Think Outside the Dataset: Finding Fraudulent Reviews using Cross-Dataset Analysis

Shirin Nilizadeh
University of Texas, at Arlington
shirin.nilizadeh@uta.edu

Hojjat Aghakhani
University of California Santa Barbara
hojjat@ucsb.edu

Eric Gustafson
University of California Santa Barbara
edg@ucsb.edu

Christopher Kruegel
University of California Santa Barbara
chris@ucsb.edu

Giovanni Vigna
University of California Santa Barbara
vigna@ucsb.edu

ABSTRACT

While online review services provide a two-way conversation between brands and consumers, malicious actors, including misbehaving businesses, have an equal opportunity to distort the reviews for their own gains. We propose OneReview, a method for locating fraudulent reviews, correlating data from multiple crowd-sourced review sites. Our approach utilizes *Change Point Analysis* to locate points at which a business' reputation shifts. Inconsistent trends in reviews of the same businesses across multiple websites are used to identify *suspicious* reviews. We then extract an extensive set of textual and contextual features from these suspicious reviews and employ supervised machine learning to detect *fraudulent* reviews.

We evaluated OneReview on about 805K and 462K reviews from Yelp and TripAdvisor, respectively to identify fraud on Yelp. Supervised machine learning yields excellent results, with 97% accuracy. We applied the created model on suspicious reviews and detected about 62K fraudulent reviews (about 8% of all the Yelp reviews). We further analyzed the detected fraudulent reviews and their authors, and located several spam campaigns in the wild, including campaigns against specific businesses, as well as campaigns consisting of several hundreds of socially-networked untrustworthy accounts.

CCS CONCEPTS

• **Information systems** → **Trust**; **Reputation systems**.

KEYWORDS

Review Websites; Fraudulent Reviews and Campaigns; Cross-Dataset Analysis; Change-Point Analysis;

ACM Reference Format:
Shirin Nilizadeh, Hojjat Aghakhani, Eric Gustafson, Christopher Kruegel, and Giovanni Vigna. 2019. Think Outside the Dataset: Finding Fraudulent Reviews using Cross-Dataset Analysis. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308558.3313647

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313647

1 INTRODUCTION

Many crowd-sourced review platforms, such as Yelp, TripAdvisor, Google and Foursquare, provide a shared space for people to write reviews and rate local businesses. With the substantial impact of businesses' online ratings on their selling [4], many businesses add themselves to multiple websites to more easily be discovered. Some might also engage in reputation management, which could range from rewarding their customers for a favorable review, or a complex review campaign, where armies of accounts post reviews to influence a business' average review score.

Several previous works have attempted to address this problem. Most of them use supervised machine learning, and only focus on textual and stylometry features [1, 16, 19, 37]. Their obtained ground truth data is also not large and comprehensive [19, 25, 27, 37, 39, 51]. Finally, these works assume a limited attacker model, e.g., an attacker's activity is assumed to be found near sudden shifts in the data [39], or they only try to detect positive campaigns.

In this paper, we propose OneReview, a system for finding fraudulent content on a crowd-sourced review site, leveraging correlations with other independent review sites, and the use of textual and contextual features. Our system leverages the intuition that an attacker may not be able to exert the same influence over a business' reputation on several websites, due to increased cost, with reviews costing between $10 and $25 each [5, 6, 45], and must be customized for each site. Even in the case of machine-generated reviews explored in [47, 49], the text generated by these systems may be customized, but the supporting metadata is not. Even when reviews can be purchased as a service [41, 44], these services charge more to target additional review sites.

OneReview focuses on isolating anomalous changes in a business' reputation across multiple review sites, to locate malicious activity without relying on specific patterns. Our intuition is that a business's reputation should not be very different in multiple review sites; *e.g.,* if a restaurant changes its chef or manager, then the impact of these changes should appear on reviews across all the websites. OneReview utilizes *Change Point Analysis* method on the reviews of every business independently on every website, and then uses our proposed *Change Point Analyzer* to evaluate change-points, detect those that do not match across the websites, and identify them as suspicious. Then, it uses supervised machine learning, utilizing a combination of textual and metadata features to locate fraudulent reviews within the set of suspicious reviews.

We evaluated our approach, using data from two review websites, namely Yelp and TripAdvisor, to find fraudulent activity on Yelp. We

obtained Yelp reviews through the Yelp Data Challenge [50], and used our Change Point Analyzer to correlate this with data crawled from TripAdvisor. Since realistic, varied ground truth data is not currently available, we used a combination of our change point analysis and crowd-labeling to create a set of 5,655 labeled reviews. We used k-cross validation (k=5) on our ground truth and obtained 97% (+/- 0.01) accuracy, 91% (+/- 0.03) precision and 90% (+/- 0.06) recall. Then the model was used on the suspicious reviews, which classified 61,983 reviews, about 8% of all reviews, as fraudulent.

We further detected fraudulent campaigns that are actively initiated by, or targeted toward specific businesses. We identified 3,980 businesses with fraudulent reviews, as well as, 14,910 suspected spam accounts, where at least 40% of their reviews are classified as fraudulent. We also used community detection algorithms to locate several large astroturfing campaigns. These results show the effectiveness of OneReview in detecting fraudulent campaigns.

