Analyzing and Detecting Collusive Users Involved in Blackmarket Retweeting Activities

UDIT ARORA, HRIDOY SANKAR DUTTA, BRIHI JOSHI, ADITYA CHETAN, and TANMOY CHAKRABORTY, IIIT-Delhi, India

With the rise in popularity of social media platforms like Twitter, having higher influence on these platforms has a greater value attached to it, since it has the power to influence many decisions in the form of brand promotions and shaping opinions. However, blackmarket services that allow users to inorganically gain influence are a threat to the credibility of these social networking platforms. Twitter users can gain inorganic appraisals in the form of likes, retweets, and follows through these blackmarket services either by paying for them or by joining syndicates wherein they gain such appraisals by providing similar appraisals to other users. These customers tend to exhibit a mix of organic and inorganic retweeting behavior, making it tougher to detect them.

In this article, we investigate these blackmarket customers engaged in collusive retweeting activities. We collect and annotate a novel dataset containing various types of information about blackmarket customers and use these sources of information to construct multiple user representations. We adopt Weighted Generalized Canonical Correlation Analysis (WGCCA) to combine these individual representations to derive user embeddings that allow us to effectively classify users as: genuine users, bots, promotional customers, and normal customers. Our method significantly outperforms state-of-the-art approaches (32.95% better macro F1-score than the best baseline).

CCS Concepts: • **Information systems → Social networks**; • **Security and privacy**

Additional Key Words and Phrases: Retweeters, collusion, blackmarket, Twitter, OSNs, multiview learning

ACM Reference format:
Udit Arora, Hridoy Sankar Dutta, Brihi Joshi, Aditya Chetan, and Tanmoy Chakraborty. 2020. Analyzing and Detecting Collusive Users Involved in Blackmarket Retweeting Activities. *ACM Trans. Intell. Syst. Technol.* 11, 3, Article 35 (April 2020), 24 pages.
https://doi.org/10.1145/3380537

A part of this research was published in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'18) [21]. U. Arora and H. S. Dutta have equal contributions. B. Joshi and A. Chetan have equal contributions.

The authors would like to acknowledge the support of Ramanujan Fellowship, DST (ECR/2017/00l691), SPARC (SPARC/2018-2019/P620), and the Infosys Centre of AI, IIIT-Delhi, India. T. Chakraborty would like to thank the support of the Google India Faculty Award.

Authors' addresses: U. Arora, H. Sankar Dutta (corresponding author), B. Joshi, A. Chetan, and T. Chakraborty, IIIT-Delhi, Okhla Industrial Estate, Phase III, New Delhi, India - 110020; emails: uditarora09@gmail.com, {hridoyd, brihi16142, aditya16217, tanmoy}@iiitd.ac.in.

2157-6904/2020/04-ART35 $15.00
https://doi.org/10.1145/3380537

1 INTRODUCTION

Twitter, being a micro-blogging site, provides a platform to share various kinds of content—personally curated content, product promotions, and opinions that give rise to online social movements. It also provides its users an option to appraise other tweets based on their liking of the content in two different ways: (i) *Content-level affirmation* (primary method of Twitter appraisal), such as `Retweets`, `Quotes`, and `Likes`—this provides a way of rebroadcasting messages and confirms the user's agreement with the message being important to others and (ii) *User-level affirmation* (secondary method of Twitter appraisal), such as `Follows`, implying the general liking of most of the tweets by that user.

The count of these appraisals, such as number of likes, retweets, or follows, indicates a sense of crowd-sourced agreement on that tweet and user, and thus, they also determine the influence of the tweet as well as the author of the tweet. Most of the time, excessive appraisal of certain content or user can make that topic or user *trend* on Twitter. This system of appraisal on Twitter creates motivation to falsely gain excessive number of likes, retweets, or follows. This has also led to the creation of certain **blackmarket services** that facilitate such inorganic appraisal (see Section 3). These blackmarket services are either paid or composed of a group of individuals who are ready to barter appraisals among themselves. These users, who are involved in "intentional manipulation" using such blackmarket services and who try to create a false impression of the popularity of their tweets or accounts through a higher number of retweets, likes, or follows, are referred to as **collusive users**. In this study, we focus on detecting users involved in collusive retweeting activities (*henceforth called as "collusive retweeters"*). For our purposes, we use the notation *collusive users*, *collusive retweeters,* and *customers* interchangeably in this article, all referring to *collusive retweeters*. Although we are interested in detecting individual collusive users, the term *"collusion"* has been used, because these users act in deceitful cooperation via blackmarket services, which is in contempt of Twitter's Terms of Service.

Technical challenges of detecting collusive retweeters: The blackmarket services that provide a platform to these collusive retweeters have set up an intelligent ecosystem for providing inorganic appraisal to these users. Users can gain retweets on their own tweets for no cost at all by retweeting the tweets submitted by other users on the platform, making them a part of the ecosystem. Furthermore, the design of this ecosystem makes their detection even more challenging because:

- Collusive retweeters are not completely bots, but mostly human accounts. Therefore, bot detection algorithms perform poorly in detecting them (first column of Table 1).
- Collusive retweeters cannot be categorized into spam or fake accounts either, as these accounts also show significant amount of genuine user activity with a well-maintained profile. Hence, spam or fake account detection algorithms also do not detect them accurately (second and third columns of Table 1).
- They show a mix of *organic* and *inorganic* retweeting activity, i.e., they participate in retweeting activity out of genuine interest for content, but also for retweeting the tweets that other users have put up on these blackmarket services to gain retweets for their own tweets. Furthermore, the proportion of organic and inorganic activity that a user might display is not fixed. It depends upon each individual's own behavior.
- Last, collusive retweeters are not synchronous in their retweeting activities. Hence, algorithms that leverage the synchronous behavior of fraudulent retweeters, like Reference [25], fail to make a mark when it comes to detection of collusive retweeters (fourth column of Table 1). We detail the asynchronous behavior of these collusive retweeters below.

Table 1. Macro F1-score of the Competing Methods

Bot Detection [15]	Spam Detection [53]	Fake Account Detection [23]	Sync. Fake Retweeter Detection [25]	Our Method
0.680	0.645	0.688	0.269	0.869

None of the existing methods can detect collusive retweeters accurately.

There has been a lot of research on detection of fraudulent activities on Twitter, such as detection of bots, fake followers, and social spam detection. However, the problem of detecting manipulation of content affirmation is largely untouched. Giatsoglou et al. [25] recently tried to tackle this problem by stating that spam retweeters have a synchronous behavior in terms of their retweeting activity and retweet the same content at the same time. However, we observe that such synchronous behavior is not followed by collusive users, as shown in Figure 1 of Reference [21]. It shows that, unlike normal retweet fraudsters where Arr-MAD (mean absolute deviation of retweet's inter-arrival times of retweet threads) is almost invariant with respect to lifespan, collusive retweeters show an increasing trend. This is because most of these users who are involved in blackmarket services are users seeking to appraise other content to get appraisal on their own content. Thus, they may not have any motive for targeted appraisal of any particular tweet. Also, blackmarket services do not mandate their users to retweet content at a particular time. Thus, these services have no control over the synchronicity of the collusive users. Aggarwal et al. [2] made a recent attempt at detection of collusive followers, but not towards collusive retweeters. They also pointed out that the existence of such blackmarket services is a threat to the credibility of Online Social Networks (OSNs). It is also important to notice that even the in-house algorithms of Twitter have been unsuccessful at detecting such collusive retweeters—based on the observation that in our dataset, 72 collusive retweeters were actually verified users marked by Twitter.

In our previous study [21], we provided a supervised method to detect collusive retweeters affiliated to blackmarket services. We had only explored user attributes to mark their retweeting pattern and only relied on attribute-level features that can be extracted from the user's own profile only, such as their retweet intervals, the length of their profile name, and so on [22]. However, collusiveness is a multifaceted characteristic, and we can detect it in a better way by utilizing both attribute-level as well as network-level features, where we study features extracted from the user's Twitter network. In this article, we present a comprehensive evaluation of collusive retweeters from a multiview perspective, along with our previous study. We also show the superiority of our new approach over our previous work [21], as well as other state-of-the-art approaches dealing with fake activity detection. In particular, the contributions of this article are as follows:

(1) We present a detailed study of the collusive activities controlled by the blackmarket services. To our knowledge, this is the first work that presents the entire landscape of the blackmarket activities in a thorough and rigorous manner.
(2) We propose a multiview learning-based approach to detect collusive retweeters involved in blackmarket services. The idea behind the approach is to create user representations from the content, attributes, and social network of the users. We create six different views: two attribute-level views and four network-level views.
(3) We describe an approach to learn a combined representation for a user from multiple views using WGCCA. We show how each view can be more or less helpful for our task (using t-SNE visualizations) and weigh each view differently for optimal performance.
(4) We conduct experiments by training state-of-the art classifiers on the combined user representations to detect three types of collusive users: bots, promotional collusive users, and normal collusive users. For this multi-class classification problem, our approach

outperforms the best baseline (BotoM) by 32.95%, and our previous work (ScoRe) by 4.55% in terms of F1-score (macro). We also conduct an experiment by combining all types of collusive users to consider the problem as a binary classification problem. Here, we find that our approach outperforms the best baseline (FakeAcc) by 26.31% and our previous work (ScoRe) by 14.34% in terms of F1-score (macro).

(5) We collect a novel dataset of 1,807 collusive and 2,706 genuine users (data collected between March 2018 and August 2018). All the collusive users are manually annotated by human annotators into three categories: bots, promotional collusive users, and normal collusive users, based on the guidelines given to them. We also collect the profile information and timelines of these users.

