Network-based spam filter on Twitter [*] [†]

Ziyan Zhou
Stanford University
ziyanjoe@stanford.edu

Lei Sun
Stanford University
sunlei@stanford.edu

ABSTRACT

Rapidly growing micro-blogging social networks, such as Twitter, have been infiltrated by large number of spam accounts. Limited to 140 characters, Twitter spam is often vastly different from traditional email spam and link spam such that conventional methods of content-based spam filtering are insufficient. Many researchers have proposed schemes to detect spammers on Twitter. Most of these schemes are based on the features of the user account or the features in the content, such as the similarity or the ratio of URLs.

In this paper, we propose a network analysis based spam filter for Twitter. By analyzing the network structure and relations between senders and receivers, this spam filter does not require large data collection up front, thus is able to provide almost real-time detection for spams. By using the public API methods provided by Twitter, our system crawled the users in the sub-graph between suspicious senders and receivers. Then we analyze the structure and the properties of the sub-graph and compare them with those we collected from legitimate senders and receivers. Our study showed that spammers rarely have a network distance of 4 or less from their victim. Using a sub-graph of diameter 4 constructed between sender and receiver, we can further improve the recall of our spam filter with promising results by utilizing network-based features such as number of independent paths and normalized page ranks.

Keywords

Twitter, Spam filter, Real-time, PageRank

1. INTRODUCTION

Online social networks have become the most popular and attractive services on Internet. According to Alexa, among the top 20 websites in the US, 8 of them are social networks. In addition, more than half of them provide some social network functionality as part of their services. Among the most successful social networks, Twitter has grown tremendously over the years and became the most fast-growing website. Globally ranked among the top 10 most visited sites by Alexa, Twitter has become one of the most popular social networks with 284 million monthly active users posting more than 500 million tweets per day[1], corresponding to 6,000 tweets in every second, which is about 4 times as many as of that in 2011. This enormous popularity of Twitter, however, inevitably attracts a lot of spammers.

There is some similarity between the purpose of the spams on Twitter and other traditional spams on email, such as distributing advertisement, phishing for account information and spreading malwares. Twitter spams have their own characteristics: The length of each tweet is limited at 140 characters, which is usually not long enough for spammers to put in the desired information. To overcome this restriction, spammers usually insert a shortened URL in their tweets. When that URL is clicked by the user, he or she will be redirected to the advertisement or phishing web page.

Due to the lack of the enough information, or even the misleading information in a spam tweet's body, most of the users are prone to become victims. For example, a previous study has showed that 45% of users on a social networking site readily click on links posted by any friend in their friend lists' accounts, even though they may not know that person in real life [3]. Spams has become one of the most serious problems faced by Twitter users. It is estimated that nearly 3% of the tweet traffic on Twitter are spams [8]. Meanwhile, the traditional methods of combating email spams is far from sufficient, not only because of the little amount of information per tweet available to spam filters, but also because of the complete ignorance of network topology.

1.1 Background

There are several types of user interactions on Twitter, all of which are often employed by spammers:

- **Follow**: A user follows another user. The user being followed will receive a notification. Spammers sometimes follow random user in attempt to get them to click on its profile that contains spam.
- **Mention**: A user post a tweet that contains another

[*]Source code for this project is available on GitHub under the MIT License at https://github.com/ziyan/unfollow

[†]Live demonstration online service for this project is available at https://www.unfollow.io/

user's username. The user being mentioned will also get a notification. Spammers often use mention to directly target users.

- **Reply**: A special case of mention, when two or more users participate in a conversation known as a "thread".
- **Retweet** and **favorite**: These actions taken by the user also generate notifications.
- **List**: A user can be added to a custom list by another user. This action also generates notifications to the user being added.

Although each of the relationship above can be modeled as a user network, in this paper, we will mainly focus our analysis and evaluation on the follower relationship.

