Are You An Egg?: Detecting Twitter Accounts affiliated with Collusion Networks

Osama Khalid
osama-khalid@uiowa.edu

Muhammad Hammad Mazhar
muhammadhammad-
mazhar@uiowa.edu

ABSTRACT

In the context of online social networks like Twitter and Facebook, actions such as ”retweets” and ”likes” have become a currency used to measure profitability and popularity. The need to boost such actions on content published on these networks have given rise to black markets, composed of colluding members which provide actions for boosting content[12], also known as collusion networks. Detecting accounts involved in these activities(i.e. falsely boosting content with no future engagement) is a difficult problem, since unlike bot-based black markets, these markets involve human workers, who are often users of the social network targeted. In this project, we propose a set of heuristics for a user on Twitter, based on their activities and of those whom they follow (called friends from this point onward), and of their followers. We then use these heuristics to detect whether an account is a part of a collusion network or not. Evaluation results show that our heuristics far outperform the state of the art in collusion network from normal accounts with an accuracy of 0.997 and a false positive rate as low as 0.007.

The remainder of the paper is organised as follows. Section 2 discusses the related works which deal with fraud detection in Online Social Networks(OSNs). Section 3 presents our data collection and our approach. Section 4 discusses and analyzes our results and Section 5 concludes on the work done and discusses the advantages and limitations.

1. INTRODUCTION

1.1 Background & Motivation

Twitter currently has around 313 million active users monthly [15]. As such, it is a valuable platform for content creators, publishers, as well as advertisers. More commonly, Twitter has become a major source of news for many people. Popularity of any content posted on Twitter is measured by retweets, which are actions by users of Twitter to post the content on their own feed, where their own followers may also retweet it further, increasing coverage. As such, ’retweets’ become a form of currency by which people can increase the reach of their content on the social network. However, gaining these retweets is highly dependent on the quality of the content posted, as well as the initial reach of the poster. This may be difficult for publishers who are new to the platform. To cater to this demand, black-market sites have emerged, which provide services such as retweets and followers for a small fee. While they advertise themselves as providers of organic retweets, which means that the users who retweet a particular post by a poster will interact more with the poster later, this is seldom true. Most of these retweets are actually conducted as a part of collusion networks, where network participants perform actions such as retweeting, for the same to be done for their content. These actions result in an audience that offers no engagement, and hence is not an attractive audience for advertisers. This is problematic for Twitter, which earns most of its revenue from advertising. Furthermore, content posted using such methods is often flawed and dangerous, with malicious links posted via retweets(add citation), as well as news that may be fake.

A key factor in combating such activities is determining the actors in collusion networks, which appear as normal active users on Twitter. As such, we design our system to determine patterns which are similar in collusion network users, as opposed to normal users. Furthermore, since the main structure in collusion networks lies on cooperating with each other, we seek to expose such relationships by going in a level further into user behavior.

1.2 Problem Statement

We define our problem as follows. Given that we have access to how a particular user and its network (i.e. followers and friends) behave on Twitter, can we figure out whether that user is part of a colluding network. Or is that user legitimate, in terms of engaging with content that they share. We have access to tweet meta-data, such as the location from which the tweet (or retweet) was posted, the application used to post the tweet, for both users and their network. A key feature to this problem are the identification of patterns that are unique to users of a collusion network. Ideally, these patterns can be identified in a automated fashion, and provide enough information to confirm whether said user is a colluding network member, with high accuracy. Our method should also be robust to errors, meaning that we have low false positive rates.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

1.3 Proposed Approach

In this paper, we propose a novel method of detecting accounts on Twitter which are parts of collusion networks and engage in retweet fraud. Unlike previous methods [12] which try to detect the accounts involved in such activities by looking at their targets, we look at the fraudsters themselves and try to infer them based on their behaviour. Our approach is more robust to evasive techniques than current state of the art in the domain, since it considers both the fraudsters and their immediate network, and uses it as leverage for accurate results. We apply our approach to various services which are involved in reputation fraud by Twitter by providing artificial retweets. Our goal is to distinguish normal users from users who are part of a collusion network and retweet for cash. We analyze differences in retweet patterns between normal users and fraudsters, combining it with knowledge of Twitter user-agent differences between the two. We also analyze the immediate network (Twitter friends and followers) of users and look at their retweet and user-agent behaviour to aid in our detection approach.