2 RELATED WORK

Text and Metadata Features. Previous works have explored a multitude of combinations of machine learning techniques and features to help locating fraudulent reviews. Most works rely only on textual [16, 25, 37, 39, 51], and stylometry [16] features. Jindal and Liu [19] first used meta-data features to detect fraudulent reviews on Amazon. OneReview uses some of the suggested features if applicable. Considerable work tried to identify the spam accounts, using regression models [26], heterogeneous graphs [48], unsupervised anomaly detection [22, 31, 32, 46], and behavioral models [32, 33]. OneReview uses user-related metadata features, but does not make any such assumptions about the users in question.

Using Temporal Data for Fraud Detection. Some works [15, 24, 27, 39] take advantage of temporal and spatial information in their detection algorithms. Li et al. [24] observed some correlations between the time of writing reviews on Dianping, and the location of their authors, with the spam reviews. Others use bursts in the number of reviews to locate suspicious patterns [15, 39, 39]. This approach has two limitations compared to OneReview. First, OneReview can distinguish if reviews that have significant impact on the overall score of the business are actually suspicious, by comparing the pattern on the other website, eliminating potential false positives. Second, our threat model considers both the overall review score, as well as the more recent effects of reviews.

Ground Truth Creation. One main obstacle in detecting fake reviews is the absence of ground truth. Some early work leverages duplicate reviews as the source of fraudulent labeled data [19]. Others explore asking humans to write deceptive reviews [25, 37, 51]. However, the ground truth dataset that has been used in these methods might not reflect the dynamics of fraudulent reviews in a commercial website [34]. Rahman et. al [39] leveraged some forums where Elite Yelp users reveal and initiate the discussion on fraudulent Yelp reviews. Unfortunately, these forums are no longer available. In contrast, our dataset (see Section 6) contains data that is user-labeled, not user-generated, and additionally contains reviews chosen based on social graph information, and duplicate reviews.

3 THREAT MODEL

We assume that the adversary wishes to modify the reputation of a business, either positively or negatively, through any means, including both the overall reputation score and its recent reviews. We assume the adversary can utilize any functionality or behavior available to a normal, registered user of the service, including creating accounts, posting reviews, adding social connections, and so on. We also assume the adversary may compromise the accounts of legitimate users, or hire some users to post fraudulent reviews [6, 30, 41, 44].

An active adversary may know of the deployment of a system such as OneReview, and actively attempt to avoid detection by cross-posting the same ratings on all the crowd-sourced review websites over similar time spans. However, there is a trade-off between costs and benefits. Since the adversary needs to avoid detection by each of these review websites, as well as detection by OneReview itself, then it needs to employ complex and expensive techniques equally on all the websites which makes it unprofitable.

4 DATA

In this paper we prototype OneReview with data from two widely popular crowd-review sites, Yelp and TripAdvisor. Both sites allow users to search for businesses in their area; the page for each business offers users an overview of its basic information, reputation, and a list of reviews. Both allow users to submit reviews with "star" ratings. Users can provide feedback about the reviews themselves, such as Yelp's "funny" or "useful" ratings. Both also offer a ranking system for users (Yelp's "Elite", or TripAdvisor's "TripCollective").

Yelp Dataset. We obtained Yelp reviews from the dataset released for the 9th Yelp Data Challenge [50]. For more easily demonstrating our system, we only examine the reviews on restaurants that are located in Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland. Our Yelp dataset includes data for 1.4M reviews for about 16K restaurants and 469K users. It also includes metadata about businesses, reviews, and users.

TripAdvisor Dataset. In March 2017, we crawled TripAdvisor, and extracted public reviews and reviewer information about restaurants in the same seven US cities we selected in the Yelp dataset, resulting in about 656K reviews, submitted by 305K users, for 10K businesses. As with Yelp, data from TripAdvisor includes various metadata about each review, reviewer, and business.

Comparing the overall star ratings, we found that businesses on TripAdvisor obtain higher average star ratings (4.11) than those on Yelp (3.76). *The difference between the overall star ratings suggests that when using two sources of data, it might not be helpful (and it can be even misleading) to directly compare the ratings. However, it can be helpful to compare the trends in this data.*

5 ONEREVIEW SYSTEM DESIGN

OneReview is a system to detect *fraudulent reviews* and campaigns in crowd-sourced review websites through comparing trends with other sites, and machine learning. Figure 1 illustrates an overview of OneReview, including: (1) obtaining data, (2) matching businesses, (3) identifying inconsistent and suspicious change points in *star* trends, (4) extracting textual and contextual features, (5) employing

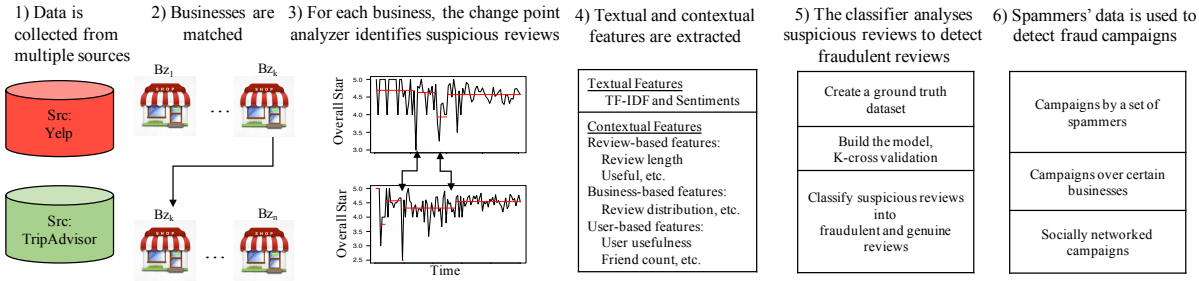

Figure 1: OneReview System Overview.

supervised machine learning to detect fraudulent reviews, and (6) detecting fraud campaigns. We explain each step in more details.

