Journal Pre-proof

GADM: Manual Fake Review Detection for O2O Commercial Platforms

Na Ruan, Ruoyu Deng, Chunhua Su

PII: S0167-4048(19)30200-7
DOI: https://doi.org/10.1016/j.cose.2019.101657
Reference: COSE 101657

To appear in: *Computers & Security*

Received date: 23 February 2019
Revised date: 21 October 2019
Accepted date: 24 October 2019

Please cite this article as: Na Ruan, Ruoyu Deng, Chunhua Su, GADM: Manual Fake Review Detection for O2O Commercial Platforms, *Computers & Security* (2019), doi: https://doi.org/10.1016/j.cose.2019.101657

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier Ltd.

Journal Pre-proof

GADM: Manual Fake Review Detection for O2O Commercial Platforms

Na Ruan, Ruoyu Deng, Chunhua Su
Dongchuan road No. 800 Shanghai, China 200240 China

***Abstract*—O2O (online to offline) commercial platforms, such as Yelp, play a crucial role in our daily purchases. Seeking fame and profit, some people try to manipulate the O2O market by opinion spamming, i.e., engaging in fraudulent behavior such as writing fake reviews, which affects the online purchasing environment. Manual fake reviews imitate honest reviews in many ways; hence they are more deceptive and harmful than botnet reviews. Several efficient methods have been proposed to detect fake reviews, but manual fake reviewers are evolving rapidly. They pretend to be benign users, control the velocity of review fraud actions, and deceive detection systems. Previous work has focused on the contents of reviews or the information of reviewers. We find that geolocation factors have potential and have been neglected in most studies. Our research indicates that geolocation can well distinguish between fake reviewers and benign users on an O2O platform. We propose a manual fake review detection model, the geolocation-based account detection model (GADM), which combines the AdaBoost model and a long short-term memory (LSTM) neural network to analyze a user's account and geolocation information, achieving 83.3% accuracy and an 86.2% F1-score on a Yelp dataset. We also propose a high-efficiency algorithm to detect review fraud groups.**

***Index Terms*—O2O, Spamming detection, Accounts and address information, LSTM, Machine learning**

I. Introduction

With the explosive growth of electronic commerce and social media, O2O (online to offline) commerce has become a hot topic. O2O refers to the use of online enticement to drive offline sales, and feedback from offline consumption can promote the online dissemination of products [1]. Different from traditional online shopping platforms, the consumption role of O2O is fulfilled offline. O2O closely connects online platforms and traditional offline stores. As on many online shopping platforms, feedback is a crucial part of O2O. Reviews of experienced users in the O2O environment can provide significant reference values for consumers and help them to make decisions. Opinions in reviews are essential to the evaluation and business volume of a target product on current O2O commercial platforms such as Dianping,[1] Booking,[2] and Yelp.[3] Positive reviews can bring profits and fame, while negative ones are harmful. Due to the pursuit of profits, deceptive reviews and manual fake reviewers appeared. These manual fake reviewers mislead, exploit, and manipulate social media discourse with rumors, spam, malware, misinformation, slander, or even just ads [2]. Moreover, fake reviews themselves evolve quickly with the continuous and rapid evolution of social media, posing a significant challenge to the community [3]. Shops often hire people to secretly promote them. People hired to write fake reviews are called manual fake reviewers. These kinds of activities are called opinion spam [4].

Researchers have worked on fake review detection for several years [5]. At the early stage, methods of opinion spam were elementary and easy to identify. Researchers proposed many approaches based on text analysis on the O2O platform [6]. Traditional machine learning methods could also be used to detect suspicious reviews [7]. The spotlight on fake review detection gradually shifted from text content to features and patterns. Some features, such as time [8], ranking patterns [9], topics [10], and volume of events [11], proved useful in fake review detection. These approaches introduced several new ideas. Commercial platform operators built systems to find deceptive and low-quality reviews [12]. Those systems helped purify the disordered review environment, but they also motivated fake reviewers to enrich their review content, and some skilled fake reviewers were able to deceive the system [13]. Fake reviewers have learned to control the rate of review fraud actions and disguise themselves as normal users. Hence classic detection approaches no longer work efficiently, and new detection methods and features are needed.

Account feature-based models have been proved efficient. Geolocation features also work well, especially on O2O platforms. We exploit a creative geolocation-based account detection model (GADM) to detect fake reviews on the O2O platform by adding the geolocation feature to the account feature-based model. GADM consists of two submodels. One analyzes users' account information using AdaBoost, which is a robust machine learning method, and the other analyzes address information to obtain geolocation features by long short-term memory (LSTM), which is a deep learning algorithm. With the help of ensemble learning [14], our GADM model

[1] www.dianping.com

[2] www.booking.com

[3] www.yelp.com

Journal Pre-proof

Journal Pre-proof

can well combine accounts and geolocation features to achieve better performance.

Account information refers to the information in users' account profiles, including how many reviews they post and how many responses they receive. Account information has been proved useful in fake review detection. Geolocation information records the locations of offline activities, and is available in reviews. Our model uses longitude and latitude in geolocation information.

Geolocation has potential in fake review detection, especially on O2O platforms. Unlike other online shopping platforms, consumption is finished offline on O2O platforms, making the geolocation of an offline shop a significant feature for users. Both fake reviewers and benign users have geolocation records in reviews, forming a sequence in order of time. The position order of benign users fits human behavior, while fake reviewers pay it little attention. These differences cause distinctions in the statistics and the frequency distribution of geolocation features. After computing on a partially labeled dataset of reviews and reviewers, we found that fake reviewers and benign users show distinct distributions in their geolocation features.

The use of geolocation features in fake review detection has been discussed before. Zhang et al. [15] used geolocation features in online social networks (OSNs) to detect fake reviewers, and Gong et al. [16] used the LSTM model and check-in information in location-based social networks (LBSNs) for malicious account detection. This work tells us that location information can reflect some features of fraudulent reviewers. They usually use the distance, latitude, or longitude of shops to directly represent geolocation features, which is less efficient. A more powerful and expressive way to use geolocation features is necessary.

Account features and geolocation features are used to describe users from different perspectives, and most prior work only uses one of them, which has limitations in fraud detection. Deng et al. [17] used a hidden Markov model (HMM) to detect fake users. However, they neglected account information in users' daily activities. We combine account and geolocation features to achieve better detection performance by a synthetic model.

Apart from detecting manual fake reviewers, we propose an effective algorithm to identify review fraud groups. Well-organized manual fake reviewers are malicious on O2O platforms. They can impact the market environment by organizing boost review fraud actions. Their detection is also essential in fake review detection. We propose an algorithm to find the relations among manual fake reviewers, and then find review fraud groups. Users whose actions are highly similar to those of known review fraud groups can be regarded as members of those groups.

Our work makes the following key contributions:

1) We add geolocation features to a traditional account feature-based detection model to detect fake reviews on O2O commercial platforms.
2) We build a GADM model that can combine account and geolocation information to detect manual fake reviewers by ensemble learning. Account information is analyzed by AdaBoost. We apply LSTM to describe the distribution distinctions of geolocation features between manual fake reviewers and benign users. GADM receives geolocation feature sequences and produces prediction results.
3) We propose an algorithm to detect review fraud groups, and organized manual fake reviewers are detected.

The remainder of this paper is organized as follows. We introduce preliminary concepts in section II. In section III, we present the design of the GADM model, and in section IV, we explore its application. The dataset, experiment, and evaluation are shown in section V. We conclude our research in section VI.