1.4 Key Results

We build six classification models using the feature described above, with different classification techniques and evaluate these against our ground truth dataset. Evaluation results show that our approach can accurately distinguish normal users from collusion network users; our true-positive rate(TPR) is 0.997 when the false-positive rate(FPR) is 0.007.

We also tested our classifier on separate data sets from our training data, and achieved similar accuracy with those data sets.

2. RELATED WORK

CrowdTarget Song et. al[12] describe an approach to identify tweets on Twitter that have been tweeted using black market retweets, also known as crowdturfing. They posit that identifying users that engage in crowdturfing is difficult. This is because crowdturfing users share characteristics with normal users, which makes them hard to detect, without incurring high false positive rates. Hence CrowdTarget focuses on finding the target of such activities, which is the tweet content. Using features of the target objects that are tweeted, CrowdTarget identifies tweet content that has been crowdturfed. They achieve an true-positive rate of 98% when the false positive rate is fixed at 1%[12]

SynchoTrap Cao et. al [2] design and implement SynchroTrap to target attack campaigns on Facebook and Instagram, which are synchronized periods of activity designed to boost page and user popularity. Such attacks are perpetrated by black-market sites, using users colluding with them in order to satisfy customer demand. SynchroTrap focuses on the loosely synchronized nature of such attacks to identify malicious accounts [2], matching actions made by users such that users with similar patterns of action (within some constraints) are flagged as participants of attack campaigns. SynchroTrap has been deployed at Facebook and Instagram, and has managed to identify more than two million malicious accounts and over a thousand large attack campaigns[2]

3. DATA COLLECTION

In this section we explain the method for the collection of our ground-truth accounts of four different types; Normal Training Accounts, Fraudulent Training Accounts, Normal Testing Accounts, Accounts in the Wild. Table 1 summarizes our dataset.

Account Type	Number
Normal Training	1206
Fraudulent Training	1830
Normal Testing Accounts	124

Table 1: Twitter Accounts used for classification

3.1 Ground Truth Data sets

3.1.1 Normal Training accounts

We randomly selected 1,206 Twitter accounts from Twitter's verified accounts' list [14]. We assumed that any individual with a verified account is a normal user. We only considered the accounts which were publicly accessible because of the ease of data access. We used these accounts to build our training data set discussed in section 4. We only consider accounts that are public, since they allow easy data access.

3.1.2 Collusion Network Members

To get a list of accounts that are involved in collusion networks, we generate our own tweets and purchase the services of 5 different black market sites, which provide human retweets. These are Retweets Pro[7], Social Shop [10], TwiGain[13], RedSocial[9] and SocioBlend[11] For each site, we created a separate tweet, the contents of which are such that they should not attract any other attention, and purchased 500 retweets per site. Using Tweepy's Streaming API, we observe our created tweets for retweets by Twitter users. Since our tweets are designed to not attract attention of real users, all retweets of our tweets are by the users of the black market sites we paid for retweets. In total, from our 5 black market sites, we were able to collect 2576 accounts. We paid an average of $4.18 for 500 retweets. Table 2 summarizes this data set.

Black-Market Site	Retweets	Cost/500 Tweets
Retweets Pro	524	$4.00
Social Shop	506	$4.89
TwiGain	513	$3.99
RedSocial	527	$4.00
SocioBlend	506	$2.00

Table 2: Cost and Retweets gained from black-market sites

3.1.3 Normal Testing Accounts

To test our classification algorithms, we manually picked 124 twitter accounts. For this dataset we choose accounts that were not verified. To ensure that our dataset only had normal users we only selected accounts of people we personally knew of. Examples of such accounts include the Twitter accounts of University of Iowa faculty members, students of the university, other people in the academia and Twitter accounts of corporations at like PTCL, and the government of Pakistan.

3.1.4 Accounts in the Wild

To better understand how common fraudulent accounts are in the Twitter ecosystem, we choose 100 completely random accounts to test our algorithms on.

