Behavioral Tracing of Twitter Accounts

Neel Guha

Stanford University

Abstract. "Trolls" - individuals who engage in malicious behavior - are a common occurrence within online communities. Yet simply banning accounts associated with trolls is often ineffective as individuals may register new accounts under pseudonyms and resume their activity. In this paper, we demonstrate how this can be addressed through a *behavioral trace*. Specifically, we show that by analyzing the posts of an account, we can derive a semantic signature unique to the account's owner. By comparing the signatures of two accounts, we can determine whether they belong to the same user. We validate our techniques on a dataset of Twitter users, and explore different properties of our methods.

1 Introduction

In recent years, online communities have increasingly struggled with the emergence of malicious accounts. These accounts engage in adversarial behavior, often spreading harmful content or attacking other individuals on the platform. This especially prominent on Twitter, where most interactions are public and accounts are not required to correspond to real life identities (unlike Facebook).

Eliminating these malicious accounts is difficult for many reasons. Firstly, the process of banning accounts is resource intensive and arduous, occurring rarely and often too late. Platforms like Twitter often rely on some human validation before banning accounts, resulting in a backlog of flagged accounts. Additionally, once an account is banned, it is trivial for the individual to create a new account under a pseudonym. They can resume their malicious behavior through this new account, thus creating a perpetual cycle.

Though the process of banning accounts will likely remain arduous and time consuming due to legal/corporate policies and procedures, it should be possible to prevent banned individuals from creating new accounts, or at least detect when an account may have been created by an individual previously banned.

In theory, this could be achieved through phone verification, or IP address blacklisting. However, these can have unintended consequences. Asking for users to validate their accounts with phone numbers may expose individuals who live in oppressive countries and have a legitimate need for privacy. Such measures hamper their ability to use mechanisms like Tor to access Twitter[5]. Banning accounts on the basis of IP address is also ineffective, as an individual could merely switch to a different network to create/access their account. If an individual uses a public machine (in a cybercafe or library), banning on IP address

may prevent large numbers of other individuals from accessing their accounts on the same machine.

In this paper, we present *behavioral tracing*: a method by which accounts created by the same individual can be identified and linked on the basis of the content of the accounts. Intuitively, an account's posts represent the topical interests and idiosyncrasies of its owner. Thus, in examining an account's posts, we should be able to derive a signature unique to the account's owner. We refer to this as a *trace*, and demonstrate it can be constructed. By comparing the traces of two different accounts, we can predict whether or not they were owned by the same individual. Applying this in the context presented above, we can use a trace to examine newly created accounts and determine if they resemble a banned account.

Our work is novel in our focus on semantic signatures. Unlike prior work, we formulate an authorship model based primarily on the content produced by a user (as opposed to the user's relations in the network graph, or lexical clues). Rather than constructing user-specific classifiers to identify accounts belonging to the same user, we introduce a single method applicable to all users. Specifically, we derive a vector space representation for each account (based on the account's post) where the distance between two accounts is indicative of the likelihood that they originate from the same user.

2 Related Work

There is a wealth of literature on different techniques for establishing authorship [9]. [18] presents methods for authorship identification of postings in online communities. The authors experimented with a variety of features (lexical, structural, content based, etc.) and models (neural networks, support vector machines, etc) to establish authorship of different posts. Though this is similar to our work, there are several key differences. The posts analyzed in [18] were on average over 150 words - well over the 140 character limit of Twitter. Additionally, the goal of the work was to establish the authorship of single posts, and not a collection of posts (corresponding to a single account). Though there has been work on establishing authorship in a Twitter context, it has primarily focused on using lexical and syntactic features [11][2]. In our paper, we demonstrate how authorship can also be determine by using extracting a semantic (topic based) signature for every user. To the best of our knowledge, this is a novel approach.

It is important to distinguish prior literature on spam and troll detection from our work. We focus on "linking" Twitter accounts to establish when two accounts were launched by the same user. Though a primary application of this work may be in detecting trolls, it could also be used to detect when a single individual is seeking to influence a discussion through the creation of multiple accounts. Much of the prior work on spamming and trolling focuses on leveraging network or language characteristics to identify common traits of banned accounts.