II. Preliminaries

A. Terminology

We first introduce some definitions in the manual fake review detection scenario.

Definition II.1. Shop: A shop is an officially registered online shop that holds a unique webpage on an O2O platform, containing detailed descriptions of the shop and a large number of reviews.

Definition II.2. User: A user is an officially registered account with a personal webpage on an O2O platform, containing a detailed personal profile and a large number of reviews the user has posted.

Remark II.3. We categorize all users as either **benign users** or **fake reviewers**. **Benign users** post honest reviews, and **fake reviewers** post fake reviews to promote target shops.

Definition II.4. Fake review: Fake reviews are posted by fake reviewers without consumption from the offline shops. Fake reviews contain fabricated texts and imaginary stories crafted to mislead consumers.

Definition II.5. Review fraud groups: Review fraud groups are well-organized manual groups of fake reviewers. Review fraud is the action of manual fake reviewers writing fake reviews.

B. Classification Algorithms in Manual Fake Review Detection

Spamming behaviors are categorized into types such as web spam [5], e-mail spam [18], telecommunication spam [19], and opinion spam [4]. The manual fake review detection problem addresses opinion spam. It can be regarded as a binary classification problem or a ranking problem. The critical problem is the selection of approaches and models. Prior

Journal Pre-proof

research has identified several approaches to the detection of manual fake reviews.

*1) **Texture-based Approaches**:* In 2008, when opinion spamming was first proposed by Jindal et al. [4], researchers focused on the classification and summarization of opinions using natural language processing (NLP) approaches and data mining techniques. From 2011, researchers tried to improve methods of text analysis. Ott et al. [20] built a support vector machine (SVM) classifier using text features including unigrams and bigrams. Shojaee et al. [6] focused on lexical and syntactic features to identify fake reviews, and Chen et al. [21] proposed a semantic analysis approach that calculates the similarity between two texts by finding their common content words. Traditional texture-based approaches are simple, and could not reach high efficiency when manual fake reviewers began to enrich their fake review content.

*2) **Feature-based Approaches**:* From 2014, with the rapid development of machine learning, a number of such algorithms were applied to fake review detection. Li et al. [22] proposed a PU-learning (positive unlabeled learning) model that can improve the performance of Dianping's filtering system by cooperating with Dianping.[4] Kumar et al. [23] proposed an improved SVM model, dual-margin multi-class hypersphere support vector machine (DMMH-SVM), to solve the web spamming problem. Chino et al. [11] trained a log-logistic distribution model based on time intervals and volumes of events generated by users to fit users' behavior, and calculated the dispersion of reviews written by different users to identify those who are isolated from the majority. Li et al. [24] achieved an excellent result with a labeled hidden Markov model (LHMM) combined with time interval features to detect fake reviews in a sizeable Dianping dataset. The feature-based approach is a powerful weapon in fake review detection, but with the evolution of fake reviewers, new powerful features are needed.

*3) **Graph-based Approaches**:* From 2016, some researchers chose graph models to find relations among products, users, and reviews. A detailed graph model can even capture deceptive reviewer clusters. Agrawal et al. [25] demonstrated an unsupervised author-reporter model for fake review detection based on a hyper-induced topic search (HITS) algorithm. Hooi et al. [26] proposed the camouflage-resistant algorithm FRAUDAR to detect fake reviews in the bipartite graph of users and products they review. Chen et al. [9] proposed to identify attackers of collusive promotion groups in an app store by exploiting unusual ranking changes of apps to identify promoted apps. They measured the pairwise similarity of ranking change patterns, formed targeted app clusters, and finally identified the collusive group members. Zheng et al. [27] proposed an ELSIEDET system to detect elite Sybil attacks and Sybil campaigns. Feature-based approaches mainly focus on feature selection, while graph-based approaches attach more importance to patterns and links.

[4]www.dianping.com

C. Long Short-Term Memory

Recurrent Neural Network (RNN) [28] is a feed-forward neural network using variable-length sequential information like sentences or time series. It takes a sequence $(x_1, ..., x_T)$ as input, and updates its hidden states $(h_1, ..., h_T)$. The output is $(o_1, ..., o_T)$, where T is the input time steps. From $t = 1$ to T, the output o_t is computed by the following equations:

$$\begin{aligned} h_t =& tanh(Ux_t + Wh_{t-1} + b) \\ o_t =& Vh_t + c \end{aligned} \tag{1}$$

where U, W and V are the input-to-hidden, hidden-to-hidden and hidden-to-output weight matrices, b and c are the bias vectors, and $tanh(.)$ is a nonlinearity activation function.

LSTM [29] is an improved version based on the traditional recurrent neural network to deal with the problem that basic RNN can not learn long-distance temporal dependencies with gradient-based optimization. LSTM has a meticulous control over the information of input sequence by using three typical gate structure including the input gate, forget gate and output gate. An LSTM unit maintains a memory cell c_t at time t. The output h_t of an LSTM unit is computed by the following equations [29]:

$$\begin{aligned} i_t =& \sigma(x_t W_i + h_{t-1} U_i + c_{t-1} V_i) \\ f_t =& \sigma(x_t W_f + h_{t-1} U_f + c_{t-1} V_f) \\ \tilde{c}_t =& tanh(x_t W_c + h_{t-1} U_c) \\ c_t =& f_t + c_{t-1} + i_t \tilde{c}_t \\ o_t =& \sigma(x_t W_o + h_{t-1} U_o + c_t V_o) \\ h_t =& o_t tanh(c_t) \end{aligned} \tag{2}$$

where σ is a logistic sigmoid function. The input gate i_t determines how much new memory is added to the memory cell. The forget gate f_t determines the degree the memory cell is about to forget. The memory c_t is combined with part of the existing memory and part of new memory $\tilde{c}_t$. The output gate o_t decides the output.

There exist some prior works that apply RNN to fake review detection task. Ma et al. [30] used an efficient algorithm to split social media reviews into groups by time and compared the performance between basic RNN, LSTM, and GRU. Ren et al. [31] applied a gated recurrent neural network into sentence representations to detect deceptive opinion spam. Jin et at. [32] proposed an RNN model with an attention mechanism to fuse multimodal features for effective rumor detection.

D. AdaBoost Classifier

AdaBoost [33] is a typical ensemble learning algorithm that can improve a group of base learners to a strong learner. The mechanism of AdaBoost is: training the first base learner by the default training set first and adjusting the distribution

Journal Pre-proof

Journal Pre-proof

of training samples according to the performance of the base learner, and then those misclassified samples will be paid more attention to. The next base learner will be trained by the adjusted training set. Iteratively, the algorithm stops when the number of base learners exceeds the default target value. Finally, those base learners are weighted combined. AdaBoost has those advantages compared with traditional binary classifiers[34][35]:

1) The AdaBoost algorithm performs high precision
2) Easy to use and transplant. A plug-and-play algorithm
3) No need for feature filtrating
4) Base learners can be customized under the framework of the AdaBoost algorithm
5) Users need not worry about overfitting problem

Due to the advantages above, we build an AdaBoost based model to detect fake reviewers by account features. AdaBoost has a better performance in fake reviews detection compared with other machine learning methods.

III. GADM: Manual Fake Review Detection Model

A. Structure Overview

In this section, we introduce our manual fake review detection model, GADM, which is composed of two submodels, the **account detection model (Account-DM)** and **geolocation detection model (Geolocation-DM)**. The structure of our fake review detection process is shown in Figure 1. It has three phases.