3.2 Collection Account Metrics

To collect various metrics we use to create our feature vectors, discussed in Section 4, we used Twitter's Stream API to monitor our honeypot tweets. We collect the list of accounts which retweeted our honeypots within 2 days of our purchasing the retweets. To get behavior metrics for both normal and fraudulent accounts, we use Twitter's REST API. Figure [number] shows an example of the metrics provided by the API

4. RESULTS AND DISCUSSION

4.1 Fraudulent Account Behavior

In this section, we analyze the behavior of fraudulent accounts and how it differs from the behavior of normal users.

4.1.1 Retweet Ratio

For each account in our dataset, we take the t most recent tweets that a user has posted on his timeline. From the set of tweet we count the number of tweets that were retweets R_T of other tweets. Using this we calculate the Retweet Ratio (R_R) as:

$$R_R = \frac{R_T}{t}$$

.

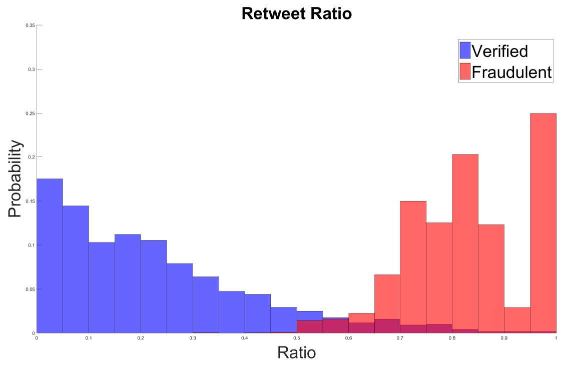

Figure 1: Retweet Ratio

Figure 1 shows the distribution of R_R of fraudulent accounts mapped against the R_R of Verified users. From the plot we can see that accounts that are part of collusion networks on average have a higher R_R compared to the R_R of real accounts. The mean of the R_R of fraudulent accounts is 0.71 while that of the real accounts is 0.24. This can be attributed to the fact that accounts who retweet for money generally retweet more than they tweet original content.

4.1.2 Web-App Ratio

For all the tweets t collected from an account, we calculated the number of times that the account had used the Twitter Web-App (n_W) as a user agent to post the tweets. The WebApp Ratio (W_R) is calculated as:

$$W_R = \frac{n_W}{t}$$

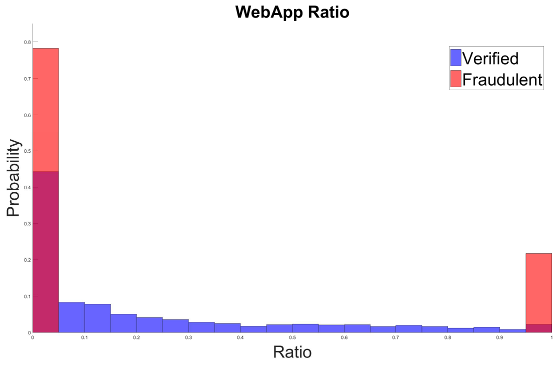

Figure 2: Web App Ratio

Figure 2 shows the distribution of Web App Ratios of Fraudulent Users compared to the distribution of Web App Ratios of Verified Users. From the distribution we can see that Fraudulent Users have a binary distribution as compared to Verified Users. Only a tiny minority of Verified accounts (0.035) uses only the Twitter Web App to post tweets, while around 0.455 of Verified accounts use multiple user agents. Whereas in case of Fraudulent accounts 0.21 use only the Twitter Web App while 0.79 do not use it. In our sample set there wasn't a single Fraudulent account which used multiple user agents.