There has been significant prior work on the role of spammers within social networks like Twitter. Many, like [17], have focused on characterizing the nature

of spamming Twitter accounts. These works have demonstrated the techniques spammers use to promote content, and various approaches that could be used to detect them. It is important to clarify the distinction between spam detection, and the focus of our paper. Spammers primarily use platforms like Twitter to propagate commercial content, and convince users to take certain actions (clicking a link, downloading some software, purchasing a product). Spam accounts tend to be "fake" accounts that aren't tied to any real individual, and are often controlled by bots. In contrast, we focus on "real" accounts that are controlled by real individuals, and represent their interests. These individuals are thus significantly less likely to follow the follow the behavioral patterns of fake spam accounts. Our work is partially inspired by our prior work in [7], which present several methods for identifying web users across different browser sessions. Though we incorporate some prior techniques, both our approaches and the nature of the problem are very different.

Prior work has also focused on identifying "trolls" or adversaries within social networks [15] [10].[4] presents techniques for detecting trolls within social media networks. However, they assume that trolling individuals create fake troll accounts in addition to their real account. Further, the fake account is followed by the real account, and regularly interacts with the real account. On a limited sample of accounts, they present techniques for identifying the authorship of individual tweets. Our work doesn't make these assumptions. [3] analyzes "anti-social users" to determine characteristics of banned users. However, their work focuses less on identifying specifier users, and more on analyzing the behavior of banned users on numerous internet forums.

Similarly, there has been significant work on de-anonymizing social network users by utilizing information about network relationships [8] [6] [19]. In particular, [14] demonstrates how anonymized users with accounts on both Flickr and Twitter can be identified using graph topology. [16] attempts to identify Twitter accounts on the basis of browsing histories. By analyzing the t.co URLs visited by a user, they can determine the combination of accounts the user must have been following, which in turn can be used to identify the user's account. However, this approach fails to derive a fingerprint based on the user's interests - a critical contribution of our work. Furthermore, they require the browsing history of a user, a data source that is not often available.

3 Behavioral Trace

In this paper, we introduce *behavioral tracing*, a method to identify when posts from two Twitter accounts were authored by the same individual. There are many cases where this technique may be applicable. For example, we could apply it to determine when a previously banned user has returned to a platform (i.e. Twitter) and continued their activity under a pseudonym. Alternatively, a user may decide to operate two accounts within a particular community (to reinforce their opinions or create a perception of popularity). Behavioral tracing would allow us to identify such cases.

Our intuition is that a user's tweets are drawn from a fixed distribution governed by the user's interests. Given enough tweets from a single user, we can derive some approximation of the original distribution (a user's *behavioral trace* - also referred to as a user's *trace*). By comparing the extracted approximations from two accounts, we can determine when two accounts are in fact run by the same user. Thus, if we compared the trace from a banned account to the trace of an active account, we can determine when a user reenters the platform under a pseudonym. In this section we formalize the notion of a behavioral trace, describe how it can be used, and where limitations exist.

This approach also assumes that users maintain a consistent interest distribution and that a significant fraction of tweets posted by a user (regardless of the account used) are drawn from this distribution. If this condition were violated - for example, if a user had different interest distributions for different accounts - then it would be significantly harder to extract a meaningful trace. Thus, we assume that when a user has been banned from the platform and returns under a pseudonym, their tweets continue to be drawn from their original distribution. In other words, a user's interests are maintained between both accounts, and their behavior does not significantly alter.

There are however, several important limitations to acknowledge. Over time, a user's interests are likely to change. Hence we can expect that in the longer term, a user's interest distribution will gradually shift, making it harder to identify a user. This is something we hope to explore in future work. Additionally, if an individual creates two accounts but uses them for significantly different aims (professional and personal), the traces extracted won't be similar enough.

We now formalize the notion of a behavioral trace. We imagine a user u having a set of topical interests characterized by a distribution B over all possible interests/topics. Furthermore, we assume that every tweet (t_i) authored by u is sampled at random from B. Thus, we should expect that as a user posts more tweets, their collection of tweets grows more representative of their interests (the distribution of t_i's should resemble B).

Underlying our approach is the assumption that with high probability, any two users u_1 and u_2 will have different interest distributions (B_1 and B_2). We reason that individuals tend to be quite diverse in their interests. Though most users undoubtedly share common interests (sports teams, hobbies, etc.), the ways in which individuals process or share information tend to be highly personalized. When examined at a highly granular level, most individuals are distinguishable from one another. Thus, in our approach we seek to construct a *trace* for each user - an approximation of that user's interesting distribution inferred from their tweets. We treat the trace as a signature, and use it to fingerprint users.

We frame our problem as follows. Given two sets of tweets (T_1 and T_2) from two different accounts, our goal is to extract a trace (referred to as $\hat{b_1}$ and $\hat{b_2}$) that approximates the interest distribution of each account. If the traces are sufficiently similar, then we can determine that they must correspond to the same interest distribution, and that the same user is responsible for writing both sets of tweets. However, if they are sufficiently different, then we can determine

that refer to different interest distributions, and that both sets of tweets were written by different users.