Phase I: Users' account and geolocation information is collected and processed for detection tasks. Statistical information, such as review numbers, is extracted, and all reviews and their geolocations are collected. Features must be processed for use in the next phase. Account features are formed as a vector, and geolocations as a sequence.

Phase II: Detection models are constructed to process account and geolocation features. Account features are input to Account-DM, which analyzes the account features using AdaBoost, and provides a prediction score of the user's identity. Similarly, geolocation features are input to Geolocation-DM, which analyzes the geolocation sequences by LSTM and produces a prediction score of the user's identity.

Phase III: The results of Account-DM and Geolocation-DM are synthesized by ensemble learning to make a final judgment of users' identities. The prediction scores from Account-DM and Geolocation-DM are input to the final linear classifier, SVM, whose result is the final assessment of users' identities.

B. Notions and Definitions

Table I lists the notions used in this paper.

TABLE I
Notions and Definitions

Notion	Interpretation
lng_i	Longitude of point i
lat_i	Latitude of point i
$C(lng_c, lat_c)$	Center point
R_i	Radius feature of review i
$r_{account}$	Output result of Account-DM
r_{geo}	Output result of Geolocation-DM
r_{final}	Output result of combination model
$\mathbf{D}(x_i, y_i)$	Training dataset, where x_i is input data and y_i is the label
$\mathcal{D}$	Weight distribution in dataset
h_t	tth base learner in AdaBoost
ϵ_t	Error of base learner h_t
ω_t	Weight of base learner h_t
$\mathcal{Z}$	Regularization factor
$\mathbf{H}(x)$	Output of AdaBoost
Seq_R	Review sequence
N_{review}	Number of reviews

C. Features

GADM detects users by account and geolocation features, which describe O2O users from different aspects. Both are easily available from an O2O platform. As shown in Figure 2, account and geolocation information is publicly accessible from user account webpages and reviews, respectively. The convenience of collecting features makes our detection system practicable and transplantable to many O2O platforms. Benign and fraudulent users have different behaviors based on the two features. The account feature describes a user's profile. It records users' historical statistics and their influences, such as numbers of reviews and friends. The geolocation feature records historical activity, including spatial and time sequence information.

*1) **Account Feature**:* Each user holds a registered account on an O2O platform. Accounts record users' historical activities and many useful statistics, like the number of reviews, useful reviews, and fans. Nine kinds of common and available statistics were chosen as account features, and their average values and standard deviations were analyzed for benign users and manual fake reviewers. The analysis results are listed in the following tables.

TABLE II
Average value of statistics of benign users and fake reviewers

	friends	reviews	firsts	useful	cool
benign users	31.57	55.17	9.73	108.161	68.59
fake reviewers	2.13	6.15	0.18	2.37	0.98
	funny	like	tips	fans	
benign users	55.21	71.92	13.84	2.717	
fake reviewers	0.8047	0.3631	0.1254	0.055	

Tables II and III demonstrate that those statistics distinguish much between fake reviewers and benign users. These features are used in the Account-DM model. Benign users interact with others much more than do fake reviewers. Account-DM can recognize and exploit the distinction in the fake review detection task.

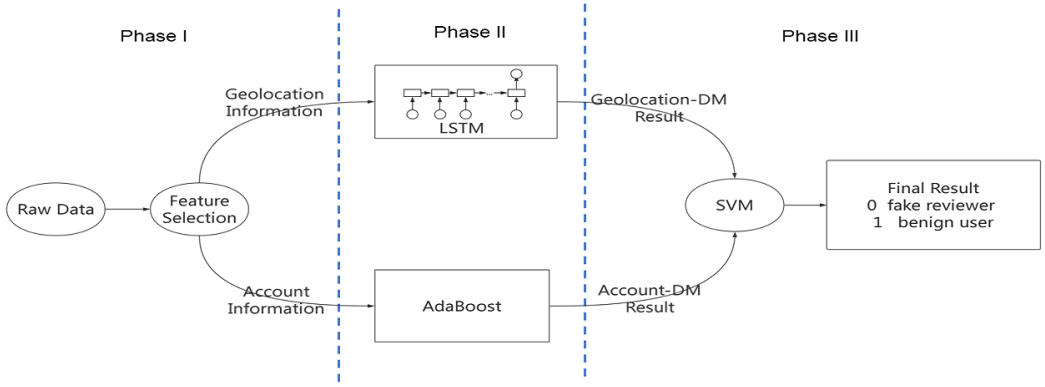
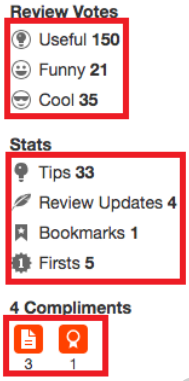
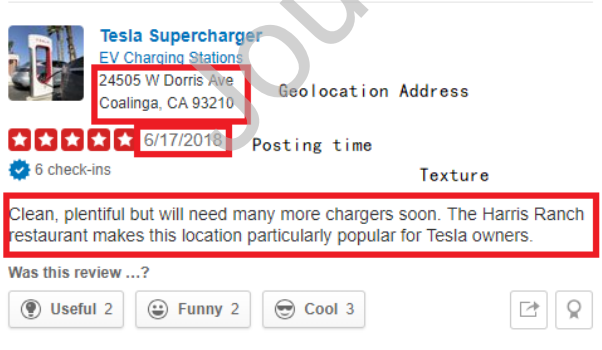


Fig. 1. Fake review detection process. Its three phases consist of extracting features, analyzing features, and synthesizing results.



(a) Account statistics of a Yelp account shown on user webpage.



(b) Review information, including geolocation address, posting time, and texture

Fig. 2. User account webpages on Yelp

TABLE III
STANDARD DEVIATION OF STATISTICS OF BENIGN USERS AND FAKE REVIEWERS

	friends	reviews	firsts	useful	cool
benign users	143.78	134.66	42.87	623.31	506.12
fake reviewers	25.63	26.69	5.18	31.24	15.84
	funny	like	tips	fans	
benign users	339.44	983.78	59.26	19.53	
fake reviewers	25.60	9.17	16.35	0.91	

*2) **Geolocation Feature**:* Consumers must consider geolocation information of shops, since offline consumption is an essential part of O2O platforms. Most O2O platforms collect users' geolocation information to recommend shops close to them, and users tend to choose shops near their homes. If a normal user purchases several times in a day, the locations of these shops should have spatial continuity because the movement track of a human also holds spatial continuity. However, fake reviewers consider employers' benefits much more than the locations of target shops; hence, their trajectories in a day can be abnormal. Figure 3 shows a fake reviewer's movement track in one day by collecting the geolocation information of the shops in the user's review list. If a user writes many reviews in a day, we use the offline consumption time instead of review time. Figure 3 illustrates the locations of shops and the user's movement track in a day. The green lines are the user's trajectories. These lines show that the user's movement track covered two cities and exceeded 410.2 km. Furthermore, even in only one city, the user's movement was disordered and covered a great area. Fake reviewers just receive assignments and never consider the practicability of achieving it in a short period.