4.1.3 Android Ratio

For all the tweets t collected from an account, we calculated the number of times that the account had used the Twitter for Android app (n_A) as a user agent to post the tweets. The Android Ratio (A_R) is calculated as:

$$A_R = \frac{n_A}{t}$$

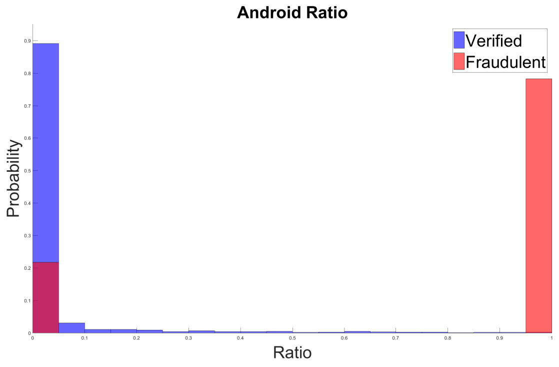

Figure 3: Android Ratio

From the distribution we can see that fake users have an extremely high rate of use of Android phones as compared to real users. Around 78.2 percent of fraudulent accounts used Android phones to tweet whereas around 89 percent of real users did not use Android phones. Again as in the case of the Web App user agent we see that the distribution is binary for fraudulent users; they either use the Android app or they do not.

4.1.4 iPhone Ratio

For all the tweets t collected from an account, we calculated the number of times that the account had used the Twitter iPhone (n_I) as a user agent to post the tweets. The Android Ratio (I_R) is calculated as:

$$I_R = \frac{n_I}{t}$$

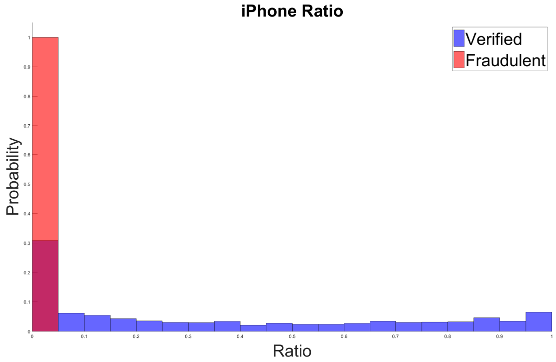

Figure 4: iPhone Ratio

From the distribution we can see that no Fraudulent user used iPhone whereas real users had a range of distributions.

4.1.5 Follower/Friend Retweet Ratio

We used the Retweet ratios of the friend and follower network of a user as leverage to measure the legitimacy of an account itself. The Friend Retweet Ratio of an account is measured as :

$$R_R^{Fr} = \frac{\sum_{n=1}^{f} r_i}{\sum_{n=1}^{f} t_i}$$

where an account has f followers. Each ith follower has r_i retweets, and t_i tweets in all.
The Follower Retweet Ratio of an account is measured as:

$$R_R^{Fo} = \frac{\sum_{n=1}^{f} r_i}{\sum_{n=1}^{f} t_i}$$

with f followers of the account
Figure 5 shows the distribution of the Retweet Ratio of the follower network of fraudulent and real users. From the distribution we can see that the Friend Retweet Ratio of real users follows a somewhat normal distribution with a mean at 0.28. The Friend Retweet Ratio of fraudulent users also follow a normal distribution however around 0.04 of the friends of Fraudulent users have a retweet ratio of 1.

Figure 6 shows the distribution of the Retweet Ratio of the friend network of fraudulent and real users. From the distribution we can see that the Follower Retweet Ratio of real users follows a somewhat normal distribution with a mean at 0.34. However with regards to fraudulent users a disproportionately large number of followers of fraudulent users have a retweet ratio of 0. This can be attributed to the fact that a large number of followers of fraudulent users have newly made accounts and have not tweeted or retweeted anything.

