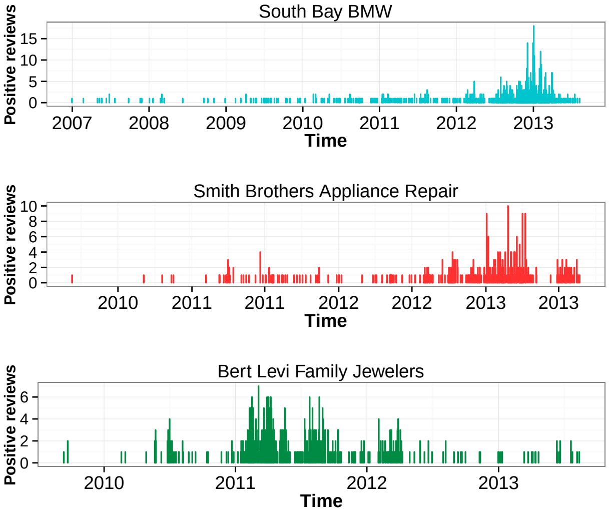


Figure 2: Timelines of positive reviews of 3 deceptive venues (see Section 3.2). Each venue has several significant spikes in its number of daily positive reviews.

Thus, $q = \frac{n^2\delta r}{5n-\sigma-n\delta r}$. Given that $\sigma \geq n$, we have $q \geq \frac{n\delta r}{4-\delta r}$. When $\delta r = 1/2$, this results in $q \geq n/7$. For $\delta r = 1$, $q \geq n/3$, when $\delta r = 2$, $q \geq n$, etc.

We say a review campaign is *successful* if it increases the rating of the target venue by at least half a star ($\delta r \geq 1/2$). We introduce now the notion of venue timeline:

**Definition 1.** *The timeline of a venue $V$ is the set of tuples $H_V = \{(U_i, R_i)|i = 1..n\}$, the list of reviews $R_i$ received by $V$ from users $U_i$, chronologically sorted by the review post time, $R_i.\tau$. Let $\Delta T = T_c - T_1$ denote the **active interval** of the venue, where $T_c$ denotes the current time and $T_1 = R_1.\tau$.*

Figure 2 illustrates this concept, by showing the evolution of the positive review (4 and 5 star) timelines of 3 venues selected from the ground truth deceptive venue dataset (see Section 3.2). Let $p$ denote the number of positive reviews received by $V$ during its active interval, $\Delta T$. We now show that:

**Claim 2.** *Assuming a uniform arrival process for positive reviews, the maximum number of positive reviews in a $\delta t$ interval is approximately $\frac{p\ \delta t}{\Delta T}(1 + \frac{1}{\sqrt{c}})$, where $c = \frac{p\ \delta t}{\Delta T\ log\frac{\Delta T}{\delta t}}$.*

*Proof.* The distribution of reviews into $\delta t$ intervals follows a balls and bins process, where $p$ is the number of balls and $\Delta T/\delta t$ is the number of bins. It is known (e.g., [27, 28]) that given $b$ balls and $B$ bins, the maximum number of balls in any bin is approximately $\frac{b}{B}(1 + \frac{1}{\sqrt{c}})$, where $c = \frac{b}{B\ logB}$. Thus, the result follows.

We introduce now the following result.

**Theorem 1.** *If $n > 49$, a successful review campaign will exceed, during the attack interval, the maximum number of reviews of a uniform review distribution.*

*Proof.* Let $p$ denote the number of positive, genuine reviews received by the target venue at the end of the review campaign. $p < n$, where $n$ is the total number of genuine reviews at the end of the campaign. According to Claim 1, a successful review campaign needs to contain at least $n/7$ positive (5 star) reviews. Then, since the expected number of positive genuine reviews to be received in a $\delta t$ interval will be $\frac{p\delta t}{\Delta T}$, following the review campaign, the expected number of (genuine plus fraudulent) positive reviews in the attack interval will be $\frac{n}{7} + \frac{p\delta t}{\Delta T}$.