The remainder of this article is composed of the following sections: Section 2 reviews the related work. We discuss the different types of blackmarket services in Section 3. Section 4 describes our dataset. Section 4 also outlines the guidelines given to the human annotators for labeling of collusive users. We present the user representations using attribute- and network-level features in Section 5. Section 6 contains the experiments conducted using this dataset, and Section 7 describes the results we obtained. Section 9 closes the article with concluding remarks.

2 RELATED WORK

Several anomaly detection techniques have been designed to identify malicious individuals, online fraudsters, spammers, and so on, across multiple online platforms. Despite the fact that a lot of literature exists on detecting these fraudulent users and bots in various OSNs, detection of collusive activities has hardly been studied. We discuss the past efforts by dividing the existing literature into two parts: (i) detection of fraudulent activities and (ii) study of collusion in OSNs.

2.1 Detection of Fraudulent Activities in OSNs

Various studies have been conducted on the detection of fraudulent and spamming activities on online media. Shah et al. [45] provided a detailed analysis of the blackmarket services and divided them into two types based on the mode of service—premium and freemium. Benevenuto et al. [4] generated features from tweets and user behavior to detect spammers. Giatsoglou et al. [26] proposed RTGen, which uses a weighted cascade model to simulate the retweeting activity of genuine and fraudulent users. Chu et al. [13] classified a user into human, bot, or cyborg based on tweeting behavior, tweet content, and user account properties. Hu et al. [31] identified spammers based on the networking properties. Gupta et al. [28] investigated spam campaigns by detecting spammers who use phone numbers to promote these campaigns on Twitter. A huge amount of work has been done on identifying bots on Twitter [9, 15, 18]. Some other studies focused on detecting frauds using URLs embedded in tweets [37, 38, 54] and blacklisted URLs [24, 27]. Recently, a series of works investigated fake followers on Twitter. It is reported that there is an increase of 1%–3% in fake followers for a Twitter account.[1] Castellini et al. [8] designed an anomaly detector using denoising autoencoder to detect fake followers. Cresci et al. [14] used 49 distinct features and 8 different "glass-box" and "black-box" machine learning classifiers to detect fake followers on Twitter. References [32, 41] are some of the network-based approaches to detect fake followers on Twitter. There also exists a tool, called Fake Follower Check,[2] that detects fake followers based on features such as ratio of friends to followers, very many retweets than tweets, incessant use of spam phrases such as "diet," "make money," and so on. Giatsoglou et al. [25] proposed NDSync to

[1] https://www.gshiftlabs.com/social-media-blog/the-fake-followers-epidemic/.
[2] https://www.socialbakers.com/blog/1099-fake-followers-check-a-new-free-tool-from-socialbakers.

tackle the problem of synchronous fraudulent activities. NDSync spots retweet fraudsters based on features collected from the retweet threads.

Several other studies attempted to detect fraudulent and spam activities on different social media platforms. Beutel et al. [6] detected lockstep behavior in Facebook Page Likes. Jindal et al. [33] investigated spam detection on e-commerce websites such as Amazon. Li et al. [39] identified review spam by employing supervised machine learning methods based on the crawled reviews from Epinions. Chen et al. [11] detected fake views caused by robots generating requests or reports in video platforms.

2.2 Study of Collusion in OSNs

Though collusion has not been studied much in the literature, it has recently gained a considerable amount of attention among researchers because of the techniques used to offer the services. Thomas et al. [49] investigated underground market profiting from Twitter credentials and its effects. They identified 27 account traders from blackhat forums, freelance websites, and so on, linked to Twitter, who bypass the automatic registration using compromised hosts and CAPTCHA solvers. Liu et al. [40] tackled the problem of a new type of malicious crowdturfing following relationship, called "voluntary following." They proposed DetectVC, which incorporates both structural information in user behavior graphs and prior knowledge gained from the follower markets. Motoyama et al. [42] analyzed six different underground forums to understand the dynamics of social networks present on these forums. Stringhini et al. [47] analyzed the growth and dynamics of Twitter follower markets. They reported the properties of the customers of these markets. Arora et al. [3] detected tweets posted on blackmarket sites by using a multitask learning framework to classify tweets as blackmarket or genuine. Zheng et al. [56] identifed sockpuppets—a set of aliases (different user-IDs) controlled by a single user (aka "Puppetmaster") in online discussion forums. Kumar et al. [36] studied sockpuppetry across nine online discussion communities. They used behavioral traces of a user such as IP addresses and data collected from a user session to identify sockpuppet groups. They also revealed that sockpuppets use more singular first-person pronouns, write shorter sentences, and swear more and participate in more controversial discussions. Solorio et al. [46] proposed a semi-supervised approach to detect linked identities in Wikipedia. Chetan et al. [12] proposed CoReRank, an unsupervised method to detect collusive retweeters. Gupta et al. [29] proposed an approach to detect malicious retweeter groups.

Several other studies reported collusiveness on other platforms such as e-commerce sites [16]. Chen et al. [10] detected multiple algorithmic pricing strategies adopted by sellers on Amazon Marketplace. They identified how sellers change prices of their product to win the Buy Box more frequently. Hannak et al. [30] studied two prominent online freelance marketplaces of online labor markets and reported how real-world bias can manifest these markets and harm the employment opportunities. Vidros et al. [52] examined the diverse aspects of employment scams and showed their resemblance with the existing fraudulent activities such as vandalism, cyber bullying, and so on.

2.3 Differences with Our Previous Studies [20, 21]

In one of our previous studies [21], we attempted to detect collusive retweeters using a supervised approach. We focused on the freemium service model, mainly due to easy accessibility of data. We divided the freemium services further into three types: social-share services, credit-based services, and auto-time retweet services. We then collected a set of 64 features that can help distinguish the different types of users. The features were then used to run six state-of-the-art classification algorithms.

This study was further extended where we provided an in-depth analysis of collusive users based on features from user profile, timeline, and network involved in both freemium and premium blackmarket services and classified users as premium customers, freemium customers, and genuine users [20]. In the current article, our methodology differs from both References [21] and [20] in the following ways:

- We create task-specific representations of users using their content and network-level features.
- We demonstrate the utility of these representations for detecting collusive users.
- With respect to Reference [21], we extend our dataset of collusive and genuine users from 753 and 1,000 users to 1,807 and 2,706 users, respectively.
- We create a user representation using the features in Reference [21] and combine it with other representations to improve the overall performance.

3 TWITTER AND BLACKMARKET SERVICES

3.1 What is Twitter Appraisal?

There are various kinds of content shared every day on micro-blogging sites such as Twitter. The content can range from personal opinions, publicity of products, quotes, facts, and simple online content promotion. There are two main ways to get more attention to such tweets:

- The user who posts the tweets takes several measures to make their tweet look attractive. They add emoticons, images, hashtags, links, and mention other users. By doing so, their tweet is displayed on the timeline of other people who might follow this particular hashtag or people who follow the account that was mentioned in the tweet. We call this approach the **Primary Method** of obtaining Twitter appraisal.
- Each tweet on Twitter can also gain attention with the help of other Twitter users in the community. These users have an option to like or retweet the given tweet or also follow the tweet creator for future updates on similar tweets. Thus, this tweet appears on the feed of the users who are connected to these users. This creates a cascading effect and is called the **Secondary Method** of obtaining Twitter appraisal.

3.2 What are Blackmarket Services?

It is often observed that the Primary Method of obtaining Twitter appraisal tends to be tedious and cumbersome. It requires the tweet poster to curate the content that can attract the highest amount of Twitter users. Thus, this is not the most efficient way of gaining popularity, as it only favors content that is, in general, very popular; and with a large number of posts tweeted every millisecond, a majority of the tweets go unnoticed. Thus, most people resort to the Secondary Method. Even with the Secondary Method, there is an important hindrance. Most content that will be liked or retweeted by the other users caters to their interests. They would not want to appraise any other tweet that is not even remotely aligned to their interests. Blackmarket services, thus, try to eradicate this problem by creating a community of users who can appraise any tweet, given an incentive.

The blackmarket services provide services for various OSNs, e.g., Facebook (followers, likes, shares, comments), Twitter (followers, retweets, likes), Instagram (followers, likes, comments). Other than OSNs, the blackmarket services also provide service to video subscription-sharing platforms, e.g., YouTube (views, subscribers, likes, comments), Vimeo (plays, followers), music-sharing platforms, e.g., SoundCloud (plays, followers, likes, reposts, comments), ReverbNation (fans), business- and employment-oriented platforms, e.g., LinkedIn (followers, connections, endorsements). Shah et al. [45] divided the blackmarket services into two types based on the model of service—*Premium* and *Freemium*.

3.2.1 Premium Services. If a customer pays a platform to receive appraisals, then the platform is said to be offering premium services. Most of these purchases are divided into tiers that cost according to their quantity—for example, purchasing retweets happens in batches of 100, 1K, 5K, and so on. Almost every platform has a different way of operating. *SocialShop*[3] asks the customers for only their Twitter Username before payment—after which they can provide the link of the tweets they want to appraise. Premium services also ensure that their customers have fully completed bios and have eye-catching profile pictures. Apart from selling packages based on the quantity of the service, some premium services such as *GettwitterRetweet*[4] and *SlickSocials*[5] are also customizable to accommodate the time at which the customer wants to receive most of the appraisals. For example, if a Twitter movement is taking place, a customer can pay to get the most number of likes in their tweet at that moment so his/her tweets can get highlighted in the movement. For content-based tweets, a customer can also choose the target audience that would be the most appropriate for them. *Twesocial*[6] provides such target-specific services where it requests the customers to enter the hashtags they want to target for a particular tweet.