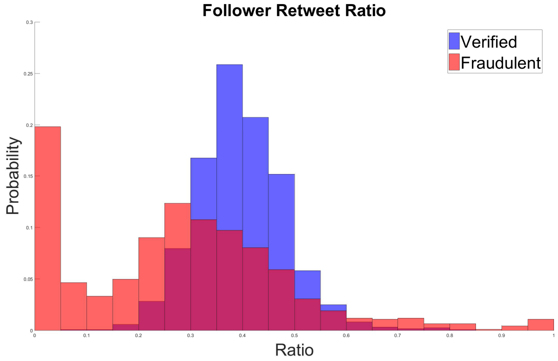

Figure 5: Follower Retweet Ratio

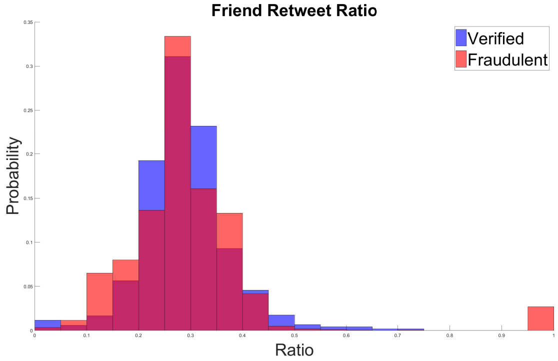

Figure 6: Friend Retweet Ratio

4.1.6 Friend/Follower Web App Ratios

We used the Web App ratios of the friend and follower network of a user as leverage to measure the legitimacy of an account itself. The Friend Web App Ratio of an account is measured as :

$$W_R^{Fr} = \frac{\sum_{n=1}^{f} W_i}{\sum_{n=1}^{f} t_i}$$

where the account has f friends, with the ith friend having W_i Web App tweets, and t_i tweets. While the Follower WebApp Ratio of an account is measured as:

$$W_R^{Fo} = \frac{\sum_{n=1}^{f} W_i}{\sum_{n=1}^{f} t_i}$$

with f followers of the account.

Figure 7 shows the distribution of the WebApp Ratio of the follower network of fraudulent and real users. Figure 8 shows the distribution of the WebApp Ratio of the friend network of fraudulent and real users.
From the Web App Ratio of the friends we can see that there isn't much difference in the distribution of the network of Real and Fraudulent Users. However from the Follower

