CSE472 Project Phase 2

Twitter Analysis – Fake Follower Detection

Group members:

Tyler Black Ujun Jeong

tqblack@asu.edu ujeong1@asu.edu

**Abstract:**

The goal of this project is to detect fake followers in a selected subset of Twitter data. Fake followers are Twitter accounts specifically created to inflate the number of followers of a target account. These fake followers artificially inflate the popularity and influence of an account within the Twittersphere by taking advantage of the herd behavior phenomenon. In this report, we show how we can distinguish fake followers by implementing a feature extraction approach with a semi-supervised learning method. In the experiment, we used about 3,000 labeled data with user information and crawled around 800 entries of unlabeled data from Twitter through the Tweepy API. We extracted 16 important features to detect fake followers. As a result, the experiment achieved 96% detection accuracy when using a Random Forest method and achieved 99% accuracy when using semi-supervised learning method with unlabeled data.

**Keywords:** Semi-supervised learning, feature extraction, twitter analysis, fake bot detection,

## Introduction

When detecting fake followers on social media, initial data is unlabeled because the provenance of users is not yet determined. The only way to establish if a user is a bot or not with complete accuracy is to manually annotate the account. In order to respond to the latest advances in fake followers and their behavior, we must train models with labeled data to deal with unlabeled data. Therefore, we propose a semi-supervised learning method which utilizes both labeled and unlabeled data. We also leverage feature selection to attain greater levels of accuracy in our bot detection efforts.

## Related literature

The traditional detection method is rule-based detection, which determines the “degree of fakeness” of fake followers based on crafted judgements. “FakeFollowerCheck tool” by Socialbakers is one example of this research, but it often requires help from a human when new type of fake follower is detected. To overcome this problem, many researchers have tried to use supervised learning methods such as k-NN to detect fake followers. For examples, “Fame for Sale” research attempts to detect bots at the account level, by processing many social media posts to extract the features. This research contributed to the knowledge of the most important features to detect fake followers. The lag does affect our understanding, because while we do understand past features, we need annotated accounts to understand any new features, which takes time. For this reason, many researches have tried to use unsupervised learning methods to attain bot evolution agnostic detection methods that don’t rely on features and annotated data. This is difficult because fake follower developers do their best to camouflage their accounts to thwart detection. Indeed, many real users coexist with fake users and interact without knowledge of the deception. For this reason, unsupervised learning with graph theory is often limited to detecting the characteristics of the fake users by itself. Therefore, it is important to analyze fake users based on their features but also confirm labels in relation to other data. Finally, this approach requires leveraging graph-based semi-supervised learning [1].

## Methodology

We attempted several supervised learning methods to define baseline of our model: 1) k nearest neighbor (k-NN) 2) Logistic regression 3) Multi-Layer Perceptron (MLP), and 4) Random Forest. These different methods were adopted to showcase the accuracy while increasing the number of features to find fake followers. This approach turned out to be effective even though we used relatively few features with these supervised methods.

In total, we used 16 different features which were chosen because they were proven to be effective in existing research [2]. It includes features such as “friend/follower ratio” or “bot in username”. We’ve found that friends/follower ratio is the most important feature for detecting fake users through our experiments. However, other features like “retweet ratio” or “number of friends” also played an important role in that it shows some features of the user in the view of network.

The highest performing model was Random Forest because it is based on ensemble model that can reflect the importance of each feature without having a single feature dominate the performance results. We’ve shown that detecting labeled users in the selected dataset is an achievable task through feature engineering.

To increase the scope of the problem, we also added unlabeled samples which comprise about 30% of original data. The unlabeled samples helped to increase the overall performance of our experiment and add weight to our predictions by showing that we didn’t simply overfit to the dataset. The key idea is to leverage the joint loss of supervised learning and unsupervised learning by a form of consistency loss. This loss function counts weights to each loss from supervised and unsupervised learning and minimizes its values while training. Finally, unlabeled samples help to clarify the distinction between fake followers and real users by analyzing how each group (cluster) behaves. We also added some network features like in-degree (followers) and out-degree (following) as well as the trend of propagation (retweets ratio), in order to better understand the characteristics of each cluster.