1.2 Spam detection by Twitter

Twitter has its own systems and tools to detect spams automatically by heuristics, such as:

- Followed and/or unfollowed large amounts of users in a short period of time, particularly by automated API requests.
- Repeatedly follow and unfollow people in attempt to gain follow back and improve follower/friend ratio.
- Tweets consist mainly of links, and not personal updates.
- Duplicate content over multiple accounts or multiple duplicate updates on one account.
- Send large numbers of duplicate replies or mentions.
- Aggressive favoriting or retweeting.

These heuristics have effectively decrease the number of spams on Twitter, but there still exist about 1% of the spams spreading on the network without being detected [6].

Besides the automatic spam filter, if Twitter users see a spam, they can manually report it. But those reports not only take a lot of users effort and time, but also require a lot of human resources to determine the correctness of the reports themselves.

2. PRIOR RELATED WORK

Many researchers have proposed schemes to detect spammers on Twitter. These schemes include:

- **Honeypot-based approaches** where honey profiles are set up to attract spammers. [4]
- **Statistical analysis** based on features such as account age, content similarity, ratio of links, and number of followers. [2]
- **Link-based filtering** where spammed links are crawled, analyzed and blacklisted. [7]
- **Network-based approaches** that classify spammer based on network distance and connectivity. [6]

In [8], the author classifies spam tweets and legitimate ones using a naive Bayesian classifier. A total of 25 thousand users, 500 thousand tweets, and 49 million follower relationships are collected from publicly available data on Twitter for training and testing purpose. For network-based features, the author defined some new relations among users, such as "friends" and "mutual friends" based on the Twitter follower relationship. The results showed that "reputation" (defined as the ratio between the number of friends and the number of followers) had the best performance. In [2], the author uses a machine learning approach to detect spammers. A large dataset is collected, which includes more than 54 million users, 1.9 billion links, and almost 1.8 billion tweets. The training data is manually labeled. A total of 60 features are examined. The top 10 performing features include 6 network-based features, 3 content-based features and 1 account feature. The results shown in this paper indicated that while almost all legitimate users are correctly classified (96%), only a majority (70%) of the spammers are found. Additionally, the authors also explored the feasibility of detecting spam instead of spammers, the corresponding results are not as good as the one for detecting spammers.

These approaches achieved high accuracy, but also have some significant problems:

First, it is often difficult to disambiguate spam from legitimate tweets based on its content. For instance, many prior research suggest that the number of tweets with links is a strong indicator of whether a user is a spammer. After all, due to the limitation of 140 characters, spammer usually needs external links to further market their products or services. However, sending tweets with links is also a very common behaviour of normal users. For example, we've found that out of 200 recent tweets posted by Bill Gates, 187 of them contains a link. Many users tread a fine line between reasonable and spam-like behavior in their efforts to gather followers and attention.

Second, spammers are getting better at disguising themselves as legitimate users. Sophisticated spammers build diversified and well-prepared account profiles that appear legitimate in every aspect. They no longer leave the profile fields mostly empty. Some will even upload a realistic profile picture to confuse spammer detectors as much as their victim users. Many classifiers in prior research use behavioural features such as tweeting interval or content similarity. With the help of Twitter's public API and careful programming, those behavioural pattern distinctions between spammers and normal users is also disappearing. Similarly, sophisticated spammers often fabricate their number of followers by creating a large number of spam accounts and manipulate them to follow each other. A common trick that we have seen is that spammers would randomly follow a victim user, trying to trick him or her into following back, then later unfollow the victim user to achieve a high follower ratio. Once a large number of followers are manifested, the spammer profile will look very reputable.

Third, for majority of the prior research, account features cannot be collected until at least a number of malicious activities have been done by the spammers. This means that spammers will be detected only after they've sent a num-

ber of spam messages. There is an inevitable delay between spam account creation and its detection. When spammers are detected and blocked, they can simply create new accounts to continue their business. Therefore, we believe shortening the detection delay is of vital importance.

3. OVERVIEW

In this paper, we propose a network based spam filter. When a message is sent out to some other users on Twitter, our system immediately re-constructs the network between the message receiver and the sender by crawling along the follow relations from the receiver. After that, some features related to network structure are used to characterize and evaluate this sub-graph, and at last, a final judgement will be made for the suspicious message.