Like any other business model, premium services often rope in customers by offering attractive deals. *Devumi*[7] and *SocialShop* include a 100% money-back guarantee if they are unable to deliver the services that were mentioned in their packages. Some services like *tweetboost*[8] also offer a one-time free trial period to customers so they can avail the services initially without installment and commitments. Premium services like *buyrealmarketing*[9] also provide pre-paid subscription plans to customers. On their platform, a user can opt for a single payment, monthly subscription, three months prepaid, or even six months prepaid subscription. As much as the question of the legality of such services is thought upon, platforms like *Devumi* mention on their FAQ page that they only boost the social media "presence" of an account, so it is completely legal. One can also set a daily limit on the number of appraisals one needs on his/her account. *Socialmediadaily*[10] can also accommodate more than one tweet for appraisal if a higher-costing package is bought from the service.

3.2.2 Freemium Services. Unlike premium services that require payment from the customers, several other services work on different operating models that do not require payment. Such services are termed as freemium services. Some of them start with a basic free service, which gets the customers hooked—thereby motivating them to subscribe to the platforms and also publicize the platforms via word-of-mouth. A freemium service functions by creating a community of customers. Thus, each member of the community avails the facilities by allowing other members to promote the content. This implies that once a user is a part of a freemium service, he/she is a customer as well as a service provider, with the service merely as an interface for such communication.

Most of the freemium services operate by having a dashboard, called the "Earning Area," where the customers can view the content of other customers. If the customers appraise the other customers—by liking their content, retweeting it, or following them—they earn credits. These credits are utilized when the customers want to get appraisals for their own tweets via the service. Some of the freemium services (e.g., Like4Like,[11] YouLikeHits[12]) also have a referral system for

[3] https://www.socialshop.co/.
[4] https://www.gettwitterretweet.com/.
[5] https://slicksocials.com/.
[6] https://www.twesocial.com/.
[7] https://devumi.com.
[8] http://tweetboost.net/.
[9] https://www.buyrealmarketing.com/.
[10] https://www.socialmediadaily.com/.
[11] https://www.like4like.org/.
[12] https://www.youlikehits.com/.

gaining credits, i.e., the customer gets points when someone joins the service using their referral code. One can also purchase credits from the freemium services using online payment systems. Freemium services (e.g., Like4Like, TraffUp[13]) also provide daily bonus points for staying active on their website.

Freemium services can be divided into three categories:

(1) **Social-share services**: Social-share services ask the customers to appraise the content of other customers on social media. Often, it is by content-matching (like hashtag matching, topical similarity of tweets) of one customer with another. This is a rather slow process, as it is not guaranteed that the other customers would appraise the content.
(2) **Credit-based services**: In credit-based services, customers gain credits to appraise other customers' content, which can then be utilized in a similar fashion for their own content. This creates some sort of a give-and-take relationship, which is a very fast and effective model.
(3) **Auto-retweet services**: In auto-retweet services, the services create several accounts internally. A customer is supposed to use an access token from Twitter to log in to the service. Only limited service is provided on these platforms, which is again limited by a fixed time window.

Since the most effective and popular service model followed by freemium service is credit-based services, our analysis will be based on data taken from such services.

3.3 Why Twitter?

Even though blackmarket services operate on several social media platforms, we chose Twitter for several reasons. Unlike Facebook and other social media platforms that create exclusive communities, Twitter is dynamic and an open platform. Most of the tweets can be viewed by others, most of the profiles are public, and almost all of the blackmarket services offer a variety of services such as retweets, likes, follows, and so on, that provide a very diverse scenario to study. Also, data collection from Twitter is less cumbersome as compared to other platforms.

4 DATASET DESCRIPTION

We conduct our experiments on a dataset composed of collusive users collected from the blackmarket services, as well as genuine users.

4.1 Dataset Collection

Collecting blackmarket customers: For the purpose of collecting data for blackmarket customers, we restricted ourselves to credit-based services. We did this for the following reasons: (i) the simple model of credit-based services makes them easier to understand and hence makes it easier to collect features. The idea is simple: To get more retweets, a customer would have to retweet tweets of other customers, thus intermittently displaying explicit anomalous retweeting behavior mixed in with normal behavior. (ii) Credit-based services are more prevalent on most of the blackmarket sites as opposed to other types of services. This is because for availing these services, customers do not need to pay the service providers any money. All that is required is simple manual labor of retweeting the tweets of other customers. Thus, these services are more prevalent and highly popular on various blackmarket service providers. Thus, focusing on credit-based services allowed us to obtain a larger dataset of blackmarket customers for analysis. In our previous work [21], we collected an initial dataset, detailed in the upper half of Table 2. To collect this

[13] https://traffup.net/.

Table 2. Statistics of the Initial and Extended Datasets

	User Type	# initial users	# users suspended/deleted	# verified users	# users taken for analysis
Initial Dataset [21]	Collusive Users	1,102	349	31	753
	Genuine Users	1,000	0	51	1,000
Extended Dataset	Collusive Users	1,854	47	72	1,807
	Genuine Users	2,721	12	550	2,706

dataset, we adopted the "active probing" strategy. Using multiple dummy accounts, we enrolled ourselves into these blackmarket services (after careful IRB approval). Then we kept retweeting tweets that were displayed to us. Every time a tweet on the service was retweeted, the Twitter User ID of the creator of the tweet and the tweet ID of the retweeted tweet were recorded. These became our dataset of blackmarket customers. For our initial work [21], we collected data in this fashion over a period of ~3 months. The upper half of Table 2 summarizes the statistics of this dataset. For this study, we continued the same process of data collection over a period of another ~6 months. The lower half of Table 2 shows the statistics of the extended dataset that we obtained over this period.

Collecting genuine users: First, we looked into the users whose tweets had been retweeted by the collusive users that we had extracted. From these users, we extracted users that had been marked as verified users by Twitter. To create our genuine user set, we took the followees of these verified users with the assumption that verified users are more likely to follow genuine users. Then, we removed users with more than 100K followers, since high follower count may resemble a celebrity, which we wanted to discard from our analysis to avoid any unnecessary bias. Further, we removed users with fewer than 50 followers or tweets, as these users do not contribute much to our large-scale analysis. Finally, the human annotators (who also annotated the collusive users as discussed in Section 4.2) were also asked to verify whether these users seem genuine by examining their profile and timeline information. We use this genuine user set for our experiments.

Further, we collected the profile-centric information and the timeline of each user in our dataset using the Tweepy[14] library. Table 2 shows the statistics of the dataset. In this work, we conducted all our experiments on the larger (extended) dataset of collusive and genuine users. As mentioned before, this extended dataset was collected in the same manner as our previous work [21], but over a longer period of time (six months). In our extended dataset, we collected 1,854 collusive users and 2,721 genuine users. Out of these users, 47 collusive users and 12 genuine users turned out to have suspended/deleted accounts. We removed these users from their respective user set. We also found three genuine users in our collusive user set, which we removed from our genuine set. Strangely, we found that ~4% collusive users have been marked as verified users by Twitter, which clearly shows that even Twitter is not yet efficient in detecting collusive users. All the experiments in this study have been conducted on this new, extended dataset.

4.2 Human Annotation of Collusive Users

We asked three human annotators[15] to divide the collusive users by labeling them as *Bots*, *Promotional collusive users,* or *Normal collusive users*. Annotators were properly given the definition of each type of collusive user along with Twitter's terms of service. Annotators were also given complete freedom to search for any information associated with the collusive users and apply their

[14] http://www.tweepy.org/.

[15] Annotators were experts in OSNs with age range between 25 and 35.

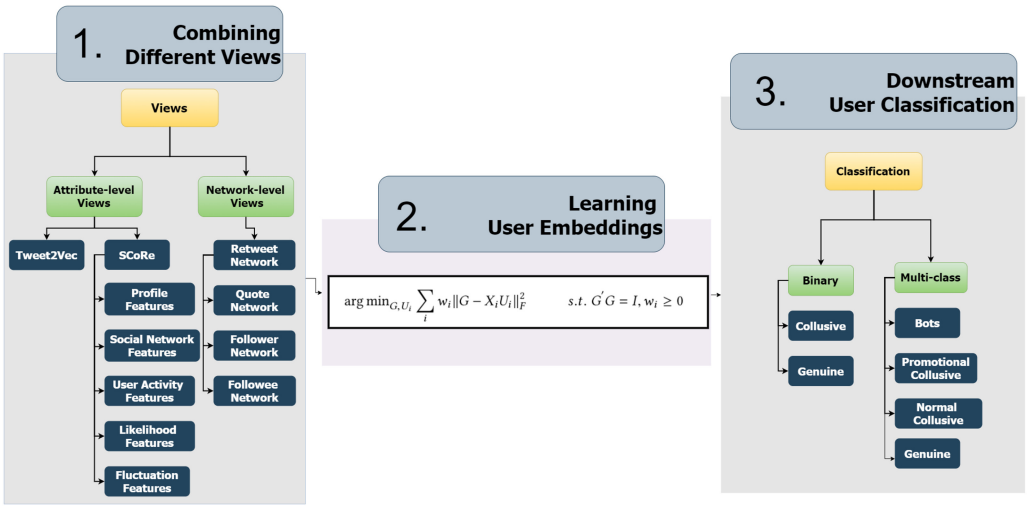


Fig. 1. Brief overview of our methodology—divided into different steps for ease of understanding and reproducibility.

own intuition. The guidelines given to the annotators to label a user into any of the three types of collusive users are as follows:

(1) **Bots:** A Twitter bot is a software that is used to access Twitter using the Twitter API. It has the capability to perform all kinds of actions such as retweeting, following, liking, and so on. The rules that should be followed by a Twitter Bot are mentioned in Twitter Automation rules.[16]
(2) **Promotional collusive users:** Users/organizations who join Twitter for promotions, big product launches, and so on, would want to explore some shortcuts to gain reputation. These promotional collusive users normally use some specific set of keywords such as "win," "ad," "giveaway," "free," "boost," and so on.
(3) **Normal collusive users:** We consider all the collusive users who do not fall under the above two categories as normal collusive users.

