Effective Machine Learning Approach to Detect Groups of Fake Reviewers

Jayesh Soni  
School of Computing and Information Sciences  
Florida International University  
Miami, USA  
jsoni@fiu.edu

Nagarajan Prabakar  
School of Computing and Information Sciences  
Florida International University  
Miami, USA  
prabakar@cis.fiu.edu

Abstract—Social networks have become very popular in recent years due to the exponential growth of the personalized computing devices. Twitter, Facebook, and LinkedIn are the evidence of massive online social networks. Product reviews through these social media are now widely used by individuals and organizations for their decision making. The trustworthiness of these reviews is unreliable when businesses generate fraudulent positive reviews for themselves or negative reviews for their competitors. For reviews to reflect genuine user experiences and opinions, such spam reviews need to be detected. Prior works focused on fake reviews and individual fake reviewers. However, to target a particular product, fake reviewers work collaboratively in groups and control the sentiment of the product. Even more challenging is there might be one real person creating multiple fake IDs to write several reviews for the same product. This paper focuses on finding such group of fake reviewers. The proposed method first uses a deep-walk approach to find behavioral representations for each fake reviewer account. Subsequently, we use several supervised and unsupervised machine learning algorithms to identify the groups of fake reviewers.

Keywords—fake reviewer groups, machine learning, deep-walk

I. INTRODUCTION

Nowadays, before buying any product, most people will first read the reviews of the product. In most cases, people are likely to buy the product if they find the majority of the reviews are positive. But, if most of the reviews are negative, they will certainly go for another product. Positive reviews in the product can result in significant financial gains and increased popularity for that organization and individuals. This gives rise to opinion spamming where organization hires reviewers to promote their product or damage the reputation of other organization product.

Research has been increased in this field since Jindal and Liu first worked on the problem of detecting fake review reviewers in 2008 [1]. Early works focused on detecting individual fake reviewers [1,2,3,4,5]. To circumvent this detection, reviewers operated collaboratively in groups with fake IDs. This led to the trend in analyzing fake reviewer groups instead of individual fake reviewers [6,7,8,9,10,11,12,13]. Earlier, supervised learning approaches were used that require datasets to be labeled to train classifiers. However, such methods are often inaccurate due to the lack of ground truth dataset for modeling and evaluation. To overcome this limitation, many unsupervised methods are proposed to detect fake reviewers, e.g. Markov Random field [2,7,9].

Detecting fake reviewer groups is not so extensively addressed as compared to detection of individual fake reviewers. Early work in this area uses the frequent item-set mining technique to generate candidate spammer group and then build a model to differentiate them into spam and non-spammer reviewer groups [6,8,11,12]. However, such approach can only detect tight spammer groups where each member of a group has to review all the targeted product as discussed in [10]. Further, spammer group often works loosely, i.e. reviewers are not required to review all targeted products. We propose a non-traditional deep-walk based approach to detect groups of fake reviewers using only the reviewer information from social network data. Our approach does not use review text analysis since it is often unreliable and inefficient [5,12,14].

We introduced a top-down computing framework, to identify a group of fake reviewers based on the topological structure of the reviewer graph. We treat the whole data as a graph-based structure where nodes represent reviewers and edges represent the number of apps reviewed by them together.

Our model uses the deep-walk approach by recursively breaking the graph into sub-graphs by taking a random walk on the graph. Further, it uses the word2vec model to generate the representation for each node.

The remaining sections of the paper are organized as follows. The next section presents the related work. In the subsequent section, we introduce our framework. In Section IV we illustrate the application. In the last two sections, we present the future scope of the work and the concluding remarks.

II. RELATED WORK

The problem of detecting fake reviews/reviewers have gained much interest in recent years. It can be summarized into three categories: fake reviewer detection, fake review content detection and detection of groups of fake reviewers. For example, Ott. et al. [5] use linguistic features analysis of review text to identify fraudulent reviews ; Liu et al. [1] employed duplicate reviews as fake reviews to train classifiers; Xie et al. [15] used temporal analysis to detect reviewers who write singleton reviews; Lim et al. [4] use behavioral features in rating patterns to detect fraudulent reviewers.
Recently, there is an increased interest in detecting groups of fake reviewers. Mukherjee et al. [8] introduced frequent item-set mining technique to generate candidate review spammer groups that take reviewer as items, and targeted products as transactions. Based on these candidate groups, many other computing frameworks have been proposed to evaluate the suspiciousness of spammers. Xu et al. [12] introduce a KNN-based approach to detect the labels for each reviewer. [8] proposed GSRank to rank candidate groups that capture the relationship among candidate groups, target products, and individual reviewers.