By analyzing the network structure and relations between senders and receivers, our spam filter does not require large data collection up front, thus is able to provide almost real time detection for spams.

3.1 Constructing graph between sender and receiver

To measure the network based characteristics between a tweet's sender and receiver, we need to construct a sub-graph $G'(V', E')$ of the whole Twitter's network $G(V, E)$. First, $G(V, E)$ has tens of millions of nodes, thus is impossible to be obtained. Second, most of the graph is irrelevant to our decision making. Moreover, due to the rate limitation of Twitter's API, we want to take one step further to minimize the number of nodes we have to fetch. Inspired by bidirectional breadth-first search, we generate $G'(V', E')$ by

- Generate $G_r(V_r, E_r)$ containing the receiver, the receiver's friends and the friends of friends, which is a tree rooted at node receiver, with the height of 2.
- Generate $G_s(V_s, E_s)$ containing the sender, the sender's followers and the followers of followers, which can also be considered as a tree rooted at node sender but with reversed links pointing to parent.
- Generate $G'(V', E')$ by joining $G_r(V_r, E_r)$ and $G_s(V_s, E_s)$

Using this method, it is guaranteed that if receiver and sender are connected, the distance between them is at most 4. An example of a such sub-graph is shown in Figure 1

To perform sub-graph construction on collected offline dataset, we have implemented a simple bidirectional breadth-first search algorithm. See Algorithm 1. For our online service that directly uses Twitter APIs, this algorithm has been adapted to utilize task queues and asynchronous callback. See Figure 2

3.2 Features

3.2.1 Distance

Distance is one of the most important features. It is defined as the minimum number of hops from source node to destination by following the out links. In Twitter, a following relation is considered as an out link to a neighbor node, and

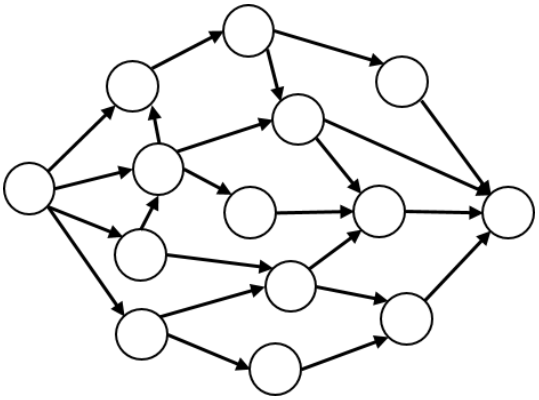

Figure 1: Example of sub-graph

Algorithm 1 Construct sub-graph using the 4 degree of distance method in Twitter network

```
1: procedure CONSTRUCT-4-DEGREE(sender, receiver)
2:     G ← empty graph
3:     queue ← {(sender, 0, friend), (receiver, 0, follower)}
4:     seen ← {sender → 0, receiver → 0}
5:     while QueueNotEmpty() do
6:         user, distance, direction ← Dequeue()
7:         if direction = friend then
8:             users' ← FriendsOf(user)
9:         else
10:            users' ← FollowersOf(user)
11:        for user' ← users' do
12:            if user' ∈ seen then
13:                continue
14:            add user' to graph G
15:            seen[user'] ← distance + 1
16:            if distance < 1 then
17:                Enqueue(user', distance + 1, direction)
18:    return G
```