IPs or MAC addresses are useful for analyzing geolocation information. However, these sensitive features are not easily obtained from many platforms. We propose a more straightfor-

Journal Pre-proof

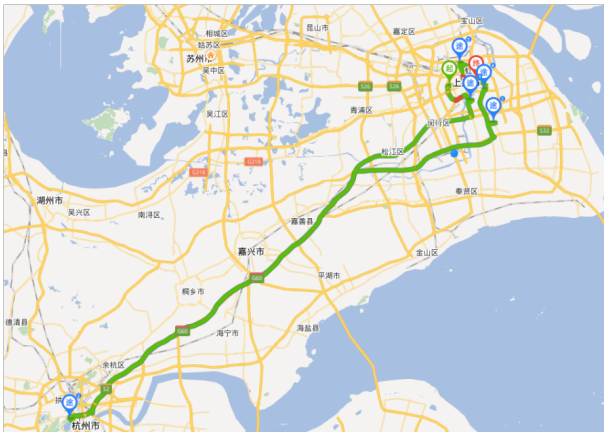


Fig. 3. A fake reviewer's movement track in one day.

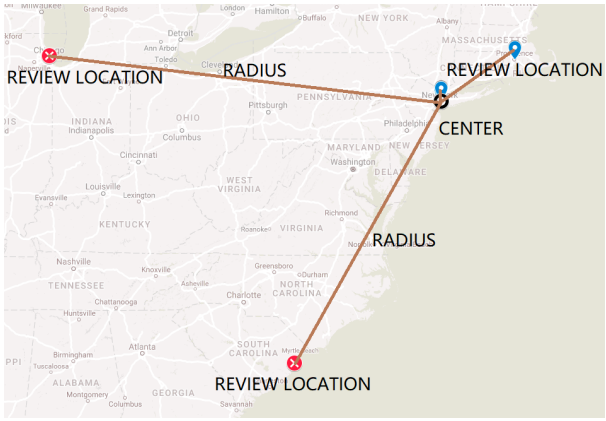


Fig. 4. Definition of radius feature on Google map.

ward location-related feature, radius, to measure the disorder degree of users' movement tracks. First, we introduce the definitions of review location, center point, and radius.

Definition III.1. Review location: Review location is the geolocation point of the shops that appear in users' reviews. It notes the location where a user purchased offline.

Definition III.2. Center point: Center point is the geometric center of the shops in a user's reviews. To determine a user's center requires two steps:

(1) Find the city that the user lives in.
(2) Find the geometric center of shops for which the user has posted reviews in the city where he or she lives.

Definition III.3. Radius: Radius is the distance between each review location and the center point.

Figure 4 shows an example of the definition of the radius feature. Most of the review locations are in New York, so the center point is also in New York. The lines connecting the center point and each review location represent the interval distances between them, which are the radii for these review locations.

The center point is the geolocation center of the user's most active area. If users are active in multiple cities that are distant from each other, then many errors will occur when measuring the center point if all review locations are considered. In Geolocation-DM, we only consider the city for which a user writes most reviews, and we calculate the center point from all review locations in that city. Expression (3) shows the calculation of center point $C(lng_c, lat_c)$, where lng_c and lat_c are the longitude and latitude, respectively, of the center point, n is the number of reviews in this city, and lng_i and lat_i are the longitude and latitude, respectively, of the ith review location, where $i \in 1, 2 \cdots n$.

$$\begin{aligned} lng_c &= \frac{1}{n}\sum_{i=1}^{n} lng_i \\ lat_c &= \frac{1}{n}\sum_{i=1}^{n} lat_i \end{aligned} \tag{3}$$

The radius feature is the shortest distance between two points on the spherical surface because the earth is approximately an ellipsoid. As shown in expression (4), the radius feature R_i is the spherical distance between the location of each review i and the center point C, where R is the approximate radius of the earth.

$$\begin{aligned} R_i =& R \times \arccos(\cos(lat_c)\cos(lat_i)\cos(lng_i - lng_c) \\ &+ \sin(lat_i)\sin(lat_c)) \end{aligned} \tag{4}$$

A user's geolocation information is converted to radius features by the calculations introduced above. Figure 5 illustrates that manual fake reviewers and benign users show distinct distributions regarding geolocation features. Both their slopes and peak positions are much different.

Radius features reflect the distance between each review location and the center point. Each review holds a radius feature, and a sequence of reviews can form a new sequence of radius features. The radius sequence can reflect both distance and spatial continuity features, which is useful when modeling features.

D. Modeling features

1) ***Account Detection Model****:* Account-DM exploits users' account information to identify suspect users. As shown in Table IV, there are nine dimensions in account features.

Account-DM is a regression model. The model input holds nine integers, and outputs a float number $r_{account}$ ranging from 0 to 1. A user with a higher $r_{account}$ is more likely to be a benign user.

Account-DM exploits the AdaBoost algorithm to perform regression tasks. Algorithm 1 demonstrates the application of

Journal Pre-proof

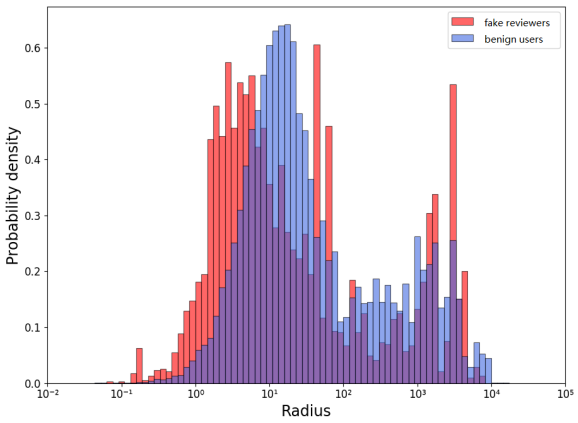


Fig. 5. Frequency distributions of radius.

TABLE IV
DIMENSIONS OF ACCOUNT FEATURES

Type	friends integer	reviews integer	firsts integer	useful integer	cool integer
Type	funny integer	like integer	tips integer	fans integer	

AdaBoost in Account-DM. In Algorithm 1, training dataset $\mathbf{D}$ contains n data samples. The pair of data samples (x_i, y_i) contains a vector x_i with nine dimensions, which represent the nine account features, and a label value $y_i \in \{0, 1\}$, where 0 represents fake reviewers and 1 represents benign users. $\mathcal{D}_t$ is the weight of data samples in $\mathbf{D}$ in the tth training iteration. h_t represents the tth base learner trained from training samples. The base learner is responsible for practically classifying data samples in Account-DM, and the weighted combination of all the base learners gives a final classification result $\mathbf{H}(\mathbf{x})$. ϵ_t is the error of h_t. ω_t is the weight of h_t. $\mathcal{Z}_t$ is the regularization factor that guarantees $\mathcal{D}_{t+1}$ is a valid distribution.

First, h_1 is calculated based on the original dataset $\mathbf{D}$ and the initial data sample weight distribution $\mathcal{D}$. All data samples initially have weight $\frac{1}{n}$. In the tth iteration, Account-DM trains h_t based on the latest data sample weight distribution $\mathcal{D}$ and dataset $\mathbf{D}$. If ϵ_t is larger than 0.5, then h_t is even worse than a random classifier and must be abandoned. ω_t is also calculated for the weighted combination. $\mathcal{D}_t$ varies according to the last training results. It gives misclassified data samples larger weights so that they will be emphasized more in the next iteration. At the end of each iteration, Account-DM updates the current regression result, as shown in expression (5).

$$H_t = H_{t-1} + \omega_t h_t(x) \tag{5}$$

After T iterations, Account-DM obtains a final classifier $\mathbf{H}(x)$, which is the weighted sum of all the base learners. In the experiment, the regression tree algorithm [36] serves as the base learner.