We found high inter-annotator agreement (Fleiss' $\kappa = 0.76$). We considered the final label for a collusive user to be the one that is marked the same by at least two annotators. For the users whose labels are not agreed upon by at least two annotators, we labeled them as normal collusive users.

5 METHODOLOGY

Figure 1 shows a depiction of our methodology. Six different user representations are taken as distinct views, which are combined into a single user embedding in an unsupervised manner using WGCCA. Finally, we use the user embedding to classify users in a supervised manner.

5.1 Creating Views for Twitter Users

Multiple studies have implemented various design alternatives for user modeling strategies. Abel et al. [1] used semantic entity and topic-based user modeling strategies of Twitter users and showed advantages over hashtag-based user modeling strategies. Xu et al. [55] modeled the user posting

[16] https://help.twitter.com/en/rules-and-policies/twitter-automation.

behavior on social media. They proposed a mixture latent topic model to combine multiple behavioral properties of a user. Ritter et al. [43] proposed an unsupervised approach to model conversations of Twitter users. However, some recent papers model users from a different aspect of user social participation. Benton et al. [5] considered a multiview approach by learning vector representations of social media users from multiple views created by capturing information from users' online activities. They also proposed an approach using WGCCA to combine multiple views to generate a single embedding for a user. Ding et al. [19] used multiview learning to combine heterogeneous information (likes, status updates) for representation of a user.

In this article, we use views corresponding to the profile attributes and the network of a user and generate a combined representation based on these views. We have multiple sources of deriving actionable information in our Twitter dataset: the content (tweets) posted by a user, attribute features such as number of followers, number of tweets per day, number of retweets per tweet, and so on, and the user's network of interactions, as indicated by their followers, followees, retweets, and quotes. We use these sources of information to generate six different views, which are divided into the following two categories:

5.1.1 Attribute-level Views. We use the content posted by users and their attributes to get the attribute-level views.

1. `Tweet2Vec` *(AV_1)*: We use the `Tweet2Vec` model [17] to find vector-space representations of all the tweets of a user. `Tweet2Vec` is a character-level encoder for social media posts trained using the associated hashtags. It considers the assumption that posts with the same hashtags should have similar embeddings. It uses a bi-directional Gated Recurrent Unit (Bi-GRU) for learning the tweet representations. To get embeddings for a particular tweet, the model combines the final GRU states by going through a forward and backward pass over the entire sequence. We use the pre-trained model provided, trained on a dataset of 2M tweets, to get the tweet embeddings. In our case, we take the mean of embeddings of all the tweets of a user to get a representation of the type of content the user tweets about.

2. `ScoRe` *(AV_2)*: We use various user attributes to create a feature vector. We use a set of 64 features proposed by Reference [21], and used further in Reference [20], that can help distinguish the different types of users. These features can be divided into five categories:

- *Profile Features (PF):* We first incorporate the profile features of a Twitter account. Here, our hypothesis is that an account with higher age tends to retweet more as compared to newly created accounts. We consider the following profile features:
  - (PF_1) *Account age* : Time (in seconds) that has elapsed since the creation of the Twitter account till the date when the data was collected.
  - (PF_2) *Screen name length*: String length of the name displayed on the public profile of the user on Twitter.
  - (PF_3) *Profile description presence*: Binary value that indicates whether the user has a description in his/her profile or not.
  - (PF_4) *Profile description length*: String length of the profile description of the user. It is set to 0 if profile description is not present.
  - (PF_5) *Profile URL presence*: Twitter allows users to add a URL to the profile separately to display publicly. This is a binary value that indicates whether the profile URL is present (1) or not (0).
- *Social Network Features (SNF):* We incorporate these features to consider the connectivity between users. Users involved in collusive retweeting activities follow a lot of other users as compared to the genuine users. Such tendency of these users is due to the reason that

this may draw attention of other users to their profiles and eventually may follow them. We consider the following social network related features:

—(SNF_1) *Followees count*: Count of the number of users that a user follows on Twitter.
—(SNF_2) *Followers count*: Count of the number of followers that the user has on Twitter.
—(SNF_3) *Followees to followers ratio*: Ratio of SNF_1 to SNF_2.

- *User Activity Features (UAF):* We hypothesize that highly active users have a higher chance of getting their tweet retweeted by a stranger. We consider the following features to validate this hypothesis:
  —(UAF_1) *Total number of tweets*: Total number of tweets that have been authored by the user since the time of account creation till the time of data collection.
  —(UAF_2) *Number of direct mentions per tweet*: Average number of mentions per tweet.
  —(UAF_3) *Number of URLs per tweet*: Average number of URLs mentioned by the author per tweet.
  —(UAF_4) *Number of hashtags per tweet*: Average number of hashtags per tweet.
  —(UAF_5) *Number of tweets per day*: Count of the number of users that a user follows on Twitter.
  —(UAF_6) *Number of retweets per day*: Average number of tweets that a user retweets in a day.
  —(UAF_7) *Number of retweets per tweet*: Average number of retweets that a user has received on one tweet.
  —(UAF_8) *Bot-score*: We also consider `BotoM` score developed by Davis et al. [15] as one of the *UAF* features, which gives us an indication whether an account is operated by human or machine.
- *Likelihood Features (LF):* Collusive users demonstrate a mix of organic and inorganic behavior. When they submit tweets to blackmarket services, they start aggressively retweeting others' tweets on these services to gain credits. However, they do not follow such aggressiveness when they publish a tweet not submitted to any of the blackmarket services. We capture this behavior through the following features:
  —($LF_{1\text{-}7}$) *Tweeting likelihood per day for seven days* (*Monday–Sunday*): Ratio of the tweets of a user per day to the total number of tweets the user posted in a week.
  —($LF_{8\text{-}14}$) *Retweeting likelihood per day for seven days* (*Monday–Sunday*): Ratio of the retweets of a user per day to the total number of retweets the user performed in a day of a week.
  —($LF_{15\text{-}21}$) *Regularity of tweeting activity per day for seven days* (*Monday–Sunday*): Regularity of tweeting activity per day is calculated by entropy, $-\sum_{i=1}^{24} p(x_i) \log p(x_i)$, where $p(x_i)$ is the fraction of tweets posted by the user at ith hour of that day.
  —($LF_{22\text{-}28}$) *Regularity of retweeting activity per day for seven days* (*Monday–Sunday*): Regularity of retweeting activity per day is calculated by entropy, $-\sum_{i=1}^{24} p(x_i) \log p(x_i)$, where $p(x_i)$ is the fraction of retweets posted by the user at ith hour of that day.
  —(LF_{29}) *Tweet steadiness*: Tweet steadiness is defined as $1/\sigma_t$, where σ_t is the standard deviation of time difference between consecutive user-generated tweets.
  —(LF_{30}) *Retweet steadiness*: Retweet steadiness is defined as $1/\sigma_{rt}$, where σ_{rt} is the standard deviation of time difference between consecutive user-generated retweets.
  —($LF_{31\text{-}37}$) *Maximum tweet likelihood per day for seven days* (*Monday–Sunday*): It is the ratio of per-day tweet count of a user to the maximum number of tweets the user posted in a day of a week.

—($LF_{38\text{-}44}$) *Maximum retweet likelihood per day for seven days* (*Monday–Sunday*): It is the ratio of per-day retweet count of a user to the maximum number of retweets the user posted in a day of a week.

- *Fluctuation Features (FF):* Collusive users of credit-based freemium services tend to show erratic behavior while retweeting, as their only goal is to gain credits. We consider the following features to capture this behavior:
  —(FF_1) *Retweet count standard deviation*: It is the standard deviation of retweet counts for all user-generated tweets.
  —(FF_2) *Retweets log-time difference average*: It is the mean of log-time difference between consecutive retweets.
  —(FF_3) *Retweets log-time difference standard deviation*: It is the standard deviation of log-time difference between consecutive retweets.

The importance of each of these features has been studied in our previous work [21].

5.1.2 Network-level Views. A user's network is the network of different types of interactions, such as retweet, follow, reply, and so on, with other users. Each user is considered as a node and a unidirectional edge between two users indicates presence of the interaction of the given type (follow, retweet, etc.) between them, with the edge weight indicating the count of the number of interactions of the given type between them. We encode a user's network in a vector representation by considering the interaction between the users in our dataset. We create the representation by constructing an adjacency matrix of the network and then consider each row that corresponds to a user as a representation of the user's network. This gives us a vector of size n = 4,010 (i.e., total number of users considered here), to which we apply Principal Component Analysis (PCA) to get our final representation of size n_{PCA} = 1K. Applying PCA helps us in capturing the user's network as a dense vector and improves the computational efficiency of WGCCA, used later in Section 5.2 to combine the representations. We construct these network representations with the motivation that similar users may have similar interaction networks. We define multiple types of network views based on the type of interaction between users.