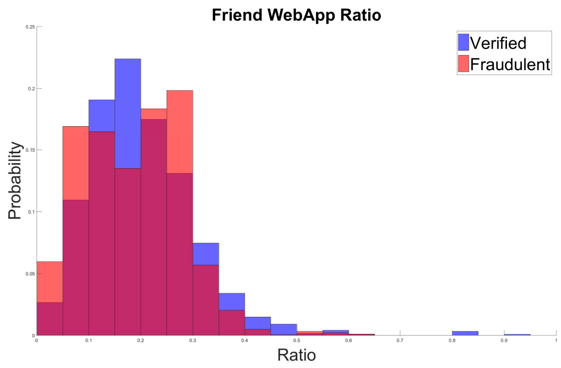

Figure 7: Friend Web App Ratio

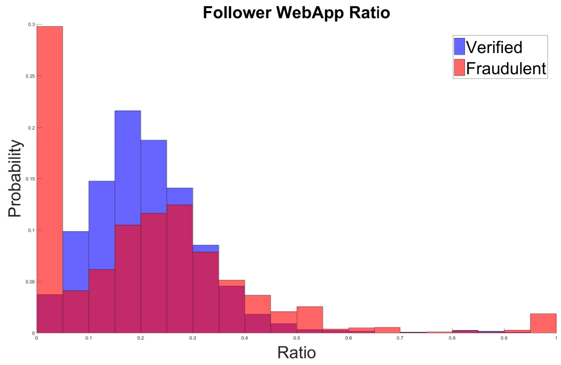

Figure 8: Follower Web App Ratio

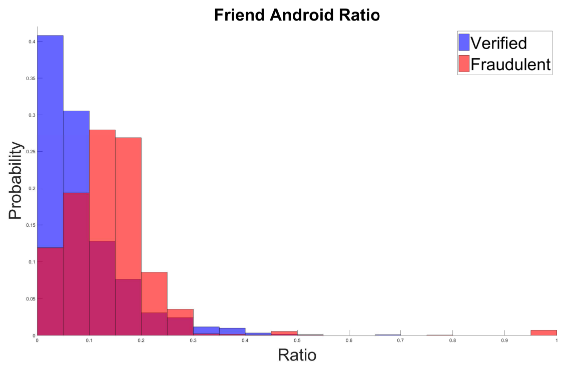

Figure 9: Friend Android Ratio

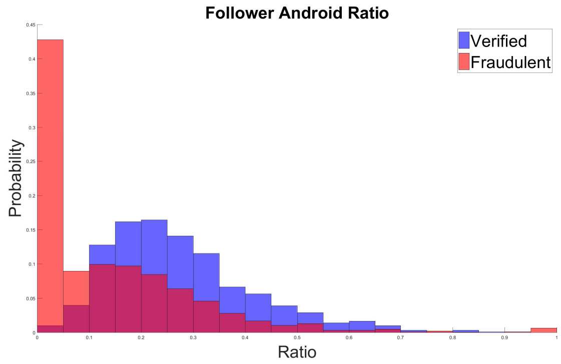

Figure 10: Follower Android Ratio

Web App Ratio we can see that an unusually high amount of Followers of Fraudulent Accounts have a Web App Ratio of 0. This as with the case of retweets can be attributed to the fact that most of the Friends have not tweeted anything during our experiment. Also the instances of a Follower of a Fraudulent account having a Retweet Ratio of 1 is much higher than that of the Follower of a Real account. This can be attributed to the fact that more real people follow real people while more fraudulent accounts follow fraudulent accounts.

4.1.7 Friend/Follower Android Ratios

We also look at the Android app usage of the friend and follower network of an account. We calculate the Friend Android Ratio as:

$$A_R^{Fo} = \frac{\sum_{n=1}^{f} A_i}{\sum_{n=1}^{f} t_i}$$

and the Follower Android Ratio as:

$$A_R^{Fr} = \frac{\sum_{n=1}^{f} A_i}{\sum_{n=1}^{f} t_i}$$

Figures 9 and 10 show the distribution of the Friend/Follower Android Ratio respectively, for both verified and fraudulent users. For verified users, the distribution is relatively normal for the people who follow them, but a high rate of their followers do not use Android. The nature of verified users suggests that the accounts that they follow do not use Android phones, which would follow from the Android Ratios that we encountered before. But for fraudulent users, there is a high ratio of users whose followers do not use Android. This would again follow from our Follower Web App Ratio insights, that the followers are relatively new to Twitter. Friends of fraudulent users, however, seen to use Android in some capacity. We posit that these might be real users these accounts would have followed to seem realistic.

4.1.8 Friend/Follower iPhone Ratios

We also look at the iPhone app usage of the friend and follower network of an account. We calculate the Friend iPhone Ratio as:

$$I_R^{Fo} = \frac{\sum_{n=1}^{f} I_i}{\sum_{n=1}^{f} t_i}$$

and the Follower iPhone Ratio as:

$$I_R^{Fr} = \frac{\sum_{n=1}^{f} I_i}{\sum_{n=1}^{f} t_i}$$

Figures 11 and 12 show the distribution of the Friend/Follower iPhone Ratio respectively, for both verified and fraudulent users. For verified users, the distributions are relatively normal. But for fraudulent users, there is a high ratio of users whose followers do not use an iPhone. These would be the new users mentioned before. But the Friend iPhone Ratio is similar for both verified and fraudulent users. This again

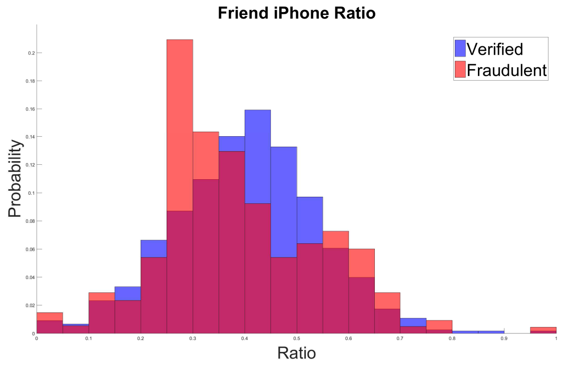

Figure 11: Friend iPhone Ratio

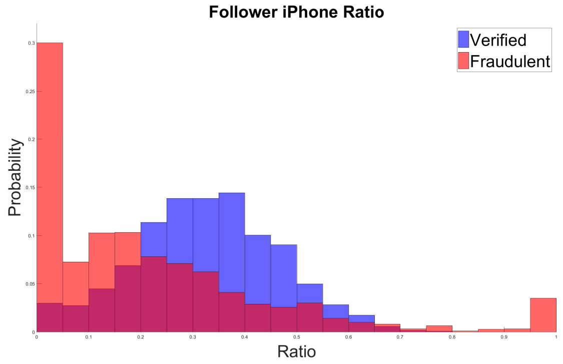

Figure 12: Follower iPhone Ratio

would indicate the possibility of following verified or real users to seem legitimate.