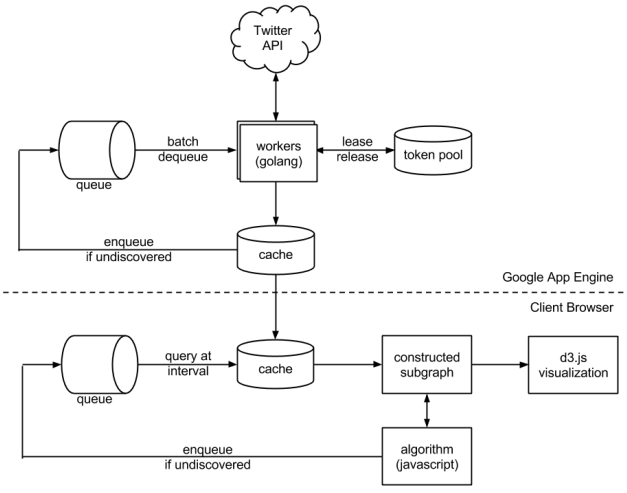

Figure 2: Cloud service architecture overview

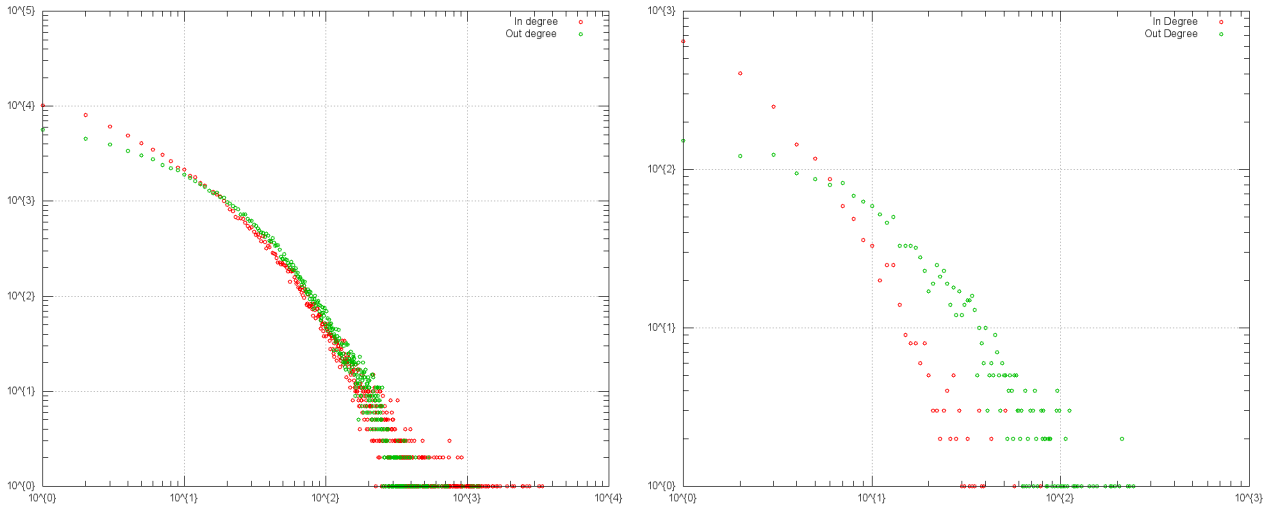

Figure 3: Degree distribution for legitimate users (left) and spammers (right)

all following relations are directional. So although it is easy for spammers to follow a large number of users, it is hard for them to get followers. Some clever spammers create many spam accounts and make them follow each other, but they still have long distance from outside legitimate users in spite of the seemingly high number of followers.

In [6], an investigation of the correlation between the distance and spammers was conducted, the result indicates that only 0.9% of the messages are spams within a distance of two and nearly 90% of the messages coming from a distance of four or more are spams.

3.2.2 *Common friends and connectivity*

The distance sometimes does not fully represents how strong the connection is between receiver and sender. Another measurement is to count the number of common friends. The reason is that even if a spammer happens to be a node on your out path within distance of 4, it is unlikely that you share more than 2 common friends. This technique is also very commonly used in modern social network to recommend "people you might know", the more common friends you have, the more likely you know each other. In general, we can model it as the connectivity of graph $G'(V', E')$. The connectivity or vertex connectivity $K(G)$ (where G is not a complete graph) is the size of a minimal vertex cut. And a cut of a connected graph G is defined as a set of vertices whose removal renders G disconnected.

3.2.3 *PageRank*

In addition, PageRank is another useful metrics on spam detection. PageRank, invented to analyze the ranking of websites, are also widely used on other networks to estimate the importance of a node. From intuition, a node becomes important if it is pointed by many other important nodes.