- *Retweet network (NV_1)*: In this network, an edge between two user nodes represents the number of times the first user has retweeted some tweets of the second user.
- *Quote network (NV_2)*: In this network, an edge between two user nodes represents the number of times the first user has quoted some tweets of the second user.
- *Follower network (NV_3)*: In this network, an edge between two user nodes indicates that the second user follows the first user on Twitter.
- *Followee network (NV_4)*: In this network, an edge between two user nodes indicates that the second user is a followee of the first user, i.e., the first user follows the second user on Twitter.

Figure 2 shows the visualization of all the views defined in this section by applying t-SNE [51] on each view vector. Collusive and genuine users are represented by the red and green points, respectively. It can be observed that each view contains some information that can be used to classify users as collusive or genuine, and that if we can effectively combine the information contained in the views, we should be able to get better performance. The importance of the views defined here can be seen in Figure 3, denoted by the red bars that indicate the accuracy of the classifier achieved considering each view individually.

5.2 Learning Multiview User Embeddings

Each of the views described in Section 5.1 contains some information that can be used to identify collusive users. However, using just a single view may lead to missing out on valuable information

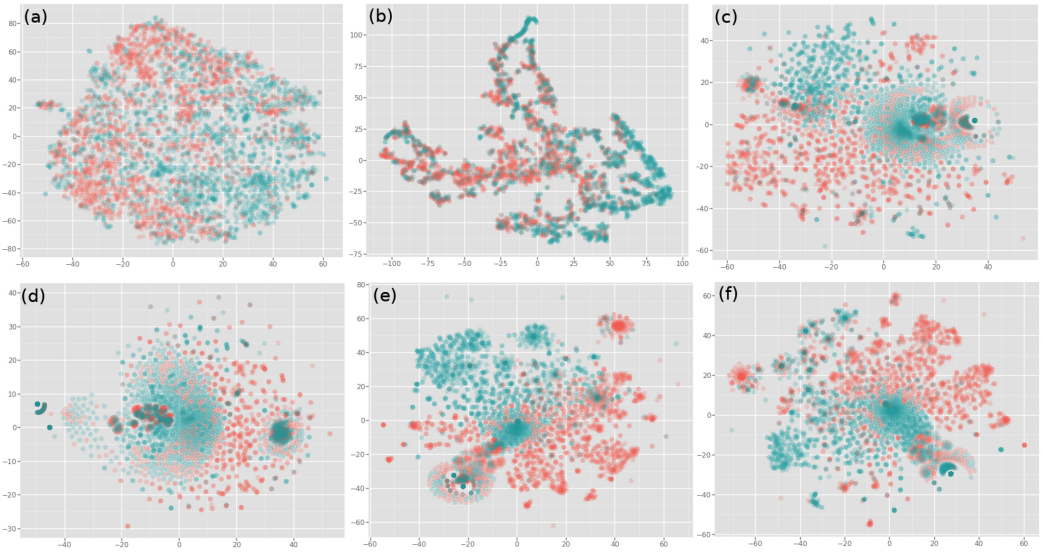


Fig. 2. t-SNE visualization of representations of collusive (red) and genuine (green) users created using (a) Tweet2Vec (AV_1), (b) ScoRe (AV_2), (c) Retweet network (NV_1), (d) Quote network (NV_2), (e) Follower network (NV_3), (f) Followee network (NV_4).

that can help improve the detection performance. A naive approach of doing so would be to simply concatenate the views together. But this would give us a large user embedding and also ignore the complementary nature of information contained in some of the views, particularly in the network views. In this section, we describe an alternate approach to learn a single embedding from multiple views using weighted generalized canonical correlation analysis.

5.2.1 Generalized Canonical Correlation Analysis (GCCA). As detailed by Velden et al. [50], GCCA is a generalization of Canonical Correlation Analysis (CCA). CCA finds linear combination for two sets of variables in such a way that the correlation between them is maximal. GCCA generalizes this to more than two sets of variables. GCCA is used to analyze several sets of variables simultaneously, which makes the method suitable for the analysis of various types of data containing different attributes. Several generalizations of CCA have been proposed [35, 44, 48]. However, we use the one proposed by Carroll [7] for our work, since it has some attractive properties that makes it well-suited for multiple-set data: (a) the method is computationally straightforward, since its solution is based on an eigenequation; (b) it is closely related to well-known multivariate techniques like principal component analysis; and (c) it takes regular CCA as a special case.

The GCCA objective can be expressed as:

$$\arg\min_{G,U_i} \sum_i \|G - X_i U_i\|_F^2 \qquad s.t.\ G^{'}G = I, \tag{1}$$

where $X_i \in \mathbb{R}^{n \times d_i}$ represents the data matrix for the ith view, $U_i \in \mathbb{R}^{d_i \times k}$ maps from the latent space to observed view i, $G \in \mathbb{R}^{n \times k}$ contains the learned user embeddings, G' represents the transpose of the matrix G, and F is the Frobenius (or Euclidean) norm.

5.2.2 Weighted Generalized Canonical Correlation Analysis (WGCCA). As we can observe from the t-SNE visualizations of the different views (cf. Figure 2), each view may be more or less helpful for our task of detecting collusive users. Hence, we weigh each view differently instead of treating each view equally—which is referred to as WGCCA [5]. Given weight w_i for each view i, where

w_i represents the importance of the ith view in determining the user embedding, the learning objective changes to:

$$\arg\min_{G,U_i} \sum_i w_i \|G - X_i U_i\|_F^2 \qquad s.t.\ G^{'}G = I, w_i \geq 0. \tag{2}$$

The columns of G are the eigenvectors of $\sum_i w_i X_i (X_i' X_i)^{-1} X_i'$ and the solution for $U_i = (X_i' X_i) X_i G$. An identity matrix scaled by 10^{-8} for regularization is added to the per-view covariance matrices before inverting, for numerical stability. The modification suggested in Reference [5] to WGCCA—where G is scaled by the square-root of the singular values of the data matrix to improve performance in downstream tasks—is also employed. Note that the GCCA objective remains unsupervised.

6 EXPERIMENTAL SETUP

We choose the same baselines for this study as our previous work [21]. In this study, we also include our previous work as a baseline. From the context of this study, this is equivalent to using the second attribute-level view AV_2 as a baseline. Further, as in our previous work, we use the scores provided by these baselines to train five traditional supervised classifiers—K-Nearest Neighbors (K-NN), Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and three ensemble classifiers—Bagging (BG), Boosting (BO), and Random Forest (RF). We report the results of the classifier that shows the best results on these baselines. We start this section by briefly explaining the baseline methods followed by the details of our experiments.

6.1 Baseline Methods

- **Baseline I:** `BotoM`: Davis et al. [15] proposed Botometer, a system to evaluate social bots on Twitter. It computes a bot-likelihood score for a Twitter user calculated from the account's recent activity. `BotoM` service is also publicly available via REST APIs.[17] We use the bot-likelihood score provided by this baseline to train the classifiers listed above and report the results of the classifier that gives us the best results.
- **Baseline II:** `FakeAcc`: Elazab et al. [23] attempted to detect fake accounts on Twitter using an effective minimum weighted feature set derived from a set of 22 features. They applied Gain measure [34] to find the weight for the attributes selected in the final feature set. The final feature set proposed by them was used to train the set of classifiers mentioned earlier, and the results of the classifier that gave the best precision were reported.
- **Baseline III:** `SpamBot`: We use the method proposed by Wang et al. [53] as our third baseline. It takes into account the graph-based features and content-based features extracted from the user's social graph and most recent tweets, respectively, to distinguish the spam bots from the genuine users. We run their suggested classifier (Naive Bayes) on our dataset for multi-class and binary classification.
- **Baseline IV:** `NDSync`: Giatsoglou et al. [25] proposed `NDSync`, a method to detect synchronous retweet fraudsters by computing a user-level suspiciousness score by combining the suspiciousness score for each retweet thread projected into a multi-dimensional feature space. We use the suspiciousness score returned by `NDSync` as a feature for training the set of classifiers mentioned above. Once again, the results of the classifier that gave the best performance were reported.
- **Baseline V:** `ScoRe`: We also consider our previous work `ScoRe` [21] as our fifth baseline. We reported that SVM is the best state-of-the-art supervised classifier on the set of 64 features.

[17] https://botometer.iuni.iu.edu/.

6.2 Weight Assignment for WGCCA

We tune the weights assigned to the views for both our tasks of multi-class and binary classification by generating user embeddings of different sizes $\in \{50, 100, 200\}$ for all possible weight assignments and choose the one that performs the best. The set of assignments $\mathcal{A}$ is given by: $\mathcal{A} = \mathcal{W}^{|\mathcal{V}|} - \mathcal{A}_0$, where $\mathcal{W} \in \{0, 0.25, 1\}$ is the set of possible weights, $\mathcal{V} = \{AV_1, AV_2, NV_1, NV_2, NV_3, NV_4\}$ is the set of views, and $\mathcal{A}_0$ is the assignment $(0, 0, 0, 0, 0, 0)$, where all the views are assigned a weight of 0.

For each tuple in $\mathcal{A}$, we assign weight w_i to each view (corresponding to the elements of the tuples). We choose the best assignment of weights from the set $\mathcal{A}$ based on the result of the best performing classifier from a set of state-of-the-art classifiers trained on the embeddings generated for each assignment. The best performing assignment of weights is discussed in the following section.

6.3 Alternate View Fusion Methods

Apart from using WGCCA to generate a combined user embedding, we use alternative methods for view fusion (VF) to compare their performance against WGCCA. We concatenate the views together to get a combined user representation of size $n = 4{,}564$, and use two different methods to generate a combined user representation:

- **Neural network (VF_{NN})** Post concatenation, we feed the representation to a neural network with a hidden layer of size $n = 100$ and use it to classify the users.
- **Principal component analysis (VF_{PCA})** Post concatenation, we apply PCA to get a final representation of size $n = 100$. This representation is then used to train the different classifiers listed above, and the results of the classifier with the best performance are reported.