4.2 Detection of Colluding Users

In this section, we explain how we build our classifiers to detect accounts that are part of a collusion network. We try to distinguish such accounts from normal accounts based on the features discussed in the previous subsection.

4.2.1 Building Classifiers

We use the Normal Training and Fraudulent Training data sets to train our classifiers, the numbers of which are shown in Table 1. We use the Normal Testing data set to test our classifiers for accuracy. We trained 6 different classifiers; Support Vector Machines [cite], Artificial Neural Networks [3], Decision Trees [8], Random Forest [6], 5-NN [1] and Naive Bayes [5], on our training data. For each of the classifier, we selected a subset of our features for training to cater to various different cases, and then cross validated our results using 10-fold cross validation [4]. Table 3 shows the summary of our results.

We trained classifiers with subsets of our complete feature sets to determine the influence of certain features on accuracy. Only using the Retweet Ratios of training accounts gave us a maximum accuracy of 96.6% using a 5-NN classifier with an FPR of 0.06, while the lowest accuracy was of the Random Forest classifier of 95.98% with an FPR of 0.06. When we included the Retweet Ratios of the users network, we saw an increase in accuracy by 1%, with the highest being 97.79% obtained using Decision Trees, with FPR 0.04. The lowest accuracy in this case was 96.83% with SVMs. This validates our initial hypothesis that network characteristics can help in identification of fraudulent accounts.

To further understand the effect of network features, we also trained our classifiers with only network features of users. This gave us a highest accuracy of 94.13% with a FPR of 0.094 with RandomForest. The lowest in this case was with Naive Bayes; 75.03% with FPR 0.133. This shows that while network characteristics may be helpful in determining fraudulent users, they are insufficient on their own.

We finally use the complete feature set to train the classifiers. Our highest accuracy was achieved by Artificial Neural Networks; 99.9% with FPR 0.002. The lowest accuracy was by Naive Baiyes; 99.2% with FPR 0.019.

4.3 Testing the classifiers

To test the classifiers we had trained on the complete feature set, we selected 124 accounts of users that we were sure were real, but were not from the Twitter Verified list. These accounts ranged from the accounts of students from the University of Iowa, to the faculty and various other academics. They also included corporations like PTCL and even governments, to give us a diverse testing set. Figure 13 shows the accuracy of our classifiers on our testing set. On the testing set, we got the highest accuracy with Random Forests which managed to classify all the accounts as real, followed by artificial Neural Nets which only classified 1 out of the 124 accounts falsely as fake. We got the lowest accuracy from Naive Bayes which classified 20 accounts falsely as fake; however Naive Bayes in general had given us the lowest accuracy in our training cases as well.

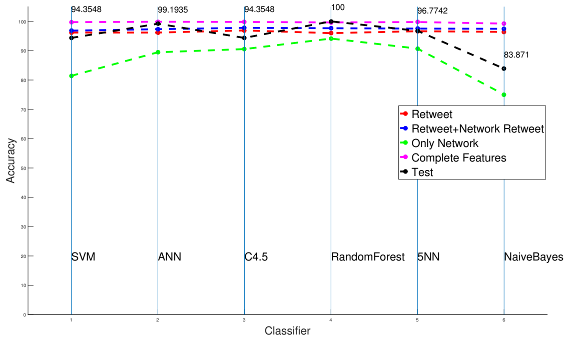

Figure 13: Accuracy achieved by different classifiers on different feature sets and test set

4.4 Running Classifier in the Wild

After testing our classifier, we ran it on a randomly selected subset of unclassified twitter accounts. Our classifiers on average classified between 26.5% to 27.3 % of the accounts as fraudulent. From this we can infer than Accounts on twitter that tweet for cash are much more common than previously estimated.