5.1 Linking Multiple Datasets

In crowd-sourced review websites, usually the name and address of businesses are not complete, which makes it not trivial matching a business between multiple websites. Our matching algorithm has two steps. Consider a restaurant in site A, called b_A, located in city c. First, the address attributes of the restaurant are compared with those of all restaurants in a site B that are also located in the same city c. Any restaurant in site B with an address similar to b_A is added to the set of possible matches M. The addresses of two restaurants located in the same city are considered similar if: 1) their ZIP codes are equal or missing, 2) their house numbers are equal or missing, and, 3) the Jaccard similarity index between their street name strings is above a threshold τ_s, or any of their street names is missing. Second, the restaurant name for b_A is compared with the names of all the restaurants in M by computing the Jaccard similarity index between their restaurant name strings. The restaurant b_A is matched to a restaurant b_B in M with the maximum Jaccard similarity index, provided that it is above a threshold τ_n.

For our purpose, the mapping algorithm should have a *perfect precision* score to avoid wrong comparison of businesses. We sampled 700 restaurants (100 restaurants per city) from the two datasets described in Section 4, and for several values of τs manually checked the precision of the mappings. We found that providing $\tau_s = 0.3$ and $\tau_n = 0.3$, the precision of matched restaurants is 1.0. The smaller dataset, Tripadvisor, includes about 10K businesses, and we could successfully match about 60% of them to the Yelp businesses. We believe that over time, eventually all businesses would have precise profiles on all websites, which results in more coverage.

Our final Yelp dataset after matching 6,068 businesses includes 805,608 reviews, and 341,399 reviewers, while our final TripAdvisor dataset includes 462,820 reviews, and 234,577 reviewers.

5.2 Comparing Trends

Intuitively, one would expect that the reputation of a business should be reasonably similar in several crowd-sourced websites. Therefore, we assume that at least the *trends* in terms of the overall ratings on multiple sites match over time, *i.e.,* within a time window, their ratings are either increasing or decreasing. OneReview analyzes the difference of trends by detecting change points in the time series that are created on reviews' overall scores. Change points are those data points at which the statistical properties such as mean, variance, correlation, or spectral density of a series of ratings change. For example, if the star ratings for a business in the months of January and February are $\{5, 5, 4, 5, 5\}$ and $\{2, 1, 5, 4\}$, respectively, then at least one change point is detected at end of month of January, because the ratings are suddenly dropped from 5s at the end of January to 2 and 1.[1]

Not all businesses receive reviews at the same rate, or at a constant rate over time, which can have an impact on the performance of Change Point Analysis. For example, based on our dataset, on Yelp less than 8% of restaurants receive daily reviews, while more than 80% obtain reviews monthly. To overcome this limitation, we compute the mean of overall ratings for each business per month, and use the sequence of these monthly mean values as the data points in the time series. While a period of one month might cause our system to miss short-lived campaigns, it is a good trade-off between detecting suspicious reviews that actually impact a business and not having effective change point analysis.

Change Point Detection. For each set of matched businesses, OneReview generates time-series from the original crawled data, one for a business' star ratings on Yelp and one for its ratings on TripAdvisor. These time-series are the inputs to the change point analysis. More formally, for each business, we have an ordered sequence of data points, $s_{1:n}^{src_1} = (s_1, \ldots, s_n)$, where s_i is the mean of stars in the i^{th} month in a data source, src_1. A change point occur at time $\tau \in \{1, \ldots, n-1\}$, such that the statistical properties of $\{s_1, \ldots, s_\tau\}$ and $\{s_{\tau+1}, \ldots, s_n\}$ are different in some way.

There are a variety of Change Point Analysis methods, differing in terms of which statistics they monitor changes in [7, 14, 21, 42, 43]. The most common method is MeanVar PELT [21], a multi-change point method, which leverages both mean and variance.

Change point analysis is a statistical analysis method. Shifts are detected as change points, if they are above certain thresholds and statistically significant. The number of detected change points then depends on the *penalty* parameter. A lower penalty value results in more change points being identified, *i.e.,* it is more sensitive. In our case study of Yelp and TripAdvisor, we tested several values including $\{p, log(n), 0.5 * log(n)\}$, where n is the number of data points in the time series and p is derived from an "elbow" plot [18].

Using MeanVar PELT method and penalty= p, $log(n)$ and $0.5 * log(n)$, we identified 26,835, 37,339 and 45,086 change points on the

[1]Note that depending on sensitivity threshold of the change point analysis method, more than one change point can be identified in these two months.