7 RESULTS

We measure the performance of the competing methods using the following metrics: Precision, Recall, F1-score, and Area under the ROC curve (AUC). We report all the above metrics in both micro and macro settings. All the supervised baselines scores are reported after 5-fold cross-validation. Note that the test set remains same across all the methods.

We design the first experiment by considering it as a multi-class (bot, promotional, normal, and genuine) classification problem. We also conduct another experiment by considering it as a binary classification problem, by combining all types of collusive users (bots, promotional, and normal) into a single class (customer). We run our experiments by training the following set of classifiers: Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), Naive Bayes (NB), and K-Nearest Neighbors (K-NN). We also use three ensemble classifiers: Random Forest (RF), Bagging (BG), and Boosting (BO). We further use a neural network–based classifier—Multi-Layer Perceptron (MLP). We perform hyperparameter optimization to get the best results. We report the best performance achieved in terms of the evaluation metrics by sweeping across user embeddings of size $\in \{50, 100, 200\}$, considering all possible weight assignments, and report the embedding size and the classifier for which we obtain the best performance.

Figure 3 shows the best F1-score (macro) obtained by sweeping across output embeddings of size $\in \{50, 100, 200\}$ and all weight assignments detailed in Section 6.2—for both multi-class and binary classification. The figure shows the best performance achieved after taking different number of views at a time (views assigned a weight = 0 are essentially dropped). When considering the views individually ($WGCCA_1$), we observe that an SVM trained on the `ScoRe` view (AV_2) gives us the best result for multi-class classification, with an F1-score (macro) of 0.661; and an SVM trained on the `Tweet2Vec` view (AV_1) gives us the best result for binary classification, with an F1-score (macro) of 0.802.

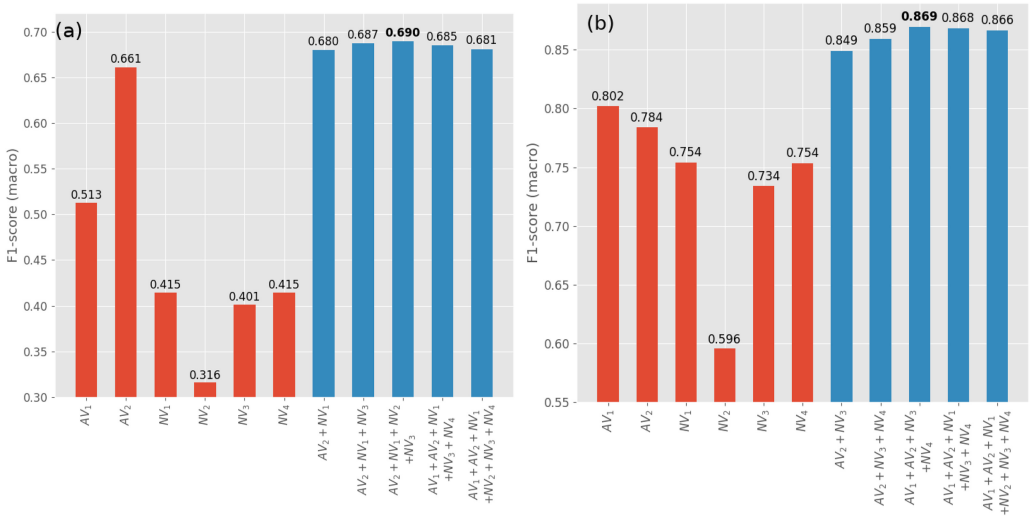


Fig. 3. Best F1-score (macro) obtained across embeddings of size $\in \{50, 100, 200\}$, taking different combinations of views (blue) as well as taking the views individually (red) for: (a) multi-class classification and (b) binary classification.

As opposed to our previous study [21], where we tried to classify collusive users based on the user representation obtained from ScoRe (AV_2), here, we have a combination of different views that can be used for classifying collusive users. This allows us to capture the multi-faceted nature of collusive users. In our previous study, we explored the feature importance of the different features that make up the ScoRe representation (AV_2). Similarly, in this study, we explore the importance of each view in classifying collusive users. To achieve this, we use WGCCA to combine different combinations of views and study how each combination performs in the task of classifying collusive users. When we apply WGCCA to combine the views, we observe the following:

- We find that the best result when taking two views at a time ($\boldsymbol{WGCCA_2}$) is achieved by using ScoRe (AV_2) and Retweet network (NV_1) views for multi-class classification, with an F1-score (macro) of 0.680 (using LR classifier on embeddings of size = 200); and by combining the ScoRe (AV_2) and Follower network (NV_3) views for binary classification, with an F1-score (macro) of 0.849 (using MLP on embeddings of size = 100).
- When we take three views at a time ($\boldsymbol{WGCCA_3}$), the best result is achieved by combining the views ScoRe (AV_2), Retweet network (NV_1), and Follower network (NV_3) for multi-class classification, with an F1-score (macro) of 0.687 (using SVM on embeddings of size = 50); and by combining the views ScoRe (AV_2), Follower network (NV_3), and Followee network (NV_4) for binary classification, with an F1-score (macro) of 0.859 (using MLP on embeddings of size = 50).
- When we take four views at a time ($\boldsymbol{WGCCA_4}$), we achieve the best result by combining the views ScoRe (AV_2), Retweet network (NV_1), Quote network (NV_2), and Follower network (NV_3) for multi-class classification, with an F1-score (macro) of 0.690 (using SVM on embeddings of size = 50)—**which is also the best score achieved overall for multi-class classification**. For binary classification, combining the views Tweet2Vec (AV_1), ScoRe (AV_2), Follower network (NV_3), and Followee network (NV_4) gives us the best result, with an F1-score (macro) of 0.869 (using MLP on embeddings of size = 50)—**which is also the best result achieved overall for binary classification**.

Table 3. Performance of Different Competing Methods for Multi-class Classification

Classifier	Micro				Macro			
	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC
BotoM (RF)	0.546	0.546	0.546	0.638	0.518	0.532	0.519	0.621
SpamBot	0.497	0.497	0.497	0.571	0.389	0.375	0.355	0.434
FakeAcc (LR)	0.562	0.562	0.562	0.635	0.398	0.320	0.298	0.499
NDSync (LR)	0.511	0.511	0.511	0.602	0.375	0.388	0.372	0.443
ScoRe (SVM)	0.637	0.637	0.637	0.893	0.668	0.665	0.660	0.880
VF_{NN}	0.571	0.571	0.571	0.802	0.542	0.540	0.541	0.788
VF_{PCA} (MLP)	0.561	0.561	0.561	0.819	0.521	0.518	0.520	0.804
$WGCCA_1$ (SVM)	0.637	0.637	0.637	**0.893**	0.668	0.665	0.660	**0.880**
$WGCCA_2$ (200, LR)	0.663	0.663	0.663	0.864	0.678	0.689	0.680	0.861
$WGCCA_3$ (50, SVM)	0.687	0.687	0.687	0.848	0.691	0.688	0.687	0.834
$WGCCA_4$ (50, SVM)	0.687	0.687	0.687	0.849	0.693	**0.691**	**0.690**	0.836
$WGCCA_5$ (50, SVM)	0.684	0.684	0.684	0.866	0.687	0.685	0.685	0.856
$WGCCA_6$ (50, SVM)	**0.692**	**0.692**	**0.692**	0.868	**0.696**	0.675	0.681	0.854

Table 4. Performance of Different Competing Methods for Binary Classification

Classifier	Micro				Macro			
	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC
BotoM (RF)	0.683	0.683	0.683	0.780	0.685	0.682	0.680	0.780
SpamBot	0.682	0.682	0.682	0.695	0.699	0.661	0.645	0.695
FakeAcc (RF)	0.694	0.694	0.694	0.692	0.694	0.688	0.688	0.692
NDSync (LR)	0.573	0.573	0.573	0.671	0.383	0.293	0.269	0.671
ScoRe (SVM)	0.766	0.766	0.766	0.799	0.771	0.759	0.760	0.799
VF_{NN}	0.781	0.781	0.781	0.779	0.779	0.779	0.779	0.779
VF_{PCA} (MLP)	0.785	0.785	0.785	0.782	0.783	0.782	0.782	0.782
$WGCCA_1$ (SVM)	0.804	0.804	0.804	0.801	0.803	0.802	0.802	0.801
$WGCCA_2$ (100, MLP)	0.851	0.851	0.851	0.849	0.850	0.849	0.849	0.849
$WGCCA_3$ (50, MLP)	0.861	0.861	0.861	0.859	0.860	0.859	0.859	0.859
$WGCCA_4$ (50, MLP)	**0.871**	**0.871**	**0.871**	**0.869**	**0.870**	**0.869**	**0.869**	**0.869**
$WGCCA_5$ (50, MLP)	0.870	0.870	0.870	0.868	0.869	0.868	0.868	0.868
$WGCCA_6$ (50, MLP)	0.868	0.868	0.868	0.866	0.867	0.866	0.866	0.866

- Considering five views at a time ($WGCCA_5$), we achieve the best result by combining the views Tweet2Vec (AV_1), ScoRe (AV_2), Retweet network (NV_1), Follower network (NV_3), and Followee network (NV_4) for both multi-class and binary classification, with an F1-score (macro) of 0.685 in the case of multi-class classification (using SVM on embeddings of size = 50), and 0.868 in the case of binary classification (using MLP on embeddings of size = 50).
- When we consider all six views ($WGCCA_6$), we achieve an F1-score (macro) of 0.681 for multi-class classification (using SVM on embeddings of size = 50) and 0.866 for binary classification (using MLP on embeddings of size = 50).