Algorithm 1: Account Detection Model based on AdaBoost

Input: Training dataset $\mathbf{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$;
Base Learner algorithm $\mathcal{L}$;
Iteration $\mathbf{T}$;
Weight distribution of Data sample $\mathcal{D}_1(x) = \frac{1}{n}$
;
for $t = 1, 2, \cdots, T$ **do**
  $h_t = \mathcal{L}(\mathbf{D}, \mathcal{D}_t)$;
  $\epsilon_t = \mathbf{P}_{\mathbf{x}\sim\mathcal{D}_t}(|h_t(\mathbf{x}) - y| \geq 0.5)$;
  if $\epsilon_t > 0.5$ **then break**;
  $\omega_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$;
  $\mathcal{D}_{t+1} = \frac{\mathcal{D}_t(\mathbf{x})}{\mathcal{Z}_t} \times \begin{cases} \exp(-\omega_t) & h_t(\mathbf{x}) = y \\ \exp(\omega_t) & h_t(\mathbf{x}) \neq y \end{cases}$;
end
Output: Regression result $\mathbf{H}(\mathbf{x}) = \sum_{t=1}^{\mathbf{T}} \omega_t h_t(\mathbf{x})$

*2) **Geolocation Detection Model**:* All the reviews that a user has posted on the O2O platforms are collected for Geolocation-DM. Each review is written for a particular shop, whose geolocation is available to the public. The review sequence is available by listing a user's reviews and shops in the order of posting time. Next, the geolocation addresses are converted to longitudes and latitudes, and the center point C and interval distance R between each shop and the center point C is calculated, as described in section III-C2. These distances are also listed as a sequence in the order of posting time for each user. We refer to a distance sequence as a review radius sequence Seq_R.

Geolocation-DM performs the regression task for Seq_R. The model input is a float vector Seq_R, and the model output is a float number r_{geo} ranging from 0 to 1. Like Account-DM, samples with a higher r_{geo} are more likely benign users. Since Seq_R is strongly related to time sequences, a neural network approach, the LSTM model, is applied in Geolocation-DM. A single-layer LSTM can perform well enough, since our dataset is not too large, and the input dimension is 1. A drop-out mechanism is added to Geolocation-DM to avoid overfitting. Some important parameters of Geolocation-DM are listed in Table V.

Activation function ReLU:

$$\phi(c_t) = \max(0, c_t) \tag{6}$$

Loss function mean square logarithmic error:

$$L(x_i, y_i) = \frac{1}{N}\sum_{i=1}^{N}(\log(x_i) - \log(y_i))^2 \tag{7}$$

Journal Pre-proof

Journal Pre-proof

TABLE V
SOME IMPORTANT PARAMETERS OF GEOLOCATION-DM

Layers	Input sequence length	Neurons
1	200	10
Activation Function	Loss Function	Batch size
ReLU	mean square logarithmic error	64
Drop-out rate	Learning rate	
0.3	0.001	

As the elements of Seq_R are entered in Geolocation-DM in turn, its weight values are continually updated. After all elements are input, a final result r_{geo} is available at the output gate. Since the radius sequences are processed by batches in Geolocation-DM, the lengths of input sequences Seq_R need to be in accordance. A maximum input length $L_{max} = 200$ is set in Geolocation-DM. If the lengths of data sequences are less than 200, Geolocation-DM will fill the data samples with zeros at the ends of sequences. If the lengths of data sequences are greater than 200, then Geolocation-DM will delete entries after 200 in the sequences. A masking layer in Geolocation-DM is added to filter the data with the value zero to eliminate errors caused by the filling operations.

*3) **Model Combination**:* The account information of a suspect user will be input to Account-DM and a result $r_{account}$ will be output, the user's review sequences Seq_R will be input to Geolocation-DM, and another result r_{geo} will be output. To combine the two kinds of detection models, we refer to stacking thoughts in ensemble learning. A new classifier, taking $r_{account}$, r_{geo}, and the number of reviews of the user No_{review} as input data, is trained, and it provides a final judgment to identify the user. This is also why we choose regression models rather than discrimination models as submodels.

Our detection system exploits SVM as the final linear classifier. The model's inputs are two float detection results, $r_{account}$ and r_{geo}, and the user's number of reviews, N_{review}. The model's output is an integer, $r_{final} \in \{0, 1\}$, where 0 and 1 respectively represent fake and benign users. The sigmoid function serves as the kernel function of SVM. Expression (8) shows the sigmoid function:

$$\mathbf{K}(x, x_i) = \tanh\left(\|(x^T x_i) + \delta)\right) \tag{8}$$

As introduced above, we have achieved the synthesis of account and geolocation information by combining Account-DM and Geolocation-DM. The evaluation and analysis will be demonstrated in section V.

IV. APPLICATION OF FAKE REVIEW DETECTION MODEL

We now discuss the application of GADM. We focus on detecting review fraud groups, which are common on O2O platforms. Fake reviewers are always organized, and they rarely write fake reviews alone. A possible situation is that a store owner needs to improve his store's reputation on the O2O platform in a short time, for which he seeks review fraud groups. It is unlikely that a shop owner hires fake reviewers for a long period, due to their cost.

A. *Review Fraud Group Detection Algorithm*

Fake reviews from the same group are similar and strongly related. Fake reviewers tend to write fake reviews for target shops in a short period. We exploit this feature and propose an algorithm to find review fraud groups.

Algorithm 2: Review fraud group detection

Input: Set of fake reviewers $\mathbf{W} = \{W_1, W_2, ..., W_n\}$, set of shops $\mathbf{S} = \{S_1, S_2, ..., W_m\}$,;
Graph $\mathbf{G}(V = W, E = \emptyset)$ // Build a graph G only contains nodes, Time window ω ;
Output: Graph G // Edges of Graph G represent performing review fraud action collaboratively

```
for i = 1, ..., m do
    remove reviews in shop S_i written by benign users;
    sort reviews in shop S_i by time;
    for review j in shop S_i do
        for review k from review j + 1 in shop S_i do
            if k.time - j.time < ω then
                if G.edge(j.reviewer, k.reviewer) exists then
                    G.edge(j.reviewer, k.reviewer).weight += 1
                else
                    add G.edge(j.reviewer, k.reviewer);
                    set G.edge(j.reviewer, k.reviewer) = 0
                end
            else
                continue
            end
        end
    end
end
```

Algorithm 2 shows the review fraud group detection. A fake reviewer graph G is created, whose each node represents a fake reviewer, and G initially has no edges. At first, reviews in a shop can be calculated by time. Then the algorithm travels all reviews by shop. If two fake reviewers write reviews for the same shop in a short time window, an edge connecting them will be added to G. If two fake reviewers write reviews for the same shop in a short time window twice or more, then the weight of the edge connecting them will increase. After traveling all the fake reviews, the fake reviewer graph G is

Journal Pre-proof

completed. We can detect review fraud groups according to the edges and weights in G.