4 Methodology

We formulate the task as follows. Given a set of tweets from n users, we partition each user's tweets into 2 separate *accounts* (giving a total of $2n$ accounts). Our goal is to re-identify the accounts by determining which originate from the same user.

4.1 Approach

We attempt to map each account to a vector based on its interests/behavior/-topics. Importantly, we seek to do so in a manner such that accounts corresponding to the same user are close to each other in this vector space. Prior work has demonstrated how word embeddings (e.g. Word2Vec) can capture rich semantic meaning in a way that traditional bag-of-words models cannot [13]. By constructing models to predict a word from its context (or vice versa), these models allow us to map words/phrases to vectors. Most notably, words that are "close" to each other in the vector space are likely to share similar contexts (and thus meaning).

In this work, we draw on Doc2Vec[12], an extension of the Word2Vec model that allows us to construct representations of variable length (i.e. documents). Our approach is motivated by the intuition that we can effectively construct a trace for each user by relying on word embeddings. In doing so, we can derive a vector for each account where the distance between accounts reflects the likelihood that they originate from the same author.

In this work, we collate all tweets from an account and treat the account like a single "document". We run Doc2Vec on the collection of accounts to derive a vector representation for each account [1]. Rather than compute a similarity score between every pair of accounts, we run k-means clustering to sort the accounts into different clusters (on the basis of their inferred vectors). In doing so, we're able to learn the "neighborhood" of an account - other accounts that look similar and are thus more likely to originate from the same user. Relying on this intuition, we thus only calculate a pairwise similarity score for accounts within the same cluster. We assume that accounts in different clusters correspond to different users. We find that this is a relatively safe assumption which allows us to significantly reduce the run time.

After deriving a location for each account in the vector space, we seek to identify the accounts in its neighborhood that could originate from the same user. For two accounts represented as the vectors a_i and a_j, we calculate $Score(a_i, a_j)$ in the following manner.

$$Score(a_i, a_j) = \frac{Cosine(a_i, a_j)}{\sum_{k=0} \frac{1}{Cosine(a_i, a_k)} + \sum_{k=0} \frac{1}{Cosine(a_j, a_k)}} \tag{1}$$

We describe this as a "weighted similarity" function, which weighs the similarity of two accounts by how dissimilar they are. It is not sufficient to say that two accounts are similar. Rather, we can only be confident that two accounts correspond to the same user if they are both similar to each other and dissimilar to other accounts. If we have two accounts a_i and a_j such that a_i is similar to a_j but both a_i and a_j are similar to the bulk of the accounts in our data set, we are less confident that a_i and a_j originate from the same user. it is probable that a_i and a_j (and the accounts they are similar to) belong to a mass of users whose behavior is too shallow or generic to discern. Conversely, if a_i and a_j were similar to each other but different from other accounts, we would be significantly more confident that both accounts originated from the same user. Hence, our scoring function is weighted by both account similarity and account dissimilarity.

For calculating the similarity between two accounts a_i and a_j, we use the cosine similarity metric, a common measure in information retrieval. For two n-dimensional vectors, the cosine similarity is calculated by

$$Cosine(s_i, s_j) = \frac{s_i \cdot s_j}{||s_i||||s_j||}$$

For each account, we deem the account with the highest score that exceeds the threshold to be from the same author. If no accounts have a score above the threshold, then the account in question is deemed not to share an author with any other account in the dataset. If multiple other accounts have scores which exceed the threshold, we only pick the account with the highest score. As we discuss in the next section, this approach is highly flexible, allowing us to achieve different types of results by varying the cutoff score used.

4.2 Evaluation

We measure the success of our approach using the precision-recall framework. Precision is defined as the proportion of account pairings we identify that are correct.

$$\text{Precision} = \frac{|S_t \cap S_p|}{|S_p|}$$

where S_p is the set of account pairings we predict and S_t is the set containing all pairs of accounts that originate from the same user (truth). Recall is defined as the proportion of same user account pairs that are identified by our methodology, or

$$\text{Recall} = \frac{|S_t \cap S_p|}{|S_t|}$$

In the context of our application, precision is the proportion of identified account pairings that do correspond to the same user. Recall is the proportion of same-user account pairings that we do identify.