Tables 3 and 4 show the results for the multi-class classification and binary classification, respectively, in terms of the evaluation metrics. As shown in Table 3, ScoRe (our previous work) turns out to be the best among all the baselines. Our approach of detecting collusive users outperforms

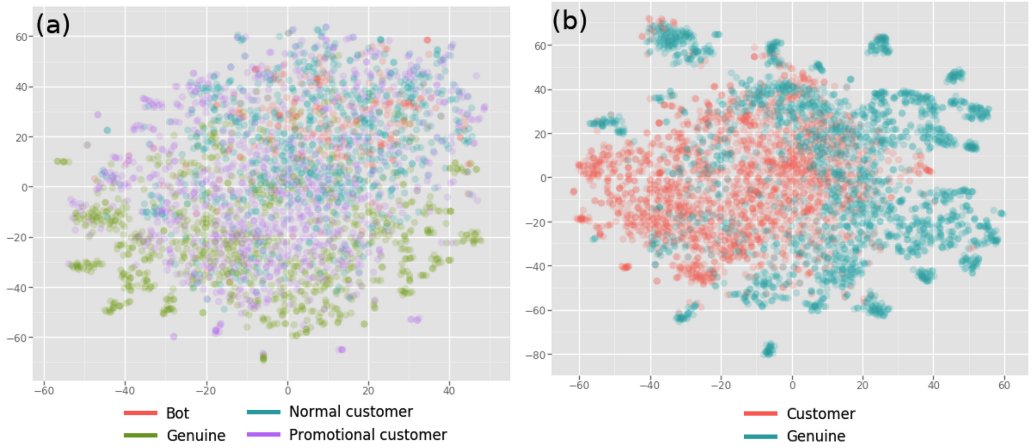


Fig. 4. Qualitative analysis: t-SNE visualization of best performing user embeddings for (a) multi-class classification and (b) binary classification.

all the earlier baselines and ScoRe, with a relative improvement of 32.95% over the best previous baseline (BotoM) and 4.55% over ScoRe, in terms of F1-score (macro). The class-wise F1-score of the best performing classifier (SVM trained on embeddings of size = 50) in detecting each type of user is as follows: 0.74 (bot), 0.57 (promotional), 0.66 (normal), 0.64 (genuine). In case of binary classification, we observe in Table 4 that our approach outperforms the best baseline (FakeAcc) by 26.31% and ScoRe by 14.34%, in terms of F1-score (macro). The class-wise score of our best performing classifier (MLP trained on embeddings of size = 50) in detecting each type in case of binary classification is as follows: 0.85 (collusive) and 0.88 (genuine). Our approach of using WGCCA to generate a combined user embedding from the views also outperforms the alternate view fusion techniques—VF_{NN} and VF_{PCA}.

Figure 4 shows the t-SNE visualizations of user embedding that gives us the best result for both multi-class and binary classification. The best result for multi-class classification is obtained when we use the weights: $(0, 1, 1, 0.25, 1, 0)$, corresponding to the views $(AV_1, AV_2, NV_1, NV_2, NV_3, NV_4)$, and generate combined user embeddings of size = 50, followed by training an SVM on the embeddings. The best result for binary classification is obtained when we use the weights: $(0.25, 0.25, 0, 0, 0.25, 0.25)$, corresponding to the same views, and generate combined user embeddings of size = 50, followed by training an MLP on the embeddings. We can see a separation of different types of users for both types of classifications in the figure. The visualization provides a qualitative way of evaluating our embedding method.

Further, Figure 5 shows the variation in F1-score (macro) when taking different ratios of collusive users to genuine users by training an SVM on the embedding that produced the best result for the entire dataset. This shows the robustness of our approach on skewed data (when the size of classes is imbalanced).

8 CASE STUDY

We have quantitatively established that detection and representation of collusive users require the introduction of a new methodology. We now study how effective is our method in distinguishing between collusive and genuine users.

Figure 6 shows screenshots of tweets made by three collusive users and a genuine user. At first glance, all of these tweets seem to be made by genuine users for a variety of reasons:

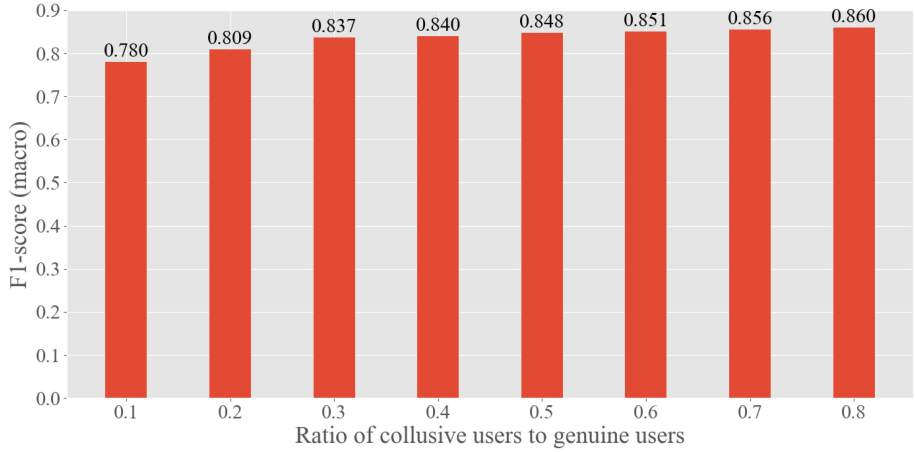
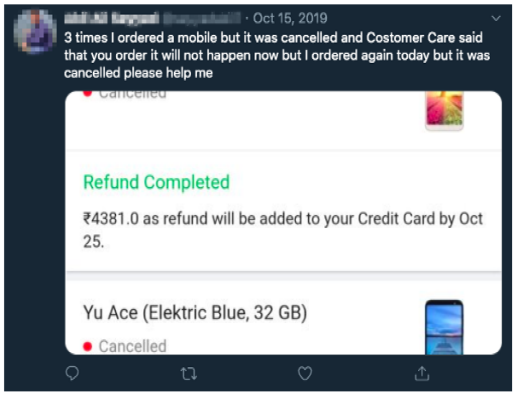


Fig. 5. Variation of F1-score (macro) for different ratios of collusive users to genuine users.

(a) Example tweet with a picture and a few hashtags similar to a genuine tweet.



(b) Example tweet with a picture along with a complaint regarding some online service.

(c) Example tweet about an opinion made about bitcoins based on a given link.

(d) Example tweet made by a genuine user, which looks like a promotional tweet.

Fig. 6. Case study—Screenshots of tweets posted by collusive users ((a)–(c)) and a genuine user (d).

- Figure 6(a) is a tweet with a picture and a few hashtags. This seems like a genuine post; however, while analyzing other tweets of this user, one can find out that they actively retweet blackmarket content (and hence have many blackmarket users in their retweet network). Thus, our multi-view approach is able to capture such behavior and flag the user as collusive, which the ground-truth annotation also supports.

- Figure 6(b) is a tweet, again with a picture, along with a complaint regarding some online service. Once again, this behavior is extremely common by genuine users on Twitter, as they often resort to complaint resolution with the help of public support. However, this user has sent multiple of their tweets to the blackmarket service, making them an active user of the blackmarket services. They were flagged as collusive by our system.
- Figure 6(c) is a tweet about an opinion made about bitcoins based on a given link. None of the hashtags seem to be suspicious, and the use of textual enhancements such as emojis and punctuation is also in order. However, according to our ground-truth dataset, this user is an active participant of the blackmarket services and is also detected as one by our system. This is because this user actively makes use of the blackmarket services to further spread his or her own opinions and also retweets content on blackmarket services on the topic of *blockchains*.
- Figure 6(d) is a tweet made by a genuine user. This is a promotional tweet, which gives an initial impression that it might be collusive; although their network does not consist of collusive users and their retweet patterns do not reflect any blackmarket engagement as well. They are flagged by the system as genuine.

As we can see, collusive users can show stark similarities with genuine users. However, there are multiple factors that can mark them as collusive. These factors are mapped in our system, owing to the multi-view approach, which helps us effectively identify collusive and genuine users.

9 CONCLUSION AND FUTURE WORK

In this article, we proposed several representations of collusive Twitter users using their attribute- and network-level features. We then used a multi-view learning-based approach (WGCCA) to combine these representations. We noticed that when we combine different user representations into a single user embedding, we are able to capture different aspects of a user's behavior on Twitter in a better way, as compared to considering the views individually. We showed the effectiveness of our approach over the baselines. The dataset we collected is also the first dataset of its kind. The dataset is composed of three types of collusive users: bots, promotional collusive users, and normal collusive users, manually annotated by human annotators.

So far, we have explored the extrinsic collusive properties of users, which are analyzed directly from the multiview approach. In the future, we wish to explore intrinsic properties of users and how these properties propagate to influence other users in their networks. These intrinsic properties are inherent traits in users and tweets that define their collusive behavior, whereas extrinsic properties are their interactions with their networks. We also wish to explore the inter-dependency of such collusive users with the tweets that are a part of this network. Moreover, we would like to create a content-level classification to study which types of tweets are submitted to blackmarket services and how these tweets are propagated. The other important direction would be to study the information diffusion of these tweets. We also plan to expand our dataset of collusive users to gain deeper knowledge about the behavioral traits of these users. The codes and datasets used in this article are available at https://github.com/uditarora/collusive-retweeters-TIST-2020.