Next, we analyze the time complexity of our algorithm. Another approach needs to travel all pairs of fake reviewers, analyze their reviews, and make a judgment. Suppose there are n fake reviewers, everyone holds p reviews on average, and using a sort algorithm to speed up, the time complexity of the traditional approach is $O(n^2 plog(p))$. However, our detection algorithm decreases the calculation cost by traveling shops rather than fake reviewers. In our detection algorithm, we only need to travel shops once and find the reviews whose time intervals are less than the threshold. Sorting reviews by time also increases efficiency. Suppose there are m shops, each holding q reviews on average, and the time complexity of our detection algorithm is $O(mqlog(q))$. Since the total number of fake reviews is constant, an equivalence expression is shown in (9):

$$np = mq \tag{9}$$

The time complexity improvement rate of our detection algorithm is

$$K = \frac{n^2 plog(p)}{mqlog(q)} = \frac{nlog(p)}{log(q)}. \tag{10}$$

In summary, our detection algorithm sharply decreases the time complexity, making the detection of review fraud groups on scalable datasets more efficient. The experimental results are shown in section V.

B. *Potential Fake Reviewer Detection*

Review fraud group detection has a significant application. Once a review fraud group is revealed, the potential fake reviewers belonging to this group are likely to be found. A potential fake reviewer is one who is misjudged by the detection model as a benign user. We propose a potential fake reviewer detection algorithm based on review fraud group detection.

Algorithm 3 addresses potential fake reviewer detection. A graph G' which contains all users is built as vertices. Edges in graph G' indicate how often two users review the same store in a short time. Once graph G' is built, potential fake reviewers can be easily found. The key idea is that a user who often performs review fraud action collaboratively with fake reviewers from the same review fraud group can be regarded as a potential fake reviewer belonging to this group.

For a user u_i, a list of fake reviewers, $list_i$, is available from graph G', which contains fake reviewers who have edges with u_i in graph G'. The algorithm sums the weights of fake reviewers in the same group, and obtains a list of groups, $group_i$, which contains the total collaborative fraud actions times between user u_i and fraudulent reviewers in each group. W_i is the sum of weights of all edges between u_i and fake reviewers in $list_i$. The maximum weight in list $group_i$ is max_i. If equation (11) is satisfied, then user i can be considered a potential fake reviewer.

Algorithm 3: Potential fake reviewer detection

Input: Set of users $\mathbf{U} = \{U_1, U_2, ..., U_n\}$, set of shops $\mathbf{S} = \{S_1, S_2, ..., W_m\}$;
Graph $\mathbf{G}'(V = W, E = \emptyset)$ // Build a graph G' only containing nodes, Time window ω;
Output: Graph $\mathbf{G}'$ // Edges of graph G' represent doing review fraud action collaboratively

```
for i = 1, ..., m do
    sort reviews in shop S_i by time;
    for review j in shop S_i do
        for review k from review j + 1 in shop S_i do
            if both reviewj and reviewkare written by benign user then
                continue
            else
                if k.time - j.time < ω then
                    if G'.edge(j.reviewer, k.reviewer) exists then
                        G'.edge(j.reviewer, k.reviewer).weight += 1
                    else
                        add G'.edge(j.reviewer, k.reviewer);
                        set G'.edge(j.reviewer, k.reviewer) = 0
                    end
                else
                    continue
                end
            end
        end
    end
end
```

$$max_i > max(W_i * M, R) \tag{11}$$

W and R are thresholds. M is a float number between 0 and 1, and R is a positive integer. $max(W_i * M, R)$ guarantees that the suspicious user's most collaborative reviewers from the same group by $W_i * M$ and it avoids coincidence by R. If a user's collaborative reviewers are evenly distributed in many groups, or the user only has one or two collaborative fraud actions in total, then max_i will be less than $max(W_i * M, R)$, and the user will not be regarded as a fake reviewer.

V. Experiments

In this section, we describe the dataset, introduce the experiment, and evaluate the performance of GADM and its applications.

Journal Pre-proof

A. Dataset Description

Dataset. We choose a real-world dataset, which is the Yelp dataset used by Santosh et al. [8]. It is a partly labeled dataset, containing user profiles, review information, and shop information. For the labeled part, reviews are labeled as fake or benign reviews by Yelp's filtering system. The dataset information is shown in Table VI. It contains $3,142$ labeled users out of 16,941 total users, and 107,264 labeled reviews out of 760,212 total reviews. There are 20,267 fake reviews among 107,624 labeled reviews. Some users only posted benign reviews, some users only posted fake reviews, and some users posted reviews of both classes. A clear boundary is necessary to cluster two kinds of users. We refer to Nilizadeh's work [10], calculate the filter rate (the percentage of filtered reviews out of all reviews) of each user, and set a boundary filter rate to cluster two kinds of users. The dataset has the characteristic that the filter rate of each user is distributed either in the range of 0-20% or 90%-100%. To separate fake reviewers and benign users, a classification standard is set. Users whose filter rates are higher than 90% are regarded as fake reviewers, and those with filter rates lower than 20% are regarded as benign users. Under this standard, there are 1,299 fake reviewers out of 3,124 labeled users. Users holding few reviews must be excluded from the dataset to decrease unexpected errors. There are 1,796 labeled users and a total of 11,917 users left if the review number threshold is set as 5.

TABLE VI
DATASET INFORMATION

	labeled	total
reviews	107624	760212
users	3142	16941
fake reviews	20267	N/A
fake reviewers	1299	N/A
users after filtering	1796	11917

Ground-truth dataset. We rely on the Yelp filtering system for labeling. This system can filter some typical inferior quality and fake reviews. These officially labeled reviews are qualified as the ground-truth dataset. Some prior work used manually labeled data for the fake review detection task. However, manual work is tedious and subjective. Manual labels have difficulty producing excellent results.

B. Model Evaluation

In this section, we present the experimental implementation and evaluation of GADM. The geolocation features are calculated by latitudes and longitudes of every review shop. These are translated from *Arcgis* map addresses by a Python package named *geocoder*. Parameters of GADM are trained from the training dataset, upon which the evaluation is based. The training and testing datasets are disjointed parts in labeled data. The ratio of manual fake reviewers and benign users in labeled data is unbalanced, and is about $1:3$. Manual fake reviewers on real O2O platforms form a minority. Classifiers are required to hold the resistance to the interference from the unbalanced dataset. Many traditional classification algorithms perform poorly in such a situation, while GADM can tolerate the impact of large volumes of misleading data and precisely recognize the minority manual fake reviewers.

GADM must be compared to other approaches to show its performance advantages. Some traditional supervised classifiers are selected as a comparison group, since GADM is a supervised method. The comparison groups contain four typical classification algorithms: Gaussian naive Bayes (Gaussian NB), k-nearest neighbors (KNN), another LSTM network with review time interval as features and SpamTracer [17] which analyzing geolocation features by Hidden Markov Model (HMM). The first two comparison models receive several account characteristics (e.g., number of friends and reviews) from the dataset and output the prediction of fake reviewers or benign users. Our experiment uses 10-fold cross validation (CV) to guarantee the evaluation result. All models and their results are presented below.

(1) **Gaussian NB**: A Gaussian naive Bayes classifier receives account information.
(2) **KNN**: A k-nearest neighbors classifier receives account information.
(3) **Time**: An LSTM model receives a user's review time intervals.
(4) **SpamTracer**: An HMM model receives a user's geolocation information.
(5) **Account-DM (account detection model)**: A submodel of GADM, which receives account information.
(6) **Geolocation-DM (geolocation detection model)**: A submodel of GADM, which receives geolocation information.
(7) **GADM**: The final model, which combines Account-DM and Geolocation-DM with ensemble learning.