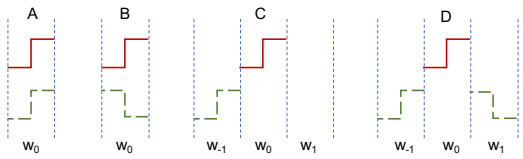

Figure 2: CPA identifies scenarios B and D as suspicious.

Yelp time-series, respectively. The set of change points identified with penalty=p is a subset of those identified with penalty $log(n)$ and $0.5 * log(n)$. The penalty can be used for tuning the sensitivity of OneReview. A more conservative system would choose the larger penalty value, which results in lower number of suspicious reviews.

In addition to identifying change points, OneReview also stores their directions, *i.e.,* positive or negative. We define a positive direction $d_{\uparrow}$ for a change point at time τ when the statistical value of $\{s_1, \ldots, s_\tau\}$ is smaller than that for $\{s_{\tau+1}, \ldots, s_n\}$. For example, a trend is considered positive, if the mean value of a business' star rating in a month is higher than its previous month. Similarly, a negative direction $d_{\downarrow}$ is defined for a change point at time τ when the statistical value of $\{s_1, \ldots, s_\tau\}$ is larger than that for $\{s_{\tau+1}, \ldots, s_n\}$.

Detecting Suspicious Change Points. We propose a Change Point Analyzer (CPA) that evaluates the change points detected in multiple websites with the goal of locating suspicious change points and reviews associated with them. The evaluation is based on several scenarios, from which four are shown in Figure 2.

First, consider at time window w_0, a change point is detected in both sources. If the direction of both change points are the same, both positive (scenario A in Figure 2) or negative, $d_{\uparrow}^{src_1}$ and $d_{\uparrow}^{src_2}$, or $d_{\downarrow}^{src_1}$ and $d_{\downarrow}^{src_2}$, then CPA labels them as *benign*. However, if their directions are not aligned, *i.e.,* one is positive and the other one is negative (scenario B in Figure 2) , $d_{\uparrow}^{src_1}$ and $d_{\downarrow}^{src_2}$, or $d_{\downarrow}^{src_1}$ and $d_{\uparrow}^{src_2}$, then they are labeled as *suspicious*.

Second, at time window w_0, a change point may be detected in src_1 but not in src_2 (scenarios C and D in Figure 2). This can indicate a suspicious behavior, however it also can be due to the sparsity of data in src_2. In other words, sometimes there is a delay in genuine change points simply because no review is provided for the business at some specific date. To minimize the false positives resulting from this phenomenon, when at time w_0 only a change point is detected in src_1 and not in src_2 , CPA extends the comparison to change points in neighbor time windows, w_{-1} and w_1. In particular, if in src_2, in time w_{-1}, any change point exists, then if the direction of any of the detected change points matches with that of the change point in src_1, then that change point in src_1 is labeled as *benign* (scenario C in Figure 2), otherwise as *suspicious* scenario D in Figure 2). This algorithm can be generalized for considering more than two review sources so that change points are labeled as *suspicious* if the majority of the change points differ in direction or do not exist. With penalty= p and $0.5 * log(n)$, CPA identified 26,835 and 61,817 change points as *suspicious*.

Finally, for every suspicious change point, OneReview retrieves the reviews contributed to that change point and label them as *suspicious*. Not all of the suspicious reviews are *fraudulent*; OneReview next employs machine learning using the textual and contextual features to identify fraudulent reviews within this set.

5.3 Feature Selection

For each review, a set of textual and contextual features from the review, its author, and the business are extracted. While inspired by previous work, we proposed some new features: "Author Overall Review Distribution," "Author Fresh Review Distribution," "Author Star Similarity," "Overall Author Usefulness," "Fresh Author Usefulness," and "Business star similarity". In this section, we use the following notations. For a review r, r_{text} and r_{date} are the text of r and the date it is posted. $A(r)$ and $B(r)$ indicate the author and business of r. $R(A)$ and $R(B)$ are reviews posted by A or written for B. A_f indicates "friends" of the author.

Review-based Features

n-grams. We extract bi-grams and uni-grams from all reviews and considered those with TF-IDF values higher than 0.5, which results in 168 uni-grams and about 6K bi-grams [40, 53].

Similarity Score. We use MinHash algorithm [23] to identify pairs of reviews that are similar to each other.

Sentiment. Li et al. [25] showed that over-emphasized sentiment offers robust cues to deceptive reviews. Using *NLTK* [8], each review is assigned a positivity, a negativity, and neutrality score.

Text Length. Previous works find that short reviews are more likely to be fraudulent [28]. Also, in our Yelp dataset, longer reviews are more likely to get *useful* votes. Text Length is defined as a logarithmic function of text length: $log(len(r_t) + 1)$.

Review Usefulness. It is defined as $\sqrt{r_{\text{useful}}}$, where r_{useful} is the number of useful votes received by others. Intuitively, *useful* votes are a gauge of the perceived review quality and trustworthiness.

Author-based Features

Author Overall Review Stars Distribution. We use Maximum Log Likelihood Estimation [2] to measure the similarity between distribution of ratings by an author, and all ratings.