REFERENCES

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. Analyzing user modeling on Twitter for personalized news recommendations. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer, 1–12.

[2] Anupama Aggarwal and Ponnurangam Kumaraguru. 2014. Followers or phantoms? An anatomy of purchased Twitter followers. *arXiv preprint arXiv:1408.1534* (2014).

[3] Udit Arora, William Scott Paka, and Tanmoy Chakraborty. 2019. Multitask learning for black-market tweet detection. *arXiv preprint arXiv:1907.04072* (2019).
[4] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on Twitter. In *Proceedings of the Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS'10)*, Vol. 6. 12.
[5] Adrian Benton, Raman Arora, and Mark Dredze. 2016. Learning multiview embeddings of Twitter users. In *Proceedings of the 54th Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 14–19.
[6] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. Copycatch: Stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 119–130.
[7] J. Douglas Carroll. 1968. Generalization of canonical correlation analysis to three or more sets of variables. In *Proceedings of the 76th Convention of the American Psychological Association*, Vol. 3. 227–228.
[8] Jacopo Castellini, Valentina Poggioni, and Giulia Sorbi. 2017. Fake Twitter followers detection by denoising autoencoder. In *Proceedings of the International Conference on Web Intelligence*. ACM, 195–202.
[9] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. DeBot: Twitter bot detection via warped correlation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'16)*. 817–822.
[10] Le Chen, Alan Mislove, and Christo Wilson. 2016. An empirical analysis of algorithmic pricing on Amazon Marketplace. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1339–1349.
[11] Liang Chen, Yipeng Zhou, and Dah Ming Chiu. 2015. Analysis and detection of fake views in online video services. *ACM Trans. Multim. Comput. Commun. Applic.* 11, 2s (2015), 44.
[12] Aditya Chetan, Brihi Joshi, Hridoy Sankar Dutta, and Tanmoy Chakraborty. 2019. CoReRank: Ranking to detect users involved in black-market-based collusive retweeting activities. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. ACM, 330–338.
[13] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is tweeting on Twitter: Human, bot, or cyborg? In *Proceedings of the 26th Computer Security Applications Conference*. ACM, 21–30.
[14] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: Efficient detection of fake Twitter followers. *Dec. Supp. Syst.* 80 (2015), 56–71.
[15] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 273–274.
[16] Sarthika Dhawan, Siva Charan Reddy Gangireddy, Shiv Kumar, and Tanmoy Chakraborty. 2019. Spotting collusive behaviour of online fraud groups in customer reviews. *arXiv preprint arXiv:1905.13649* (2019).
[17] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2Vec: Character-based distributed representations for social media. In *Proceedings of the 54th Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 269–274.
[18] John P. Dickerson, Vadim Kagan, and V. S. Subrahmanian. 2014. Using sentiment to detect bots on Twitter: Are humans more opinionated than bots? In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE Press, 620–627.
[19] Tao Ding, Warren K. Bickel, and Shimei Pan. 2017. Multi-view unsupervised user feature embedding for social media-based substance use prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2275–2284.
[20] H. S. Dutta and T. Chakraborty. 2020. Black-market-driven collusion among retweeters—Analysis, detection, and characterization. *IEEE Trans. Inf. Forens. Sec.* 15 (2020), 1935–1944. DOI: https://doi.org/10.1109/TIFS.2019.2953331
[21] Hridoy Sankar Dutta, Aditya Chetan, Brihi Joshi, and Tanmoy Chakraborty. 2018. Retweet us, we will retweet you: Spotting collusive retweeters involved in black-market services. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'18)*. 242–249.
[22] H. S. Dutta, V. R. Dutta, A. Adhikary, and T. Chakraborty. 2020. HawkesEye: Detecting fake retweeters using Hawkes process and topic modeling. *IEEE Trans. Inf. Forens. Sec.* (Jan. 30, 2020). DOI: https://doi.org/10.1109/TIFS.2020.2970601
[23] Ahmed El Azab. 2016. Fake accounts detection in Twitter based on minimum weighted feature. *International Scholarly and Scientific Research and Innovation* 10, 1 (2016), 13–18.
[24] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia, and Alok N. Choudhary. 2012. Towards online spam filtering in social networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'12)*, Vol. 12. 1–16.
[25] Maria Giatsoglou, Despoina Chatzakou, Neil Shah, Alex Beutel, Christos Faloutsos, and Athena Vakali. 2015. Ndsync: Detecting synchronized fraud activities. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 201–214.
[26] Maria Giatsoglou, Despoina Chatzakou, Neil Shah, Christos Faloutsos, and Athena Vakali. 2015. Retweeting activity on Twitter: Signs of deception. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 122–134.

[27] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. 2010. @ spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*. ACM, 27–37.

[28] Srishti Gupta, Abhinav Khattar, Arpit Gogia, Ponnurangam Kumaraguru, and Tanmoy Chakraborty. 2018. Collective classification of spam campaigners on Twitter: A hierarchical meta-path based approach. *arXiv preprint arXiv:1802.04168* (2018).

[29] Sonu Gupta, Ponnurangam Kumaraguru, and Tanmoy Chakraborty. 2019. Malreg: Detecting and analyzing malicious retweeter groups. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. ACM, 61–69.

[30] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in online freelance marketplaces: Evidence from TaskRabbit and Fiverr. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'17)*. 1914–1933.

[31] Xia Hu, Jiliang Tang, Yanchao Zhang, and Huan Liu. 2013. Social spammer detection in microblogging. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'13)*, Vol. 13. 2633–2639.

[32] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2014. Catchsync: Catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 941–950.

[33] Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th International Conference on World Wide Web*. ACM, 1189–1190.

[34] Asha Gowda Karegowda, A. S. Manjunath, and M. A. Jayaram. 2010. Comparative study of attribute selection using gain ratio and correlation based feature selection. *Int. J. Inf. Technol. Knowl. Manag.* 2, 2 (2010), 271–277.

[35] Jon R. Kettenring. 1971. Canonical analysis of several sets of variables. *Biometrika* 58, 3 (1971), 433–451.

[36] Srijan Kumar, Justin Cheng, Jure Leskovec, and V. S. Subrahmanian. 2017. An army of me: Sockpuppets in online discussion communities. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 857–866.

[37] Eric Lancaster, Tanmoy Chakraborty, and V. S. Subrahmanian. 2018. MALTP: Parallel prediction of malicious tweets. *IEEE Trans. Computat. Soc. Syst.* 5, 4 (2018), 1096–1108.

[38] Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering social spammers: Social honeypots+ machine learning. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 435–442.

[39] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Learning to identify review spam. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 22. 2488.

[40] Yuli Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2016. Pay me and I'll follow you: Detection of crowdturfing following activities in microblog environment. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'16)*. 3789–3796.

[41] Ashish Mehrotra, Mallidi Sarreddy, and Sanjay Singh. 2016. Detection of fake Twitter followers using graph centrality measures. In *Proceedings of the 2nd International Conference on Contemporary Computing and Informatics (IC3I'16)*. IEEE, 499–504.

[42] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. 2011. An analysis of underground forums. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement Conference*. ACM, 71–80.

[43] Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 172–180.

[44] Peter M. Robinson. 1973. Generalized canonical analysis for time series. *J. Multivar. Anal.* 3, 2 (1973), 141–160.

[45] Neil Shah, Hemank Lamba, Alex Beutel, and Christos Faloutsos. 2017. The many faces of link fraud. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'17)*. IEEE, 1069–1074.

[46] Thamar Solorio, Ragib Hasan, and Mainul Mizan. 2013. Sockpuppet detection in Wikipedia: A corpus of real-world deceptive writing for linking identities. *arXiv preprint arXiv:1310.6772* (2013).

[47] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. 2013. Follow the green: Growth and dynamics in Twitter follower markets. In *Proceedings of the Conference on Internet Measurement Conference*. ACM, 163–176.

[48] Arthur Tenenhaus and Michel Tenenhaus. 2011. Regularized generalized canonical correlation analysis. *Psychometrika* 76, 2 (2011), 257.

[49] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. 2013. Trafficking fraudulent accounts: The role of the underground market in Twitter spam and abuse. In *Proceedings of the USENIX Security Symposium*. 195–210.

[50] Michel van de Velden. 2011. On generalized canonical correlation analysis. In *Proceedings of the 58th World Statistical Congress*.

[51] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (2008), 2579–2605.

[52] Sokratis Vidros, Constantinos Kolias, Georgios Kambourakis, and Leman Akoglu. 2017. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Fut. Internet* 9, 1 (2017), 6.

[53] Alex Hai Wang. 2010. Detecting spam bots in online social networking sites: A machine learning approach. In *Proceedings of the IFIP Conference on Data and Applications Security and Privacy*. Springer, 335–342.

[54] De Wang, Shamkant B. Navathe, Ling Liu, Danesh Irani, Acar Tamersoy, and Calton Pu. 2013. Click traffic analysis of short URL spam on Twitter. In *Proceedings of the 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom'13)*. IEEE, 250–259.

[55] Zhiheng Xu, Yang Zhang, Yao Wu, and Qing Yang. 2012. Modeling user posting behavior on social media. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 545–554.

[56] Xueling Zheng, Yiu Ming Lai, Kam-Pui Chow, Lucas C. K. Hui, and Siu-Ming Yiu. 2011. Sockpuppet detection in online discussion forums. In *Proceedings of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'11)*. IEEE, 374–377.

Received July 2019; revised January 2020; accepted January 2020