The maximum number of positive genuine reviews posted during an interval $\delta t$, assuming a uniform distribution, is, according to Claim 2, approximately $\frac{p\ \delta t}{\Delta T} + \sqrt{\frac{p\delta t\ \log\frac{\Delta T}{\delta t}}{\Delta T}}$. Thus, the number of positive reviews generated by a review campaign exceeds the maximum positive reviews of a uniform distribution if

$$\frac{n}{7} + \frac{p\delta t}{\Delta T} > \frac{p\delta t}{\Delta T} + \sqrt{\frac{p\delta t\ \log\frac{\Delta T}{\delta t}}{\Delta T}}.$$

Since $n > p$, this converts to $\frac{n}{49} > \frac{\log\frac{\Delta T}{\delta t}}{\frac{\Delta T}{\delta t}}$ Since $\Delta T > \delta t$, we have that $\frac{\log\frac{\Delta T}{\delta t}}{\frac{\Delta T}{\delta t}} < 1$. Thus, the above inequality trivially holds for $n > 49$.

**Detect abnormal review activity.** We exploit the above results and use statistical tools to retrieve ranges of abnormal review activities. In particular, our goal is to identify spikes, or outliers in a venue's timeline. For instance, each venue in Figure 2 has several significant review spikes. The RSD module of Marco uses the measures of dispersion of Box-and-Whisker plots [22] to detect outliers. Specifically, given a venue $V$, it first computes the quartiles and the inter-quartile range IQR of the positive reviews from $V$'s timeline $H_V$. It then computes the upper outer fence ($UOF$) value using the Box-and-Whiskers plot [22]. For each sub-interval $d$ of set length (in our experiments $|d| = 1$ day) in $V$'s active period, let $P_d$ denote the set of positive reviews from $H_V$ posted during $d$. If $|P_d| > UOF$, the RSD module marks $P_d$, i.e., a spike has been detected. For instance, the "South Bay BMW" venue (see Figure 2) has a $UOF$ of 9 for positive reviews: any day with more than 9 positive reviews is considered to be a spike.

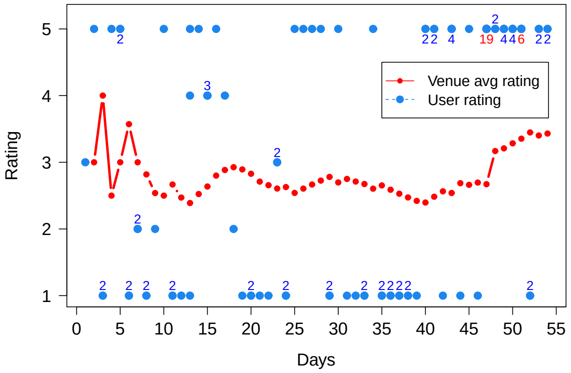


Figure 3: Evolution in time of the average rating of the venue "Azure Nail & Waxing Studio" of Chicago, IL, compared against the ratings assigned by its reviews. The values in parentheses denote the number of reviews that were assigned a corresponding rating (shown on the y axis) during one day. The lack of consensus between the many low and high rated reviews raises a red flag.

The RSD module outputs two features (see Table 3): $SC(V)$, the number of spikes detected for a venue $V$, and $SAmp(V)$, the amplitude of the highest spike of $V$, normalized to the average number of reviews posted for $V$ during an interval $d$.

### 4.4 Aggregate Rating Disparity (ARD).

A venue that is the target of a review campaign is likely to receive reviews that do not agree with its genuine reviews. Furthermore, following a successful review campaign, the venue is likely to receive reviews from genuine users that do not agree with the venue's newly engineered rating.

Let $\rho_V(T)$ denote the average rating of a venue $V$ at time $T \in [T_1, T_c]$. We define the rating disparity of a review $R$ written at time $R.\tau$ for $V$ to be the divergence of $R$'s rating from the average rating of $V$ at the time of its posting, $|R.\rho - \rho_V(R.\tau)|$. Let $R_1, .., R_N$, $N = n + q$, be all the reviews received by $V$ (both genuine and fraudulent) during its active interval $\Delta T$. We define the aggregate rating disparity score of $V$ to be the average rating disparity of all the reviews of $V$:

$$ARD(V) = \frac{\sum_{i=1}^{N} |R_i.\rho - \rho_V(R_i.\tau)|}{N}$$

By influencing the average rating of a venue, a review campaign will increase the rating disparity of both fraudulent and of genuine reviews. This is illustrated in Figure 3, that plots the evolution in time of the average rating against the ratings of individual reviews received by the "Azure Nail & Waxing Studio" (Chicago, IL). The positive reviews (1 day has a spike of 19, 5-star reviews, shown in red in the upper right corner) disagree with the low rated reviews, generating a high ARD value. The ARD module contributes one feature, the $ARD$ score, see Table 3.