Author "Fresh" Review Stars Distribution. Similar to the previous feature, it measures the similarity between distribution of all reviews stars and that of *fresh* reviews written by the author in the same month that the review has been posted.

Friend Count. This feature is defined as $log(|A(r)_f| + 1)$. Spam reviews are usually posted by users with fewer friends [28].

Elite Score. Yelp users can be recognized as "Elite" for each year they qualify. Elite status is intended to indicate that reviews are more helpful. We define this feature by $\sum_{e \in elites} \frac{1}{(|r_{\text{date}}.\text{year} - e| + 1)^2}$, where e is the year the user was awarded an Elite badge, $elites$ are all the Elite badges received by the author, and r_{date} is the review date. It measures the distance between the time of the review's publication and the user's Elite status being awarded.

Author Review Count. It is a logarithmic function of number of reviews by the author: $\log(|R(A)| + 1)$.

Author Star Similarity. It measures the similarity of a review's rating to the average ratings of all reviews by A: $|r_{stars} - \text{average}(R(A))|$.

Overall Author Usefulness. It measures the number of "useful" votes that an author has received: $\log(|\text{usefulness}(A)| + 1)$.

Fresh Author Usefulness. This feature measures the number of "useful" votes that an author has received for their *fresh* reviews.

Business-based Features

Business Overall Review Stars Distribution. This feature measures the similarity between distribution of all reviews stars and the distribution of reviews stars that have been written for B.

Business Fresh Review Stars Distribution. It compares distribution of all reviews with that of B's *fresh* reviews.

Business Star Similarity. The similarity of a review's rating to the average rating of all reviews for business B: $|r_{\text{stars}} - \text{average}(R(B))|$.

5.4 Classification

OneReview employs supervised machine learning using the features to classify a review as *fraudulent* or *benign*. Obtaining a ground truth of fraud on content containing people's opinions is a hard problem. We discuss our approach in Section 6. For classification, we chose Random Forests [11, 20]. This is due to its usefulness in a wide variety of applications [12], its resistance to over-fitting, and its utility in understanding feature importance [17]. For validating our classification, we apply k-fold cross validation, and use the resulting model to classify *fraudulent reviews*.

6 GROUND TRUTH DATASETS

There is no pre-existing, and reliable ground truth corpus of fraudulent reviews (discussed in Section 2). We employed a combination of human workers and algorithms to create a ground truth dataset of real-world fraudulent reviews. One major advantage of this dataset is that it includes both the *text* of the reviews and their *metadata*.

Obtaining Fraudulent Labeled Data

Our fraudulent ground truth dataset has 841 reviews, including:

379 Yelp "Not Recommended" reviews. Yelp employs a fraudulent detection mechanism to detect *fraudulent reviews* [35], and separate them as 'Not Recommended reviews.' We crawled Yelp on March 2017, about 2 months after Yelp dataset was published, and identified 1,341 reviews that since then Yelp had identified as Not Recommended. Yelp filters reviews for reasons beyond fraud, such as harassment, discrimination, or other offensive language. Therefore, we contribute 379 Not Recommended reviews that are also among suspicious reviews identified during change-point analysis.

370 duplicate reviews. We found 370 duplicate reviews that are posted for the same businesses, and are Yelp-to-Yelp matches. Interestingly, none were written by the same user, therefore, considered as fraudulent reviews. It is not trivial to accurately match users in Yelp to the users in TripAdvisor, and we could not validate if a duplicate review is written by the same user. As a result, we discarded all the 2,761 duplicate Yelp-to-TripAdvisor and TripAdvisor-to-Yelp duplicate reviews that are written for the same businesses. We do not consider identical reviews that have a length shorter than 100 characters, because they can be genuine identical short reviews.

92 crowd-labeled fraudulent reviews. To find reviews above and beyond those that are merely duplicates, or already labeled by Yelp's algorithm, we conduct a study using Amazon Mechanical Turk (AMT) to identify fraudulent reviews. Our research study was approved by the IRB of University of California, Santa Barbara.

Study Design. We created a pool of 1,700 *Human Intelligence Tasks* (HITs) [3]. Each HIT shows five different reviews to the worker, where they decide whether a review is *Strongly Fraudulent*, *Fraudulent*, *Cannot Tell*, *Benign*, and *Strongly Benign*. We only allow a single submission per Turker. Turkers are located in United States, and maintain an approval rating of at least 98% and have more than 10,000 approved HITs. To reduce the risk of dishonest AMT workers [38], we inserted verification questions into each HIT, asking the Turker to pick a specific option. Turkers are expected to spend 15-20 minutes on a HIT, and are paid 2.25 US dollar for an accepted submission. Reviews shown to the workers are sampled randomly. To give the workers some context, we provide them with additional supporting data, including review, author, and business information. We also provide a sample of four reviews posted for this business around the date of the review in question.

Study Results. We employed 1,837 Turkers, and had to discard 137 responses. Previous research [10, 36] has shown that manual labeling of opinions is not easy and human judgment is not particularly accurate at determining a fraudulent review. Therefore, we computed the majority vote by count of four, *i.e.,* if four Turkers identify a review *Strongly Fraudulent (Strongly Benign)* or *Fraudulent (Benign)*, then that review is labeled as *Fraudulent (Benign)*. Ultimately, from 1,700 reviews, we obtained 92 fraudulent reviews.