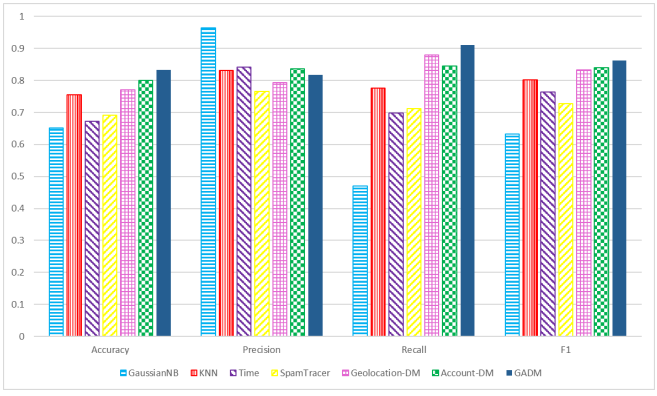


Fig. 6. Precision, Recall, Accuracy, and F1-score of Models.

The evaluation of models is based on four acknowledged standard performance measures: accuracy, precision, recall,

Journal Pre-proof

and F1-score. Figure 6 illustrates the four performance measures of all five models and shows that GADM performs most stably in all four measures. Gaussian NB holds the highest precision but performs poorest in the other three measures. KNN, Time, and SpamTracer perform with almost the same precision as Geolocation-DM and Account-DM, but they still fluctuate much, and they fall far behind the two submodels in other measures. This is reasonable because a poorly performing model will fluctuate on these measurements. For example, if a model is not sensitive in classifying a user as a fake user, the model will have higher precision, but the recall will be lower. Our two submodels perform better than the first three comparison models. The final model, GADM, has higher recall, accuracy, and F1-score, and the same level of precision as the two submodels, which means our ensemble model improves performance in detecting manual fake reviewers. In summary, GADM is the most stable model in our experiment.

We find that Gauusian NB has high precision but low accuracy, indicating that it has high accuracy in identifying fraudulent users, and for benign users, Gaussian NB performs poorly. To verify this, we estimate models' detection abilities for benign and fraudulent users. The results are shown in Table VII. The result verifies it, and shows that Gauusian NB overfits for fraudulent users and our proposed model can combine the advantages of the two sub-models.

TABLE VII
ACCURACY FOR FRAUDULENT AND BENIGN USERS OF MODELS

	Accuracy for Fraud users	Accuracy for Benign users
Gaussian NB	**0.985**	0.440
KNN	0.727	0.789
Geolocation-DM	0.745	0.763
Account-DM	0.719	0.861
GADM	0.859	**0.858**

In conclusion, GADM has excellent stability and performs above average in all four measures under an unbalanced dataset. With interference from an unbalanced dataset environment, these classical approaches cannot find a compromise among those measures. GADM improves two sub-models' performance by ensemble learning.

C. Account-DM vs. Geolocation-DM

*1) **Diversity**:* Apart from accuracy evaluation, the evaluation of diversity among sub-models is necessary. There are two main diversity measures: the disagreement measure and correlation coefficient. A prediction contingency table of models h_i and h_j is shown in Table VIII for a given dataset $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}$ and $y_i = \{+1, -1\}$ in a binary classification task. The measure value a represents the number of samples that both h_i and h_j predict correctly, b represents the number of samples that h_i predicts correctly and h_j predicts incorrectly, and so are c and d.

TABLE VIII
PREDICTION CONTINGENCY TABLE OF MODELS ACCOUNT-DM AND GEOLOCATION-DM

	$r_{account} = +1$	$r_{account} = -1$
$r_{geo} = +1$	a	c
$r_{geo} = -1$	b	d

Based on the prediction contingency table, we calculate two main diversity measures for pairs of submodels Account-DM and Geolocation-DM:

1) **Disagreement measure**

$$dis = \frac{b+c}{a+b+c+d}$$

2) **Correlation coefficient**

$$\rho = \frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$$

The value of dis belongs to [0,1]. A larger dis represents a better diversity measure between Account-DM and Geolocation-DM. The value of ρ belongs to [-1,1]. If Account-DM is independent of Geolocation-DM, then ρ will be 0. A positive value of ρ shows a positive correlation between Account-DM and Geolocation-DM, and vice versa.

The disagreement measure between the two submodels is $dis = 0.26$, and the correlation coefficient is $\rho = 0.46$. Geolocation-DM and Account-DM have a high disagreement, which means that the two models perform a remarkable diversity. The high disagreement measure and correlation coefficient make it possible to combine the two submodels by ensemble learning.

*2) **Importance**:* We use random forest [37], a machine learning method, to test the importance of Geolocation-DM and Account-DM in the ensemble model GADM. Our ensemble model is fed with three features: two detection results of the submodels, $r_{account}$ and r_{geo}, and the user's number of reviews, N_{review}; their importance ratios are listed in Table IX. The Account-DM and Geolocation-DM results are much more important than the number of reviews.

TABLE IX
IMPORTANCE OF FEATURES IN THE ENSEMBLE MODEL.

Feature	Importance
Account-DM result	45.3%
Geolocation-DM result	36.7%
Review number	18.0%

D. Detect Review Fraud Groups

We proposed an approach to detect review fraud groups in section IV. We will evaluate its performance by applying the group detection algorithm on the completely labeled dataset. Figure 7 shows part of the visualized graph G output by the group detection model. The thickness and color of edes represent the frequency with which pairs of fake reviewers work together. Thick lines and dark colors represent that pairs

of fake reviewers often conduct group work, and thin lines and light colors represent that pairs of fake reviewers sometimes conduct group work. The time window was set as three days, and 4,818 organized fake reviewers were found.

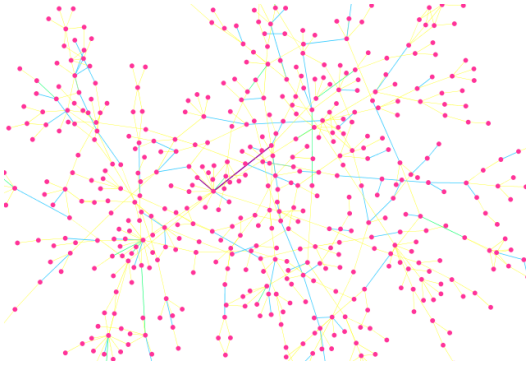


Fig. 7. Visualized graph G output by our group detection model.

As we can see from Figure 7, almost all the fake reviewers are connected, since it is common to find two fake reviewers from different review fraud groups working together by coincidence, even if just once. To eliminate the connections caused by coincidence, a threshold on edge weight is necessary, i.e., only edges whose weights are more than the threshold are considered meaningful. Figure 8 shows the individual review fraud groups our model found when the threshold of edge weights was 2. Compared with Figure 7, the boundaries of groups are more clear. It forms an unconnected graph. Each unconnected component can be regarded as a fraud review group.

Fig. 8. Visualized graph G after setting the threshold of edge weights.

VI. Conclusion

In this paper, we proposed a GADM detection model to detect manual fake reviewers on an O2O commercial platform. GADM improves traditional account-based fake review detection models by adding geolocation features. GADM detects manual fake reviewers by exploiting the unique distinctions of account and location features between fake and benign users. Our evaluation is based on a large Yelp dataset, and the results demonstrate that our approach can perform the fake review detection task with excellent accuracy and stability. We also proposed efficient algorithms to detect review fraud groups on an O2O platform, which help us to detect more suspicious fake reviewers.

References

[1] C. W. Phang, C. H. Tan, J. Sutanto, F. Magagna, and X. Lu, "Leveraging o2o commerce for product promotion: An empirical investigation in mainland china," in *IEEE Transactions on Engineering Management*, vol. 61, no. 4, 2014, pp. 623–632.