PageRank is defined and calculated as

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

where $p_1, p_2 ... p_N$, are the pages under consideration. $M(p_i)$ is the set of pages that link to p_i, $L(p_j)$ is the number of outbound links on page p_j, and N is the total number of pages. Its value are also computed by the left eigenvectors x_L of the transition probability matrix P such that

$$x_L P = \lambda x_L$$

where λ is eigenvalue. The N entries in the eigenvector x_L are the steady-state probabilities of the random walk corresponding to the PageRank values of the nodes.

It is worth mentioning that if we compute the PageRank values on $G'(V', E')$, sender can often get a high PageRank score if they are connected, even if it is a spammer. This result is expected because different from a general network, $G'(V', E')$ is only a sub-graph with edges flowing from receiver to sender. Therefore, we also need to build some baseline to compare the PageRank values for a spammer and the value for a legitimate user.

4. DATA COLLECTION

To properly evaluate different spam filtering methods, we have utilized an existing Twitter network dataset from [5]. This offline dataset contains 81,306 users and 1,768,149 follow relationships formatted as a directed graph. However, this dataset is unlabeled and was collected from over 2 years ago. Therefore, we re-collected up-to-date information about the given Twitter users using Twitter APIs.

Based on the data collected, we discovered that:

- 3,589 of these Twitter user accounts have since been suspended or deleted.
- There are 9,598 verified users, 3,454 protected profiles, and 1,221 users with default profile image.
- 468 users did not post any tweet.

4.1 Data Labeling

In order to correctly label nodes in the dataset, we use heuristics such as follower-friend ratio to rank all nodes in the network. Starting from the lowest-ranked node, we manually selected 100 spammer nodes by checking out their Twitter profile and tweet history. The 3,589 deleted / suspended accounts cannot be labeled as spammmers because we were not able to verify whether they are truly spammers.

Out of the set of all users followed by these 100 spammers, we randomly selected 100 nodes who we consider as legitimate users being spammed by the spammers. Lastly, out of the set of all users following these 100 normal users, we randomly selected another 100 nodes as our control set for comparison purpose. We manually verified that these randomly selected 200 legitimate users are indeed not spammers.

Figure 3 shows the degree distribution for legitimate users versus spammers. For legitimate users, both in-degree and out-degree follow power law distribution and these two curves almost overlap. For spammers, the distinction is rather significant. Most of the spammers have less than 50 followers but at the same time they usually have 10 times as many following links as followers. This result is expected because although it is easy for spammers to follow other users, it is difficult for them to get follow backs.

4.2 Twitter API Challenges

Twitter provides API for data collection. Originally we planned to collect a much larger dataset from the live Twitter network, however there are severe imposed rate limits for API requests that prevented it from being realistic. Each application registered is allotted a certain number of requests per 15 minutes window. As Twitter users authorize the application, which is also known as "Sign in using Twitter", the application is allotted additional rate limit quota for requests to be made on behalf of each user. Additionally, Twitter sets rather strict rate limit for the purpose of our data collection. For example:

- For retrieving a list of follower IDs of a particular user, the API returns a maximum of 5,000 user IDs per request, and limits the request rate to 15 requests in a 15-minute-window per authorized user.
- To retrieve recent tweets posted by a particular user, the API returns a maximum of 200 tweets, and limits the request rate to 180 requests in a 15-minute-window per authorized user.
- All API methods fall into one of these two rate limit buckets, namely 15 or 180 requests every 15 minutes per authorized user.

In other words, with only 10 valid user access tokens, we can look up detailed user profiles at the averaged rate of 12,000 nodes per minute, but we are only allowed to crawl incoming and outgoing edges for each node at the averaged rate of 10 nodes per minute.