Using the precision-recall framework to evaluate our approach allows us to modulate the type of result achieved based. Depending on the context in which

we're applying the methodology, this can differ. Sometimes perhaps, we may require a strategy that delivers high precision. This would be preferable, for example, if we chose to be conservative in our identification of accounts. Alternatively, we may want to flag as many accounts as possible. In this case, we would prefer a strategy which delivered a high recall (even at the cost of precision).

4.3 Baseline

To establish a baseline, we simulate an adversary randomly guessing accounts as pairs. We do this by randomly generating a score between $[0, 1]$ for each pair of accounts. We pick the cutoff that maximizes the F1 score and report results at that threshold.

In addition, we offer a more advanced baseline by running K-Means clustering directly on the generated Doc2Vec vectors for each account. Specifically, we set the number of desired clusters equal to the number of users. If two accounts are contained in the same cluster, we predict those two accounts to originate form the same user.

5 Data

Using the Twitter API, we collected 1,270,999 tweets from 1849 users. Of these, 678,403 were retweets and 592,596 were original tweets by users. Figure 1 shows a cumulative histogram of the number of tweets for every account in the dataset. The vast majority have fewer than 2000 tweets. Figure 2 is a histogram of the proportion of retweets for all accounts (the fraction of an account's tweets that are retweeted). The majority of accounts in our dataset are regularly active, with half posting at least 1.84 times per day.

Given that the focus of this work was on using semantic clues to develop unique identifiers for different Twitter accounts, we took care to clean tweets so that the algorithm would not identify accounts on the basis of their network properties. Specifically, we removed all account handles from the text from every tweet (e.g, "**@**exampleAccount").

Method	F1 Score	Precision	Recall
Behavioral Tracing (No Retweets)	0.54	0.54	0.54
Behavioral Tracing (All Tweets)	0.69	0.70	0.69
Raw K-Means	0.45	0.45	0.45
Randomized Baseline	0.00045	0.00086	0.000345

Table 1. Results of different approaches

We evaluated our algorithm as follows. We split our dataset of users into two groups - a "training" set and a "testing" set. Within each set, we split each user into two separate accounts (with each account containing half of the user's

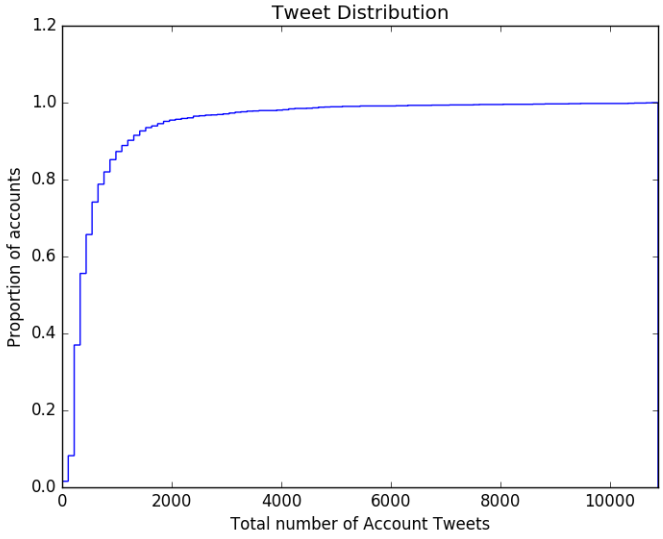

Fig. 1. Cumulative histogram of the number of tweets for every user in our data set

tweets). We applied our algorithm on the training set, and identified the cutoff distance at which the F1 score was maximized. We then applied our algorithm to the test set, using the cutoff score derived from the training set to predict which accounts belonged to the same user. We repeated this process 25 different times (sampling a different training and testing set each time).

This particular procedure allows us to justify the final cutoff used to identify accounts. We can imagine that in different contexts, a different cutoffs might be necessary. "Learning" it in this manner will allows us to better approximate an optimal cutoff.

Additionally, we experimented with the effect of retweets on our approach's performance. We thus ran two variations of our strategy. In the first, we ignored all retweets by accounts, using only "original" tweets to construct account traces. In the second variation, we used all of an account's tweets (including retweets) to construct the trace. Table 5 presents the results of these two versions, along with the baseline performance.

We experimented by varying the number of tweets each account was generated from. We see that as the number of tweets per account increases, the algorithms performance improves (Figure 4). However, we observe that the overall performance of the algorithm appears to level off after roughly 200 tweets.

We also experimented by varying the number of users targeted by our approach. We find that generally, as the number of users analyzed increases, the algorithm's ability to extract a uniquely identifiable fingerprint decreases.