## Dataset

To generalize our dataset, we combined different Twitter dataset with labeled data. There are 2 datasets used in this report: 1) vendor-purchased-2019 2) Fame for sale. These datasets are designed to detect users who try to manipulate the popularity of a certain merchandise in twitter. Therefore, it can reveal characteristics of bots designed for these purposes.

|  |  |  |
| --- | --- | --- |
| Label | Hum | Bot |
| Vendor purchased | N/A | 1,088 |
| Fame for sale | 1,950 | N/A |

<Fig : distribution of each labled dataset>

## Experiment

The results when using only labeled data with a feature (friend/follower ratio) are given in the following table:

|  |  |
| --- | --- |
| Model | Accuracy |
| k-NN | 83.3% |
| MLP | 84.1% |
| Logistic Regression | 81.7% |
| Random Forest | 84.1% |

<Fig: Supervised models with 1 feature >

The results when using labeled data with all 16 features (retweet ratio, followers, etc.) are given in the next table:

|  |  |
| --- | --- |
| Model | Accuracy |
| k-NN | 89.4% |
| MLP | 90.9% |
| Logistic Regression | 91.8% |
| Random Forest | 95.7% |

<Fig: Supervised models with 16 features>

We crawled user information from Twitter to collect the unlabeled dataset. Here, the most recent users collected in 2020 are used to reflect the evolution of fake users. This data consists of 830 users and we used Tweepy API to look up detailed information of each user. This data was subject to the same feature engineering as the other data to use semi-supervised learning in our report.

Finally, we’ve found that by using both labeled and unlabeled data with semi-supervised learning method we can achieve a 99.6% rate of accuracy in identifying bots in the dataset. This shows that semi-supervised learning is effective when detecting new users in social media even though their label is not yet annotated by human. This shows that our model has a potential to be extended to detect bots with different purposes beyond herd behavior manipulation.

## Observation and Discussion

We can see that the methods used in this research were successful with this dataset. As noted, feature engineering can be a useful tool for discovering simple bots. Semi-Supervised Learning also brought us a high degree of success. However, this success is not guaranteed to work in the future. Further research lies in the area of bot detection when features evolve to circumvent detection algorithms. In “A Decade of Social Bot Detection” [8], the issue of evolving features is highlighted. The potential solution that is given is the “group approach” method, which emphasizes unlabeled data being used to train bot detection models. Further work and research are needed in this area to expand the field and achieve greater levels of success. Because while our method is highly effective with our current dataset, it is not within the realm of certainty that our methods would be successful with a randomly selected dataset from Twitter that could be instantly collected.

## Deliverables

**Features used:**

1) friends/(follower\*follower) ratio

2) “bot” in biography

3) “bot” in name

4) Profile has name

5) Profile has image

6) Profile has address

7) Profile has biography

8) Followers greater than 30

9) Belongs to a list

10) URL in profile

11) Followers greater than 50

12) Default image after 2 months

13) Friends greater than 100

14) No contents in biography

15) No location

16) Number of friends

## References

[1] Balaanand, Muthu, et al. "An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter." The Journal of Supercomputing 75.9 (2019): 6085-6105.

[2] Yang, Chao, Robert Harkreader, and Guofei Gu. "Empirical evaluation and new design for fighting evolving twitter spammers." IEEE Transactions on Information Forensics and Security 8.8 (2013): 1280-1293.

[3] Breiman, L. (2001). Random forests. Machine learning, 45(1):5–32.

[4] Iscen, Ahmet, et al. "Label propagation for deep semi-supervised learning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2019.

[5] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In ECCV, 2018

[6] Yang, Kai‐Cheng, et al. "Arming the public with artificial intelligence to counter social bots." Human Behavior and Emerging Technologies 1.1 (2019): 48-61.

[7] Cresci, Stefano, et al. "Fame for sale: Efficient detection of fake Twitter followers." *Decision Support Systems* 80 (2015): 56-71.

[8] Stefano Cresci. “A Decade of Social Bot Detection.” *Communications of the ACM*, October 2020, Vol. 63 No. 10, Pages 72-83  
10.1145/3409116