We manually inspected 25 accounts that were classified as real by our classifiers along with 15 accounts that were classified as fraudulent. All the 15 fraudulent accounts had a very high Retweet Ratio and were basically following the

Feature set	SVM	Neural Net	Decision Tree	Random Forest	5NN	NaiveBayes
Retweet Ratio(User Only)	96.2%(0.069)	96.2%(0.069)	96.9%(0.064)	95.98%(0.061)	96.6%(0.061)	96.4%(0.063)
Retweet Ratio(User & Network)	96.8%(0.067)	97.3%(0.050)	97.8%(0.043)	97.7%(0.043)	97.5%(0.052)	97.5%(0.058)
Network Only	81.4%(0.228)	89.5%(0.141)	90.5%(0.129)	94.1%(0.094)	90.7%(0.109)	75.0%(0.133)
All Features	99.7%(0.007)	99.9%(0.002)	99.8%(0.005)	99.6%(0.007)	99.8%(0.006)	99.2%(0.019)

Table 3: Classification accuracy on 10-fold cross validation (false positive rates in parentheses)

trend we had discussed earlier.

4.5 Robustness

In this sub-section we discuss the robustness of our feature set and whether users can alter their twitter behavior to escape our classifiers. Of the 8 feature sets we had selected for our classifiers, users can easily manipulate their Retweet Ratio by tweeting random strings of data. However, it is a bit harder for the average twitter user to alter their twitter agent, because that usually involved purchasing a different device. As we have seen from the trend in the twitter agents that the fraudulent users have used; especially the fact that they do not use expensive devices like iPhones, we can easily assume that it is beyond the purchasing powers of the fraudulent users to have multiple devices. Fraudulent users can also alter their network by only following verified accounts. But this is much harder to do, since it involves manually and selectively following accounts which will alter the features of the network in a certain way that helps the Fraudulent accounts look real.

A feature we did not use in our experiments was the age of a twitter account. We acknowledge that age is an extremely robust feature because it is extremely hard to spoof the age of a twitter account and the average fraudulent user will not have the technical knowledge to spoof their age.

5. CONCLUSION

In conclusion, we designed an approach to detecting accounts on Twitter that engaged in reputation manipulation using retweets. We improved upon performance by other fraud detection methods, by incorporating features from the network associated to a user, not only their own features. We analyzed the behavior of verified and fraudulent users using features derived from their activities on Twitter, and showed how they differ between the two. Our methods achieved high accuracy, with false positive rates as low as 0.007 for true positive rates as high as 0.997. We also validated our approach by testing it on manually selected real non-verified users and randomly chosen users.

6. CONTRIBUTIONS

6.1 Hammad

I worked on collecting user data from Twitter's REST and Stream APIs, and deriving the features from the data in an automated fashion. I also typeset the report on LaTeX, along with figures and tables.

6.2 Osama

I worked on classification and testing of classifiers on our data and feature sets, as well as compiling the list of real non-verified users. I also wrote down most of the report in Word, which Hammad later typeset, as well as creating the figures and tables that were to be used. The features to derive were also of my design.

7. REFERENCES

[1] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[2] Q. Cao, X. Yang, J. Yu, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In *ACM Conference on Computer and Communications Security*, 11 2014.

[3] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. JesÃžs. *Neural Network Design*, volume 20. PWS, 2nd edition, 1996.

[4] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[5] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998.

[6] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[7] R. Pro. Retweets pro. http://retweets.pro/.

[8] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[9] RedSocial. Redsocial. https://www.redsocial.net/.

[10] S. Shop. Social shop. https://www.socialshop.co/.

[11] SocioBlend. Socioblend. https://socioblend.com/.

[12] J. Song, S. Lee, and J. Kim. Crowdtarget: Target-based detection of crowdturfing in online social networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 793–804, New York, NY, USA, 2015. ACM.

[13] TwiGain. Twigain. http://twigain.com/.

[14] Twitter. Twitter verified lists. https://twitter.com/verified/lists.

[15] Twitter. Company. https://about.twitter.com/company, June 2016.