To work with these strict limitations, we have developed the following strategies:

Table 1: Node pairs distance distribution from legitimate users to spammers

Distance	2	3	4	>4
Percentage	0.1%	3%	12%	>84%

- We avoided performing bulk data collection. Instead, we traverse the Twitter network locally around our users and trace nodes only when needed.
- We developed algorithms that can be resumed by breaking them down into smaller sequential pieces and execute them on a task queue. Whenever a rate limit is hit, we perform exponential back-off retries.
- We utilize aggressive caching in database and avoid duplicate API requests. We also diversify our API usage as much as possible because the rate limit is isolated per API method.
- Finally, we use global pooling of user tokens to utilize idle tokens that have not saturated their rate limit. Therefore, as we gain in the number of authorized users, we will have more tokens available for immediate API requests. This gradually enables us to achieve real-time performance.

5. EXPERIMENTS AND RESULTS

This section will present how the features perform on our collected offline data and how they can be used to differentiate legitimate users and spammers. And at last we will show the accuracy of our system when all features are combined to detect spams.

5.1 Distance

Distance is calculated on a set of 10,000 distinct pairs of nodes within legitimate user sets as well as pairs of nodes from legitimate user set to spammer set. As shown in Figure 4, red line represents the distance distribution between two legitimate users, and green line represents the distance distribution from legitimate users to spammers. On the whole network, the average distance between any two users is around 4.

In comparison, legitimate users have longer distance to spammers. Table 1 showed that only 3% of users have a distance less than 3 to spammers, and only only 15% of legitimate users are within 4 hops to reach spammers. Thus, if a message is from a user whose distance is less than 4, it is likely not a spam message. This result is similar to that concluded in [6].

Using this distance heuristic is not sufficient. 15% of legitimate users are indeed within 4 hops of a spammer. These users are susceptible to spamming. However, this result allows us to limit the size of the sub-graph that needs to be further considered and focus our following experiments on the sub-graph constructed with distance of 4 from user to suspicious spammers.

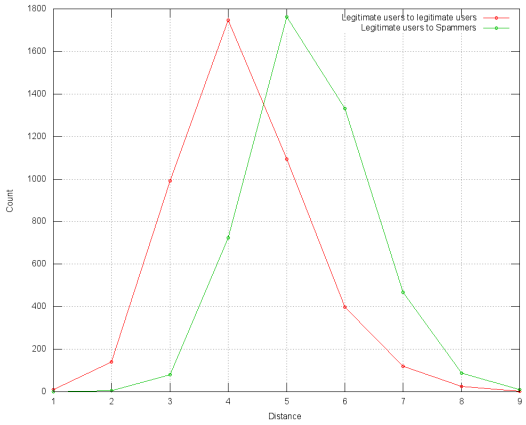

Figure 4: Node pairs distance distribution

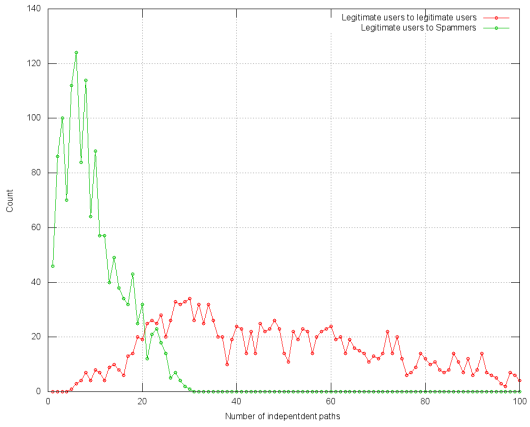

Figure 5: Number of independent paths

5.2 Number of independent paths

Due to the large out degrees in the Twitter dataset, number of independent paths are slow to calculate. Thus, we sampled a set of 1000 sub-graphs generated by randomly picking nodes in our labeled sets and reconstructing the network between them. The results shows a clear difference between legitimate users and spammers, an improvement from using merely node distance.

For pairs of nodes from the legitimate user sets, the number of independent paths averaged around 44.45. On the other hand, for pairs of nodes between legitimate user and spammers, the average number of independent paths is only 9.74. Furthermore, more than 92% of such pairs have less than 20 independent paths.