Leman et al. [16] propose FRAUDEAGLE framework which uses the relational structure among reviewers and products to rank fake reviewers. Shebuti et al. [9] propose SPEAGLE, that extends FRAUDEAGLE with the introduction of review nodes and additional information (e.g., star ratings, timestamps, etc.) that greatly improve the ranking precision. Xu et al. [17] propose FRAUDINFORMER framework to detect a group of fake reviewers via heterogeneous pairwise features extracted from rating behaviors and linguistic patterns.

Unlike the above-mentioned approaches to detect a group of fake reviewers, we propose a deep-walk based computing approach which is solely based on the topological structure of the reviewer graph revealing the behavioral similarity between the reviewers.

Review spamming techniques are evolving continuously. Although many spamming detection techniques are being proposed by researchers, there is no overall success in discovering all kinds of spamming strategies. The best strategy is to use a combination of relevant techniques.

III. PROPOSED FRAMEWORK

Our framework consists of four phases as represented in Figure 1.

A. The Data
1) The graph
We have review data for 640 apps (Google Play Store) with a co-review graph for each app. Each vertex of a co-review graph represents a reviewer account and each edge indicates the number of apps co-reviewed by the reviewer accounts of the associated vertices. We integrated all 640 co-review graphs into one unified graphs with 38123 nodes and 3572409 edges.

2) Ground Truth
A set of multiple fake reviewer accounts belongs to a unique individual reviewer. The ground truth cluster data contains 2207 nodes (reviewer accounts) of 23 clusters (individual reviewers).

B. Deep-walk approach
To get a high-level representation of the reviewers, we used the deep-walk based neural network designed by Perozzi et al. [18]. Deep-walk uses techniques from neural language modeling to model community structure in networks. The only required input is a corpus and a vocabulary. Deep-walk considers a set of short truncated random walks as its corpus, and the graph vertices as its vocabulary. In short, it consists of random-walk and the word2vec model.

Word2vec: This model and its application by Mikolov et al. [19] have attracted a great amount of attention in recent two years. The vector representations of words learned by word2vec model have been shown to carry semantic meanings and are useful in various NLP tasks. It basically uses the Skip-gram model.

The Skip-Gram model: It is constructed with the focus word as the single input vector, and the target context words are at the output layer. The activation function for the hidden layer simply amounts to copying the corresponding row from the weight matrix \( W_1 \) (linear). At the output layer, it outputs \( C \) multinomial distributions instead of just one. The training objective is to minimize the summed prediction error across all context words in the output layer.

Deep-walk needs to be performed with several range of values for each of the following parameters:

- \( NW \): No. of walks starting from each node
- \( T \): Walk length (the number of nodes on the path)
- \( D \): Distance traversed (newly introduced)
- \( R \): Representation-size (number of features for the path)
- \( W \): Window size
We ran the deep-walk algorithm for a range of parameter values. For each parameter value, deep-walk outputs a Word2Vec matrix file where the rows are the reviewer ID’s, and columns are the feature vectors (representation of the nodes). We only process the reviewer ids of the Word2Vec file for which we have the ground truth reviewer accounts to identify the most suitable parameters for deepwalk.

C. Training a machine learning model
To find the optimal representation of the reviewer accounts, we employed supervised machine learning approaches. We used three most widely used machine learning models namely Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), and Random Forest (RF). Each algorithm requires configuration of its hyper-parameters. We used K-Fold Cross-validation and Grid Search cross-validation approach to find the optimal hyper-parameter values. With this approach, we finally found the best Word2Vec file that has the highest classification accuracy as shown in Table 1.

<table>
<thead>
<tr>
<th>Machine Learning Model</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>81%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>85%</td>
</tr>
<tr>
<td>SVM</td>
<td>87%</td>
</tr>
</tbody>
</table>

Still, we do not know the association of reviewer accounts that are not present in the ground truth data, with human reviewers. A cluster of reviewer accounts is the set of reviewer accounts related to the same human reviewer. We use the entire best word2vec file to find clusters for all unprocessed reviewer accounts of that file.