[2] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," in *Commun. ACM*, vol. 59, no. 7, 2016, pp. 96–104.

[3] M. Chakraborty, S. Pal, R. Pramanik, and C. R. Chowdary, "Recent developments in social spam detection and combating techniques: A survey," in *Information Processing & Management*, vol. 52, no. 6, 2016, pp. 1053–1073.

[4] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, New York, USA, 2008, pp. 219–230.

[5] N. Spirin and J. Han, "Survey on web spam detection: Principles and algorithms," in *SIGKDD Explor. Newsl.*, vol. 13, no. 2, 2012, pp. 50–64.

[6] S. Shojaee, M. A. A. Murad, A. B. Azman, N. M. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *13th International Conference on Intellient Systems Design and Applications*, Malaysia, 2013, pp. 53–58.

[7] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in *Proceedings of the 21st International Conference on World Wide Web*, Lyon, France, 2012, pp. 191–200.

[8] S. KC and A. Mukherjee, "On the temporal dynamics of opinion spamming: Case studies on yelp," in *Proceedings of the 25th International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland, 2016, pp. 369–379.

[9] H. Chen, D. He, S. Zhu, and J. Yang, "Toward detecting collusive ranking manipulation attackers in mobile app markets," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, UAE, 2017, pp. 58–70.

[10] S. Nilizadeh, F. Labrèche, A. Sedighian, A. Zand, J. Fernandez, C. Kruegel, G. Stringhini, and G. Vigna, "Poised: Spotting twitter spam off the beaten paths," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, USA, 2017, pp. 1159–1174.

[11] D. Y. T. Chino, A. F. Costa, A. J. M. Traina, and C. Faloutsos, "Voltime: Unsupervised anomaly detection on users' online activity volume," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, Houston, USA, 2017, pp. 108–116.

[12] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Proceedings of the 7th International Conference on Weblogs and Social Media*, Boston, USA, 2013, pp. 409–418.

[13] Y. Yao, B. Viswanath, J. Cryan, H. Zheng, and B. Y. Zhao, "Automated crowdturfing attacks and defenses in online review systems," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas, USA, 2017, pp. 1143–1158.

[14] L. Rokach, "Ensemble-based classifiers," in *Artificial Intelligence Review*, vol. 33, no. 1. Springer, 2010, pp. 1–39.

[15] X. Zhang, H. Zheng, X. Li, S. Du, and H. Zhu, "You are where you have been: Sybil detection via geo-location analysis in osns," in *2014 IEEE Global Communications Conference*, Austin, USA, 2014, pp. 698–703.

[16] Q. Gong, Y. Chen, X. He, Z. Zhuang, T. Wang, H. Huang, X. Wang, and X. Fu, "Deepscan: Exploiting deep learning for malicious account detection in location-based social networks," in *IEEE Communications Magazine, Feature Topic on Mobile Big Data for Urban Analytics*, vol. 56, no. 1, 2018.

[17] R. Deng, N. Ruan, R. Jin, Y. Lu, W. Jia, C. Su, and D. Xu, "Spamtracer: Manual fake review detection for o2o commercial platforms by using geolocation features," in *International Conference on Information Security and Cryptology*. Springer, 2018, pp. 384–403.

[18] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007, pp. 423–430.

[19] W. Yao, N. Ruan, F. Yu, W. Jia, and H. Zhu, "Privacy-preserving fraud detection via cooperative mobile carriers with improved accuracy," in

Journal Pre-proof

Journal Pre-proof

2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking, San Diego, USA, 2017, pp. 1–9.

[20] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "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*, vol. 1, Stroudsburg, USA, 2011, pp. 309–319.

[21] C. Chen, K. Wu, V. Srinivasan, and X. Zhang, "Battling the internet water army: Detection of hidden paid posters," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Niagara Falls, Canada, 2013, pp. 116–120.

[22] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *2014 IEEE International Conference on Data Mining*, Shenzhen, China, 2014, pp. 899–904.

[23] S. Kumar, X. Gao, I. Welch, and M. Mansoori, "A machine learning based web spam filtering approach," in *IEEE 30th International Conference on Advanced Information Networking and Applications*, Crans-Montana, Switzerland, 2016, pp. 973–980.

[24] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao, "Bimodal distribution and co-bursting in review spam detection," in *Proceedings of the 26th International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland, 2017, pp. 1063–1072.

[25] M. Agrawal and R. Leela Velusamy, "Unsupervised spam detection in hyves using salsa," in *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, New Delhi, 2015, pp. 517–526.

[26] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 2016, pp. 895–904.

[27] H. Zheng, M. Xue, H. Lu, S. Hao, H. Zhu, X. Liang, and K. W. Ross, "Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks," in *The 2018 Network and Distributed System Security Symposium*, San Diego, USA, 2018.

[28] J. Schmidhuber, "A neural network that embeds its own meta-levels," in *IEEE International Conference on Neural Networks*, vol. 1, 1993, pp. 407–412.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural computation*, vol. 9, no. 8, 1997, pp. 1735–1780.

[30] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, USA, 2016, pp. 3818–3824.

[31] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," in *Information Sciences*, vol. 385, 2017, pp. 213–224.

[32] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proceedings of the 2017 ACM on Multimedia Conference*, California, USA, 2017, pp. 795–816.

[33] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the 1995 Computational Learning Theory: Second European Conferencem*, Berlin, Heidelberg, 1995, pp. 23–37.

[34] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear estimation and classification*. Springer, 2003, pp. 149–171.

[35] ——, "Explaining adaboost," in *Empirical inference*. Springer, 2013, pp. 37–52.

[36] A. Prasad, L. Iverson, and A. Liaw, "Newer classification and regression tree techniques: Bagging and random forests for ecological prediction," in *Ecosystems*, vol. 9, no. 2, 2006, pp. 181–199.

[37] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," in *Computational Statistics & Data Analysis*, vol. 52, no. 4, 2008, pp. 2249–2260.

Journal Pre-proof

Journal Pre-proof

Na Ruan is currently an Assistant Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. She received the B.S. degree in Information Engineering and the M.S. degree in Communication and Information System from China University of Mining and Technology in 2007 and 2009 respectively. She received Ph.D. degree from the Department of Information Science and Electrical Engineering, Kyushu University, Japan in 2012. Her current research interests include network security and privacy protection.

Ruoyu Deng received the B.S degree in Shanghai Jiao Tong University, China, in 2018. He is currently a master student of Computer Science and Engineering in Shanghai Jiaotong University. His research interests include privacy protection and big data.

Chunhua Su received the B.S. degree for Beijing Electronic and Science Institute in 2003 and received his M.S. and PhD of computer science from Faculty of Engineering, Kyushu University in 2006 and 2009, respectively. He is currently working as an Associate Professor in Division of Computer Science, University of Aizu. He has worked as a research scientist in Cryptography & Security Department of the Institute for Infocomm Research, Singapore from 2011-2013. From 2013-2016, he has worked as an Assistant professor in School of Information Science, Japan Advanced Institute of Science and Technology. From 2016-2017, he worked as Assistant Professor in Graduate School of Engineering, Osaka University. His research interests include cryptanalysis, cryptographic protocols, privacy-preserving technologies in data mining and IoT security & privacy.

Journal Pre-proof

Journal Pre-proof

DECLARATION OF INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Journal Pre-proof