5.3 Page Rank

Page rank is another useful feature that can help us distinguish spammers from legitimate users. We computed the page rank scores on the same set of sub-graphs constructed above. In order make meaningful comparison of the page rank score for the sender node between different sub-graphs, we normalized the score based on the number of nodes in the sub-graph:

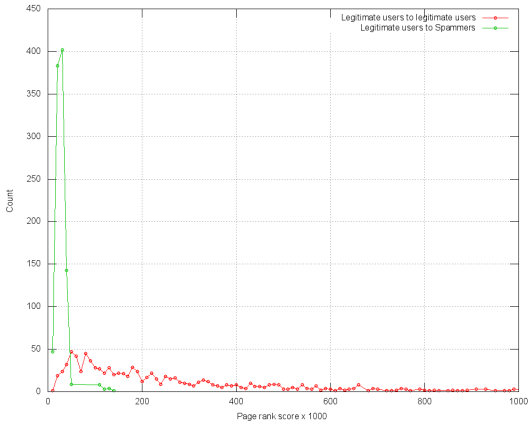

Figure 6: Normalized PageRank distribution

$$PR'(sender) = \frac{PR_G(sender)}{\sum_{v \in G} PR_G(v)} |G|$$

The distribution of the normalized page rank scores of the sender node is shown in Figure 6. In order to generate histogram of page ranks, we have rounded the scores to the nearest integer value after multiplying them by 1,000.

Similarly to number of independent paths, the page rank for spammers are concentrated at the lower score range, while the legitimate users usually have much higher scores. This is not surprising because there are only a small number of paths from the source node that can reach the spammer node, and also, spammers themselves don't have a lot of incoming links with high page rank scores to contribute. Even though spammers can follow each other to improve follower ratio, this behaviour does not contribute to a higher page rank score. Thus page rank cannot effectively spread to the spammer nodes.

6. CONCLUSION

Spams on Twitter social network is different from traditional E-mail spams, rendering traditional spam filters less effective. In this paper, we proposed a Twitter spam filter that works purely based on the structure of the network between the tweet sender and receiver. We present a methodology to collect limited but most useful subset of Twitter user information. Then three different network structure features are studied, experimented and evaluated on the collected dataset. These approaches showed promising results in distinguishing spammers from legitimate users. In further study, we will use a larger dataset to construct and train classifiers based on these network-based features.

7. LIVE DEMONSTRATION

As part of our project, we have set up live demonstration as a cloud service at https://www.unfollow.io/.

This service is hosted on Google App Engine. We developed this service to automate the collection of up-to-date data from the Twitter network. However, it is also very useful

as a Twitter social network explorer for running javascript-based graph algorithms. See Figure 7.

This service implements the strategies discussed in this paper in order to mitigate the effect of Twitter API rate limits. See Figure 2 for an architecture overview. The server aggressively caches discovered Twitter network topology using Google's big table. All data collections are performed asynchronously via task queues. Tasks are automatically retried with exponential back-offs.

We have also implemented a fine-grain token pool using task queues. When a new user sign in using Twitter account, the access token for the user gets added to the pool. We make multiple copies of the same access token in the pool. Each copy is tagged with a particular API endpoint. When we run data collection in the background, the algorithm leases tokens with the right tag from the pool. Then after use, it releases the token back to the pool with the proper updated remaining quota. When a token's quota is about to be exhausted, we automatically prevents it from being leased until the reset time is reached. With very few number of users, we were able to efficiently crawl a dataset of 80,000 nodes under 15 minutes using this system. (Crawling of edges take significantly longer due to stricter rate limits for those API endpoints.)

Users of our online demo is able to explore their Twitter network in real time. For example, the second graph in Figure 7 shows that the Twitter distance, based on reciprocal follow relationships, between Ziyan Zhou (@ziyan) and Dr. Jure Leskovec (@jure) is at most 3. There are also several distinct shortest paths to traverse between the two nodes.

8. REFERENCES

[1] Twitter usage and company facts. https://about.twitter.com/company.

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[3] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identify theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[4] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[5] J. McAuley and J. Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):4, 2014.

[6] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.