D. Clustering approach
We applied clustering algorithm namely K-Means clustering algorithm on this best Word2Vec file. In K-Means algorithm, we specify value for the parameter K to form the reviewer accounts into K number of clusters. Since there is no well-known method to find the best value of K, we use the following three different approaches:

1) Elbow Method
It computes the sum of squared error (SSE) for some values of K. The SSE is defined as the sum of the squared distance between each member of the cluster and its corresponding centroid as in (1).

$$\sum_{j=1}^{k} \sum_{i: x_i = j} \| x_i - \mu_j \|^2_2$$  \hspace{1cm} (1)

When we plotted k against the SSE, we observed that the error decreases as K gets larger; this is because when the number of clusters increases, the cluster size should be smaller, hence the distortion is also smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly.

From the graph in Figure 2, we can say that the ideal no. of clusters are in the range from 400 to 700. Further, to find the optimal value of K, we use silhouette measure.

2) Silhouette Measure
The Silhouette Coefficient is defined for each data point and is composed of two scores:

- $\mu_{in}(X_i)$: The mean distance between a data point and all other points in the same cluster.
- $\mu_{out}(X_i)$: The mean distance between a data point and all other points in the next nearest cluster.

The Silhouette Coefficient S for a single data point is then given as in (2):

$$S_i = \frac{\mu_{out}(X_i) - \mu_{in}(X_i)}{\max(\mu_{out}(X_i), \mu_{in}(X_i))}$$  \hspace{1cm} (2)

The silhouette coefficient is the mean silhouette coefficient of all data points of the data file that represents the number of optimal clusters for the data set.

We first calculated the silhouette coefficient for ground truth data points with the number of clusters ranging from 26 to 36, to cross-check the correctness of this method. We found it matched with the number of clusters of the ground truth data set (23 clusters) as illustrated in Figure 3.

Subsequently, we computed the silhouette coefficient for all data points of the best word2vec file with the number of clusters ranging from 400 to 700. The optimal number of clusters for the data set is 430 as shown in Figure 4.
3) Average Measures

We used the mean of the following measures to find the optimal number of clusters, K:
- Homogeneous
- Completeness
- Silhouette
- Adjusted Rand Index

We observed in this method also the highest value of K is at 430 as depicted in Figure 5.

IV. APPLICATION IN FUTURE FRAUDULENT DETECTION

If a particular reviewer reviews a new app, and we want to know whether the review is suspicious or not, instead of looking for all reviewers of the entire dataset, we can narrow down our search to the cluster the reviewer belongs to. After finding the cluster for the suspicious reviewer, we have to find that whether the other reviewers from that cluster have reviewed that same new app or not. If most of the reviews from the cluster for the same app have a significant overlap with the suspicious review, then we can conclude that the review is fraudulent.

V. FUTURE SCOPE

The present investigation clusters groups of reviewers to find the optimal number of clusters of all reviewers that matches with the ground truth data. The scope of the work can be expanded in future to include text-based modeling using review text, star rating, etc. In the present work, we use clusters of fake reviewers to find cluster for all reviewers. We will extend this work by taking into account the types of apps to identify the type of apps that are reviewed most commonly by fake users (reviewers). Furthermore, our proposed methodology which uses deep-walk analysis is not only limited to clustering group of fake reviewers, but it can also be used in a variety of applications. For instance, we can assess customer purchase behavior through a finite number of categories using a small set of ground truth data. We can apply a similar strategy to the financial market analysis.

VI. CONCLUSION

We used a large social media data set of online app reviewers along with the ground truth cluster of fraudulent reviewers. The objective is to cluster all reviewers for future fraudulent reviewer’s validation. We employed a deep-walk approach to characterize different reviewers. As this approach depends on many parameters, we experimented with a wide range of values for each parameter. To speed up this process, we parallelized
the task using massive computational resource. We evaluated the different combination of parameters and found the most suitable combination that gives the highest classification accuracy. Next, we performed K-Means clustering method for a different range of clusters. Finally, using elbow method and silhouette coefficient we identified the optimal no. of clusters that match the ground truth clusters. In this approach, the large volume of the dataset and the need for a massive computational resource for this analysis is a concern. Nonetheless, it is desirable to perform this analysis offline, possibly in a cloud environment, to expedite the real-time validation of future suspicious reviewers quickly. We conducted experiments on real-world datasets with/without ground-truth to evaluate the performance. Experimental results show that our proposed framework detects fraud reviewers groups and outperforms several state-of-the-art approaches, including both graph based and non-graph based methods.

REFERENCES