| Notation | Definition |
|---|---|
| $f(U)$ | The number of friends of $U$ |
| $r(U)$ | The number of reviews written by $U$ |
| $Exp_U(V)$ | The expertise of $U$ around $V$ |
| $c_U(V)$ | The number of check-ins of $U$ at $V$ |
| $p_U(V)$ | The number of photos of $U$ at $V$ |
| $feedback(R)$ | The feedback count of $R$ |
| $Age_U(R)$ | Age of $U$'s account when $R$ was posted |

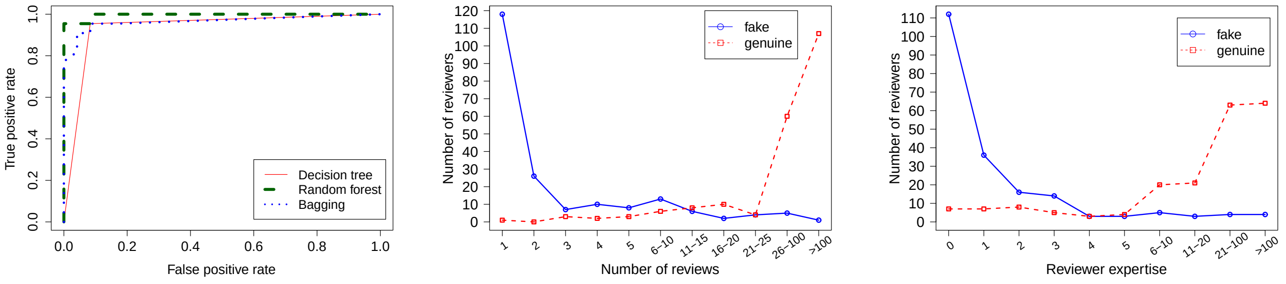
Table 2: Features used to classify review $R$ written by user $U$ for venue $V$.

### 4.5 Fraudulent Review Impact (FRI) Module.

Venues that receive few genuine reviews are particularly vulnerable to review campaigns (see also Theorem 1). Furthermore, long term review campaigns that post high numbers of fraudulent reviews can re-define the "normal" review posting behavior, flatten spikes and escape detection by the RSD module. They are also likely to drown the impact of genuine reviews on the aggregate rating of the venue. Thus, the ARD of the campaign's target venue will be small, controlled by the fraudulent reviews.

We propose to detect such behaviors through fraudulent reviews that significantly impact the aggregate rating of venues. For this, in a first step, the FRI module uses machine learning tools to classify the reviews posted for $V$ as either fraudulent or genuine. It uses features extracted from each review, its writer and the relation between the review writer and the target venue (see Table 2). Specifically, let $R$ denote a review posted for a venue $V$, and let $U$ denote the user who wrote it. In addition to the friend and review count of $U$, we introduce the concept of *expertise* of $U$ around $V$. $Exp_U(V)$ is the number of reviews $U$ wrote for venues in the vicinity (50 mile radius) of $V$. Furthermore, FRI uses the number of activities of $U$ recorded at $V$, the feedback of $R$, counting the users who reacted positively to the review, and the age of $U$'s account when $R$ was posted, $Age_U(R)$. Section 5.1 shows that the Random Forest tool achieves 94% accuracy when classifying fraudulent and genuine reviews.

In a second step, the FRI module introduces the notion of *fraudulent review impact*, to model the impact of fraudulent reviews on the final rating of the venue. Let $\rho^g_V = \frac{\sigma}{n}$ denote the genuine rating of $V$, computed as an average over its $n$ genuine reviews. Then, $FRI(V) = \rho_V(T_c) - \rho^g_V$, where $\rho_V(T_c)$ is the average rating of $V$ at current time $T_c$. Note that $FRI(V)$ can be negative, for a bad-mouthing campaign. The



(a) (b) (c)

Figure 4: (a) ROC plot of Random Forest (RF), Bagging and C4.5 Decision Tree (DT) for review classification (200 fraudulent, 202 genuine). RF performs best, at 94% accuracy. (b) Distribution of reviewers' review count: fraudulent vs. genuine review sets. (c) Distribution of reviewers' expertise levels: fraudulent vs. genuine sets. Note their symmetry: unlike genuine reviewers, fraudulent reviewers tend to have written only few reviews and have low expertise for the venues that they reviewed.