Obtaining Benign Labeled Data

Useful reviews by Elites. We use two of Yelp's meta-datas, including Elite and Usefulness. To be more cautious, we obtained reviews that are tagged as *useful* at least 3 times. We trust the reviews in the intersection of these two sets to be benign, $\text{Benign}_{\{\text{elite, useful}\}} = R_{\text{elite}} \cap R_{\text{useful}^{\geq 3}}$, which includes more than 46K reviews.

Reviews by trusted real-life acquaintances. We noticed that reviews that users find "useful," and users that Yelp marks as "Elite" share a bias towards longer, more detailed reviews, from very active accounts. Thus, we supplemented the dataset with reviews obtained using the authors' Yelp social connections, yielding 614 reviews.

Unbalanced Dataset

Prior work has shown that constructing labelled datasets, where the ratio of benign to malicious samples does not match that in practice can result in differences of more that an order of magnitude in the classification errors [29, 52]. Thus, we create a more realistic ground truth dataset with a genuine-to-fraudulent ratio of 4:1. We chose this ratio due to an estimation that the ratio of fake reviews on Yelp is 20% [28]. Therefore, we sampled 4,200 useful review by Elites. *Our final benign ground truth dataset is created by union of these samples,* $\text{Benign} = \text{Benign}_{\{\text{elite, useful}\}} \cup \text{Benign}_{\{\text{connections}\}}$, *including 4,814 reviews.*

7 EVALUATION

We show the effectiveness of our approach by statistical analysis. We first examined the classifier and the affect of Change Point Analysis on the results. We then apply OneReview on our dataset to detect and characterize real-world fraud campaigns.

Classification Performance. Table 1 shows the performance of the random forest classifier on our ground truth dataset. The classifier successfully detects fraudulent reviews with high accuracy (97%), precision (91%) and recall (90%). The standard error values (+/-) indicate that the performance is not drastically different for different number of trees or cross-folds. The reported results are with $k = 5$ folds, and 1000 trees. We picked Gini impurity to measure the quality of a split, and used bootstrapping when building trees. Table 1 also shows the performance of classifier with only textual features (TF-IDFs and sentiment features). The performance

Table 1: The performance of OneReview.

Features	Accuracy	F1-score	Precision	Recall
All	0.97 (+/- 0.01)	0.90 (+/- 0.04)	0.91 (+/- 0.03)	0.90 (+/- 0.06)
Textual	0.86 (+/- 0.01)	0.59 (+/- 0.05)	0.55 (+/- 0.05)	0.64(+/- 0.11)

is poor with 55% precision and 64% recall. We also tested various subsets of features however, none reached the same level of performance. The results illustrate that using all the contextual features substantially increases the performance. Since *similarity*, *review usefulness*, and *Elite* scores are used in the creation of the ground truth dataset, we did not use these features for classification. We overcome the limitation of unbalanced ground truth dataset, by using a well-known over-sampling technique called SMOTE [13].

Impact of Change Point Analysis. We tested our classifier on all the reviews in our Yelp dataset. It classified 375,359 reviews as fraudulent, which is equivalent to about 47% of all the 805K Yelp reviews. It is not reasonable to consider 47% of the reviews as fraudulent. However, from our MTurk study, we found that fraudulent reviews appear in suspicious reviews with higher probability. In particular, from 92 fraudulent reviews identified by Turkers, about 32% also appear in suspicious dataset while this dataset consists of only 24% of all the reviews. OneReview only performs the classification on the suspicious reviews to predict *fraudulent reviews*. As explained in Section 5.2, we obtained two sets of suspicious reviews with (1) $penalty = p$, and (2) $penalty = 0.5 * log(n)$, where the first set is a subset of the second. From 73K suspicious reviews in the first set, OneReview identified 61,983 fraudulent reviews, about 85% of the suspicious and 8% of all the 805K Yelp reviews. From 165K suspicious reviews in the second set, it identified 96,445 fraudulent reviews, about 58% of the suspicious and 12% of all the reviews. These results provide a lower and upper bound for the number of *fraudulent reviews* in Yelp. In the reminder of the paper, we focus on the *fraudulent reviews* from the smaller set of suspicious reviews.

Comparison with Yelp Reviews. We compared characteristics of detected *fraudulent reviews* and their authors with those of Yelp reviews and Not Recommended reviews. Due to space brevity, we do not provide the details of this comparison. In summary, this analysis showed that OneReview does not detect only one specific type of fraudulent review, it detects both positive and negative reviews, short and long reviews, those posted for both new and popular restaurants, and even those posted by Elite users.

Examining Fraud Campaigns. We refer to a *campaign* as one or more human actors controlling more than one account, with the aim of posting multiple reviews to influence the reputation of a business. These untrustworthy accounts can be bots, sybil accounts or individual spammers. We found that some untrustworthy accounts are created and used only once for a specific campaign while some are used several times for multiple campaigns.