[7] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 447–462. IEEE, 2011.

[8] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

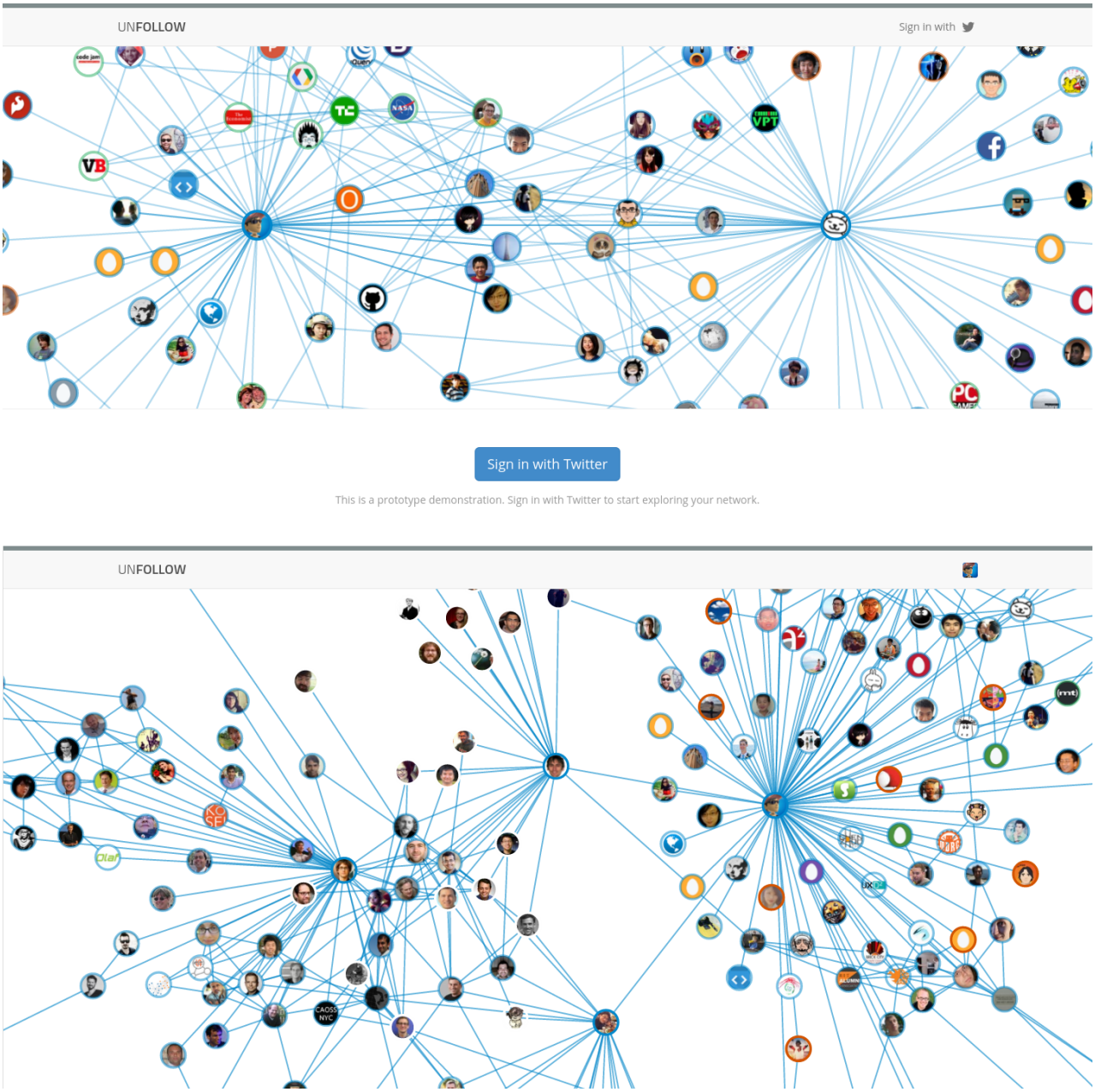

Figure 7: Live demonstration screenshots