| Notation | Definition |
|---|---|
| $SC(V)$ | The number of review spikes for $V$ |
| $SAmp(V)$ | The amplitude of the highest spike |
| $ARD(V)$ | Aggregate rating disparity |
| $FRI(V)$ | The fraudulent review impact of $V$ |
| $CF(V)$ | Count of reviews classified fraudulent |
| $\rho_V$ | The rating of $V$ |
| $N$ | The number of reviews of $V$ |
| $cir(V)$ | The number of reviews with check-ins |
| $pr(V)$ | The number of reviews with photos |
| $Age(V)$ | The age of $V$ |

Table 3: Features used to classify a venue $V$ as either deceptive or legitimate.

FRI module contributes two features, $FRI(V)$, and the percentage of reviews classified as fraudulent for $V$, $CF(V)$ (see Table 3).

**4.6 Venue Classification.** In addition to the features provided by the RSD, ARD and FRI modules, we also use the rating of $V$, $\rho_V$, its number of reviews $N$, its number of reviews with associated user check-ins, $cir(V)$, and with uploaded photos, $pr(V)$, and the current age of $V$, $Age(V)$, measured in months since $V$'s first review. Table 3 lists all the features we selected. Section 5.2 shows that the features enable the Random Forest classifier to achieves 95.8% accuracy when classifying the venue sets of Section 3.2.

## 5 Empirical Evaluation

In this section we show that Marco is scalable as well as efficient in detecting fraudulent reviews and deceptive venues. We have implemented Marco using (i) Python, to extract data from parsed pages and compute the proposed features, (ii) the statistical tool R, to classify reviews and venues. We used MySQL to store collected data and features.

**5.1 Review Classification.** We investigated the ability of the FRI module to classify reviews, when using 5 machine learning tools: Bagging, $k$-Nearest Neighbor (kNN), Random Forest (RF), Support Vector Machines (SVM) and C4.5 Decision Trees (DT). We used 10-fold cross-validation over the 200 fraudulent and 202 genuine reviews of Section 3.2. Figure 4a shows the receiver operating characteristic (ROC) curve for the top 3 performers: RF, Bagging and DT.

The overall accuracy ($\frac{TPR+TNR}{TPR+TNR+FPR+FNR}$) of RF, Bagging and DT is 94%, 93.5% and 93% respectively. TPR is the true positive rate, TNR is the true negative rate, FPR the false positive rate and FNR the false negative rate. The (FPR, FNR) pairs for RF, Bagging and DT are (7%,5%),(8%,5%) and (8%,6%) respectively. In the remaining experiments, the FRI module of Marco uses the RF classifier.

The top 2 most impactful features for RF are $r(U)$ and $Exp_U(V)$. Figure 4b compares the distribution of the $r(U)$ feature for the 200 fraudulent and the 202 genuine reviews. We emphasize their symmetry: few fraudulent review writers posted a significant number of reviews, while few genuine review writers posted only a few reviews. Figure 4c compares the distribution of the $Exp_U(V)$ measure. The distributions are also almost symmetric: most writers of genuine reviews have written at least 4 reviews for other venues in the vicinity of the venue of their selected review.

**5.2 Venue Classification.** We have used 10-fold cross-validation to evaluate the ability of Marco to classify the 90 deceptive and 100 legitimate venues of Section 3.2. Figure 5 shows the ROC curve for Marco when using the RF, Bagging and C4.5 DT

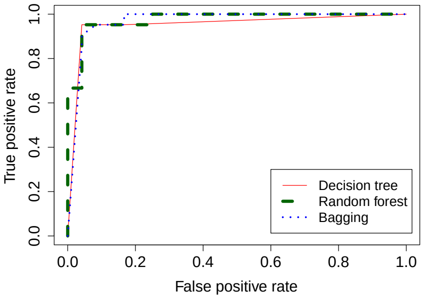


Figure 5: ROC plot of RF, Bagging and C4.5 DT for the 90 deceptive/100 legitimate venue datasets. RF and DT are tied for best accuracy, of 95.8%.