Untrustworthy Accounts. We analyzed untrustworthy accounts, *i.e.,* Yelp users who have posted *fraudulent reviews*. In total, 58,157 untrustworthy accounts are identified (17% of total Yelp users). We found that about 8% of them (i.e., 7,064) have only posted one fake review. In the Yelp dataset, 27,347 of users only posted one review. Thus, OneReview does not simply identify every reviewer with one review as a untrustworthy account. Moreover, 14,910 of them are mainly posting fraudulent reviews with more than 40% of their reviews being fraudulent. This can indicate untrustworthy accounts who post legitimate reviews in an attempt to avoid the detection.

Campaigns Targeting Specific Businesses. Out of 6,068 businesses in our dataset, fraudulent reviews are posted for 3,980 businesses (about 66%). Some businesses have received many fraudulent reviews, *e.g,* 1,332 and 56 of businesses have received more than or equal to 10 and 100 fraudulent reviews, respectively. We further examined if these fraud campaigns are positive or negative. A fraudulent review is considered positive or negative if its rating is bigger or smaller than the overall rating of the business. Overall, the set of *fraudulent reviews* consists of 40,069 (65%) positive and 21,914 (35%) negative reviews. While it is expected that most campaigns are positive, it is interesting that still a good amount of *fraudulent reviews* are negative. Among 3,980 businesses with *fraudulent reviews*, 501 and 344 only have received positive and negative fraudulent reviews, respectively.

Socially-Networked Fraud Campaigns. A set of untrustworthy accounts can simply add each other as friends. OneReview does not consider the structure of the social network as a feature, still it can detect *fraudulent reviews* posted by fraud campaigns. We constructed the social network of untrustworthy accounts, which includes 16,738 nodes and 24,909 edges. We used the Louvain community detection algorithm [9], and found several communities. Some of these communities include several hundreds of nodes, *e.g.,* the top 5 largest communities include 1671, 1428, 1385, 565, and 539 of them. We found some patterns that can indeed distinguish them as untrustworthy accounts. For example, we found 40 untrustworthy accounts from a campaign which posted 40 *fraudulent reviews* for a single restaurant. These 40 users, between them have ever posted only 90 reviews, all within a month, spread across 47 distinct venues, while all sharing one venue in common. We further studied the content of these reviews and compared with the *fraudulent reviews*. We found that the authors of these reviews tended to use more superlative words, such as "definitely", "great" or "really." For example, the word occurrence ratio of the word "love" is almost two times bigger than the corresponding ratio in *fraudulent reviews*, while for the word "like" the opposite trend is observed.

8 LIMITATIONS

We included data from a mix of sources, to help reduce any bias during creation of the ground truth dataset, although any bias still present in the data is inherently difficult to measure. Change Point Analyzer evaluates time-series on a one-month interval. While this may produce some latency between a review being posted and OneReview's classification in a real deployment, this is an artifact of the sparsity of our data sources. This parameter can be adjusted, even at the granularity of a single business, when higher-frequency data is available.

9 CONCLUSION

We presented OneReview, an approach for finding fraudulent activity in crowd-sourced review services, using correlated analysis across multiple independent services. Our method leveraged change point analysis, in tandem with a cross-dataset matching scheme, to find points in the stream of reviews where a significant change in

a business' reputation did not occur evenly across the review websites. We believe our approach of integrating external data sources and features, or "thinking outside the dataset," is an important part of our system's efficacy, and allows our system to perform in a way that no machine or human could before. For future work, we aim to evaluate OneReview when having access to data from a third review website such as Google.

REFERENCES

[1] Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Kruegel, and Giovanni Vigna. 2018. Detecting Deceptive Reviews using Generative Adversarial Networks. *arXiv preprint arXiv:1805.10364* (2018).

[2] Hirotogu Akaike, BN Petrov, and F Csaki. 1973. Information theory and an extension of the maximum likelihood principle. (1973).

[3] Amazon. 2018. *About Amazon Mechanical Turk.* https://www.mturk.com/worker/help

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

[5] Anonymous. 2018. Get Paid to Write Reviews: 27 Sites That Pay You (with Cash & Free Stuff!). http://moneypantry.com/get-paid-to-write-reviews/. (2018).

[6] Anonymous and Symon, Evan V. . 2016. I Get Paid To Write Fake Reviews For Amazon. http://www.cracked.com/personal-experiences-2376-i-get-paid-to-write-fake-reviews-amazon.html. (2016).

[7] Ivan E Auger and Charles E Lawrence. 1989. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology* 51, 1 (1989), 39–54.

[8] Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions.* Association for Computational Linguistics, 69–72.

[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[10] Charles F Bond Jr and Bella M DePaulo. 2006. Accuracy of deception judgments. *Personality and social psychology Review* 10, 3 (2006).

[11] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[12] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An Empirical Comparison of Supervised Learning Algorithms. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06).* ACM, New York, NY, USA, 161–168. https://doi.org/10.1145/1143844.1143865

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[14] Jie Chen and Arjun K Gupta. 2011. *Parametric statistical change point analysis: with applications to genetics, medicine, and finance.* Springer Science & Business Media.

[15] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, MalÃž Castellanos, and Riddhiman Ghosh. 2013. Exploiting Burstiness in Reviews for Review Spammer Detection.. In *ICWSM.* The AAAI Press.

[16] Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2.* Association for Computational Linguistics, 171–175.