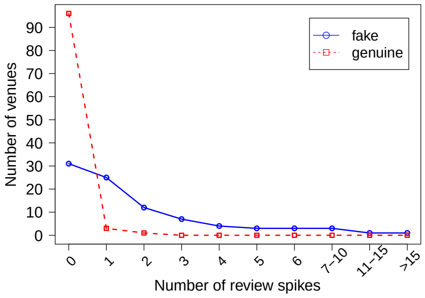


Figure 6: Distribution of SC(V), for the 90 deceptive and 100 legitimate venues. 60 deceptive venues have at least one review spike. 1 legitimate venue has 1 spike.

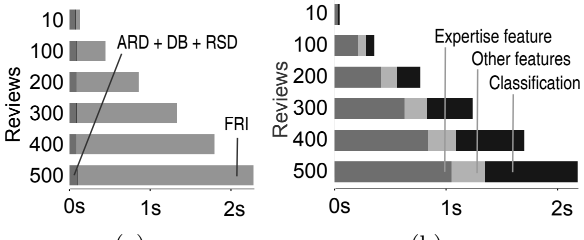
classifiers on the features listed in Table 3. The overall accuracy for RF, Bagging and DT is 95.8%, 93.7% and 95.8% respectively, with the corresponding (FPR,FNR) pairs being (5.55%,3%),(8.88%,4%) and (5.55%,3%) respectively.

Figure 6 shows the distribution of $SC(V)$ for the 190 venues. Only 1 legitimate venue has a review spike, while several deceptive venues have more than 10 spikes. Furthermore, 26 deceptive venues have an FRI score larger than 1; only 1 legitimate venue has an FRI larger than 1.

**Comparison with state-of-the-art.** We compared Marco with the three deceptive venue detection strategies of Feng et al. [19], $avg\Delta$, $dist\Phi$ and $peak \uparrow$. Table 4 shows the FPR, FNR and overall accuracy of Marco, $avg\Delta$, $dist\Phi$ and $peak \uparrow$. Marco achieves a significant accuracy improvement (95.8%) over $dist\Phi$, the best strategy of Feng et al. [19] (72.1%).

| Strategy | FPR | FNR | Accuracy |
|---|---|---|---|
| Marco/RF | $5/90 = 0.055$ | $3/100 = 0.3$ | 95.8% |
| $avg\Delta$ | $33/90 = 0.36$ | $31/100 = 0.31$ | 66.3% |
| $dist\Phi$ | $28/90 = 0.31$ | $25/100 = 0.25$ | 72.1% |
| $peak \uparrow$ | $41/90 = 0.45$ | $37/100 = 0.37$ | 58.9% |

Table 4: Marco vs. the three deceptive venue detection strategies of Feng et al. [19]. Marco shows over 23% accuracy improvement over $dist\Phi$.



(a) (b)

Figure 7: (a) Marco's per-module overhead: FRI is the most expensive, but under 2.3s even for venues with 500 reviews. (b) Zoom-in of FRI module overhead. Computing the $Exp_U(V)$ feature takes the most time.

| City | Car Shop | Mover | Spa |
|---|---|---|---|
| Miami, FL | 1000 (6) | 348 (8) | 1000 (21) |
| San Fran., CA | 612 (59) | 475 (45) | 1000 (42) |
| NYC, NY | 1000 (8) | 1000 (27) | 1000 (28) |

Table 5: Collected venues organized by city and venue type. Values between parentheses show the number of venues detected by Marco to be deceptive. San Francisco has the highest percentage of deceptive venues.

### 5.3 Marco in the Wild.

Marco takes only a few seconds to classify a venue, on a i5@2.4GHz, 4GB of RAM Dell laptop. Figure 7a shows the per-module overhead of Marco (averages over 10 experiment runs), as a function of the review count of the venue classified. While the FRI module is the most time consuming, even for venues with 500 reviews the FRI overhead is below 2.3s. The RSD and ARD modules impose only a few ms (6ms for 500 reviews), while DB access and data retrieval take around 90ms. Figure 7b zooms-in into the FRI overhead. For 500 reviews, the most time consuming components are computing the user expertise, $Exp_U(V)$ ($\approx 1.1$s), computing *all* the other features ($\approx 0.4$s) and classifying the reviews ($\approx 0.8$s).

We have used Marco to classify the 7,435 venues we collected from Miami, San Francisco and New York City. We have divided the set of 7,435 venues into subsets of 200 venues. We trained Marco on the 190 ground truth/gold standard venues and tested it separately on all subsets of 200 venues. Table 5 shows the total number of venues collected and the number of venues detected to be deceptive, between parentheses. San Francisco has the highest concentration of deceptive venues: Marco flags almost 10% of its car repair and moving companies as suspicious, and upon our manual inspection, they indeed seemed to engage in suspicious review behaviors. While the FRI of San Francisco's collected genuine venues is at most 1, 60% of its deceptive venues have an FRI between 1 and 4.

## 6 Conclusions

We presented Marco, a system for detecting deceptive Yelp venues and reviews, leveraging a suite of social, temporal and spatial signals gleaned from Yelp reviews and venues. We also contribute a large dataset of over 7K venues, 270K reviews from 195K users, containing also a few hundred *ground-truth* and *gold-standard* reviews (fraudulent/genuine) and venues (deceptive/legitimate). Marco is effective in classifying both reviews and venues, with accuracies exceeding 94%, significantly outperforming state-of-the-art strategies. Marco is also fast; it classifies a venue with 500 reviews in under 2.3s.

## References

[1] David Segal. A Rave, a Pan, or Just a Fake? the New York Times, www.nytimes.com/2011/05/22/your-money/22haggler.html, 2011.

[2] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 219–230, New York, NY, USA, 2008. ACM.

[3] Nitin Jindal, Bing Liu, and Ee-Peng Lim. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1549–1552, 2010.

[4] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 939–948, 2010.

[5] Yelp. http://www.yelp.com.

[6] Yelp admits a quarter of submitted reviews could be fake. BBC, www.bbc.co.uk/news/technology-24299742.

[7] Sponsored Reviews. www.sponsoredreviews.com/, Last accessed October 12, 2013.

[8] Posting Positive Reviews. postingpositivereviews.blogspot.com/, Last accessed October 12, 2013.

[9] Pay Per Post. https://payperpost.com/, Last accessed October 12, 2013.

[10] A.G. Schneiderman Announces Agreement With 19 Companies To Stop Writing Fake Online Reviews And Pay More Than $350,000 In Fines. www.ag.ny.gov/press-release/ag-schneiderman-announces-agreement-19-companies-stop-writing-fake-online-reviews-and.

[11] Michael Anderson and Jeremy Magruder. Learning from the crowd: Regression discontinuity estimates of the effects of an online review database. *Economic Journal*, 122(563):957–989, 2012.

[12] Michael Luca. Reviews, Reputation, and Revenue: The Case of Yelp.com. Available at hbswk.hbs.edu/item/6833.html.

[13] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, , and Natalie Glance. What yelp fake review filter might be doing. In *Proceedings of the International Conference on Weblogs and Social Media*, 2013.

[14] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the Int'l Conference on World Wide Web*, 2012.

[15] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the International Conference on World Wide Web*, pages 201–210. ACM, 2007.

[16] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 632–640, 2013.

[17] Fangtao Li, Minlie Huang, Yi Yang, Xiaoyan Zhu, and Xiaoyan Zhu. Learning to identify review spam. In *IJCAI*, pages 2488–2493, 2011.

[18] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Human Language Technologies (HLT)*, 2011.

[19] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM)*, 2012.

[20] Hide My Ass! Free Proxy and Privacy Tools. http://www.hidemyass.com/.

[21] Death By Captcha. www.deathbycaptcha.com/.

[22] A. C. Tamhane and D. D Dunlop. *Statistics and data analysis: From elementary to intermediate*. Upper Saddle River, NJ: Prentice Hall, 2000.

[23] Spelp. www.yelp.com/topic/boston-spelp-9.

[24] Flelp. www.yelp.com/topic/miami-flelp-we-rock.

[25] 3 Tips for Spotting Fake Product Reviews - From Someone Who Wrote Them. MoneyTaksNews, www.moneytalksnews.com/2011/07/25/3-tips-for-spotting-fake-product-reviews—from-someone-who-wrote-them.

[26] John R. Douceur. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002.

[27] Petra Berenbrink, André Brinkmann, Tom Friedetzky, and Lars Nagel. Balls into bins with related random choices. In *Proceedings of the Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2010.

[28] Martin Raab and Angelika Steger. Balls into Bins" - A Simple and Tight Analysis. In *Proceedings of Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 159–170, 1998.