[17] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning.* Vol. 1. Springer series in statistics Springer, Berlin. 587–588 pages.

[18] Douglas M Hawkins. 2001. Fitting multiple change-point models to data. *Computational Statistics & Data Analysis* 37, 3 (2001), 323–341.

[19] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08).* ACM, New York, NY, USA, 219–230. https://doi.org/10.1145/1341531.1341560

[20] H Tim Kam. 1995. Random decision forest. In *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition, Montreal, Canada, August.* 14–18.

[21] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. 2012. Optimal detection of changepoints with a linear computational cost. *J. Amer. Statist. Assoc.* 107, 500 (2012), 1590–1598.

[22] Raymond Y. K. Lau, S. Y. Liao, Ron Chi-Wai Kwok, Kaiquan Xu, Yunqing Xia, and Yuefeng Li. 2012. Text Mining and Probabilistic Language Modeling for Online Review Spam Detection. *ACM Trans. Manage. Inf. Syst.* 2, 4, Article 25 (Jan. 2012), 30 pages. https://doi.org/10.1145/2070710.2070716

[23] Hee Andy Lee, Rob Law, and Jamie Murphy. 2011. Helpful reviewers in TripAdvisor, an online travel community. *Journal of Travel & Tourism Marketing* 28, 7 (2011), 675–688.

[24] Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. 2015. Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns.. In *ICWSM.* 634–637.

[25] Jiwei Li, Myle Ott, Claire Cardie, and Eduard H Hovy. 2014. Towards a General Rule for Identifying Deceptive Opinion Spam.. In *ACL (1).* Citeseer, 1566–1576.

[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.* ACM, 939–948.

[27] Yuming Lin, Tao Zhu, Hao Wu, Jingwei Zhang, Xiaoling Wang, and Aoying Zhou. 2014. Towards online anti-opinion spam: Spotting fake reviews from the review sequence. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on.* IEEE, 261–264.

[28] Michael Luca and Georgios Zervas. 2016. Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science* (2016).

[29] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. 2011. Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 30.

[30] Microsoft. 2017. Febipos.A Malware. http://www.microsoft.com/security/portal/threat/encyclopedia/Entry.aspx?Name=Trojan:JS/Febipos.A. (September 2017).

[31] 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.* ACM, 632–640.

[32] Arjun Mukherjee, Bing Liu, and Natalie Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web.* ACM, 191–200.

[33] Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. 2011. Detecting Group Review Spam. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11).* ACM, New York, NY, USA, 93–94. https://doi.org/10.1145/1963192.1963240

[34] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. What yelp fake review filter might be doing?. In *Seventh international AAAI conference on weblogs and social media.*

[35] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. What Yelp Fake Review Filter Might Be Doing? http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6006

[36] Myle Ott, Claire Cardie, and Jeffrey T Hancock. 2013. Negative Deceptive Opinion Spam.. In *HLT-NAACL.* 497–501.

[37] 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.* Association for Computational Linguistics, 309–319.

[38] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. 2010. Running experiments on amazon mechanical turk. (2010).

[39] Mahmudur Rahman, Bogdan Carbunar, Jaime Ballesteros, George Burri, Duen Horng, et al. 2014. Turning the Tide: Curbing Deceptive Yelp Behaviors.. In *SDM.* SIAM, SIAM, 244–252.

[40] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning.*

[41] Reviews That Stick. 2017. Buy Positive Yelp Reviews. http://reviewsthatstick.com/yelp/. (2017).

[42] Gordon J Ross et al. 2013. Parametric and nonparametric sequential change detection in R: The cpm package. *Journal of Statistical Software* 78 (2013).

[43] ANDREW JHON Scott and M Knott. 1974. A cluster analysis method for grouping means in the analysis of variance. *Biometrics* (1974), 507–512.

[44] Review Shepherd. 2017. How To Get Yelp Reviews. https://reviewshepherd.com/articles/get-yelp-reviews/. (2017).

[45] Tim Parker. 2017. Posting Fake Reviews For Your Business May Cost You. https://quickbooks.intuit.com/r/marketing/posting-fake-reviews-for-your-business-may-cost-you/. (2017).

[46] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards Detecting Anomalous User Behavior in Online Social Networks. In *23rd USENIX Security Symposium (USENIX Security 14).* San Diego, CA, 223–238.

[47] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. 2014. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers.. In *USENIX Security Symposium.* 239–254.

[48] Guan Wang, Sihong Xie, Bing Liu, and Philip S Yu. 2012. Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 4 (2012), 61.

[49] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y. Zhao. 2017. Automated Crowdturfing Attacks and Defenses in Online Review Systems. In *ACM Conference on Computer and Communications Security (CCS '17).* Dallas, Texas.

[50] Yelp. 2016. Yelp Dataset Challenge. (September 2016). https://www.yelp.com/dataset_challenge.

[51] Kyung-Hyan Yoo and Ulrike Gretzel. 2009. Comparison of deceptive and truthful travel reviews. *Information and communication technologies in tourism 2009* (2009), 37–47.

[52] Bianca Zadrozny, John Langford, and Naoki Abe. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 435–442.

[53] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of TF* IDF, LSI and multi-words for text classification. *Expert Systems with Applications* 38, 3 (2011), 2758–2765.