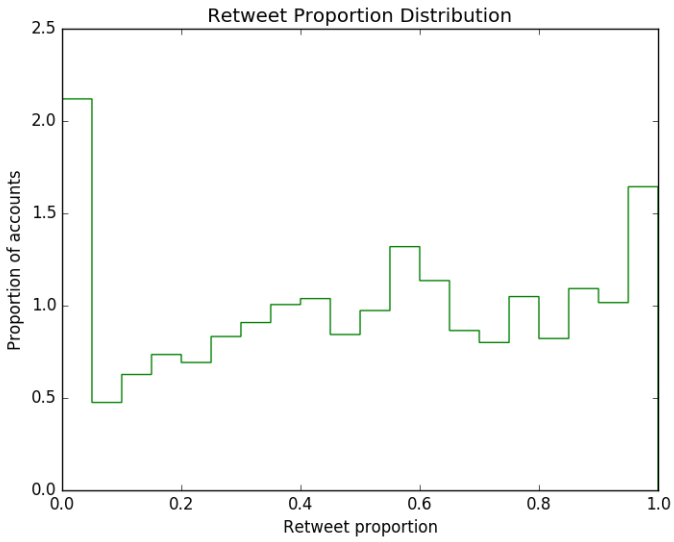

Fig. 2. Histogram of the proportion of tweets that were retweets for every user in our dataset

6 Discussion and Conclusion

The results in Table 5 demonstrate the effectiveness of our approach. Our algorithm exceeds both the randomized and naive clustering baseline, suggesting that our methods are capable of both successfully constructing unique traces, and using these trace to identify when tweets from two accounts are authored by the same user.

Our techniques demonstrate significantly improved results when we use an account's reweets to derive a semantic signature. There are several ways to interpret this result. Its possible that by using an account's retweets, our extracted semantic signature is influenced by the user's location in the Twitter network. Users are more likely to retweet accounts that they are following/followed by. Thus, when the majority of a user's tweets are retweets, the extracted semantic signature is effectively a reflection of the network structure surrounding the user.

However, a user's retweets are likely to reflect their interest profile. Furthermore, every account they retweet is also likely to have an extractable semantic signature (given enough tweets). Thus, we can view the extracted semantic signature for user not solely as their own, but as a composition of the semantic signatures of the accounts they frequently retweet.

We also find that performance in general improves as the number of tweets sampled for each account increases. Intuitively, this follows. As we gather more tweets from an account, we're able to better approximate the user's profile, and

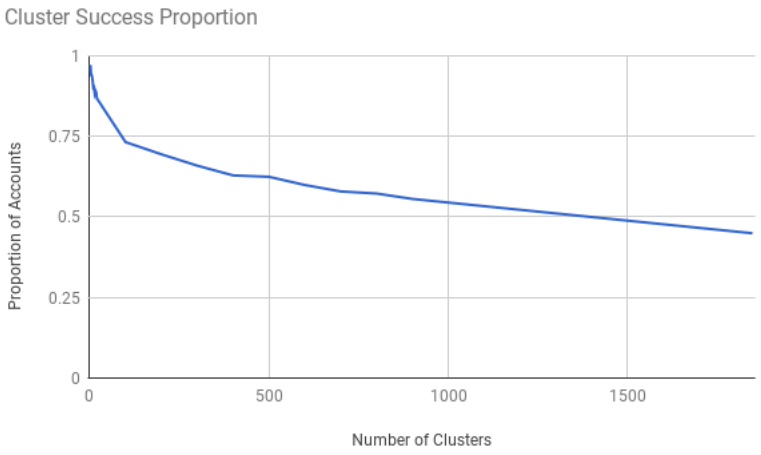

Fig. 3. Proportion of accounts grouped into the same cluster for different numbers of clusters

thus build a better trace. After a while however, there appear to be diminishing returns.

Additionally, we find that as the number of accounts we run our algorithm on increases, performance tends to decrease. As we grow our sample, we can imagine that accounts grow less distinguishable, and tend towards a more general, "average" interest profile. In these cases, it becomes hard for us to extract a unique trace for each account. However, the results in Figure 5 suggest that our strategy still finds success for larger samples of users. It's likely that our approach is conducting a variant of "outlier detection", in effect identifying users who are sufficiently different from all others.

Additionally, we find that that when the sample of users is small, the cutoff learned on the training set results in poorer performance on the testing set (farther away from the optimal point). When the cutoff is large however, we find that the performance on the training set is comparable to the test set.

The methods we present can be extended beyond the problem posed in this paper. The ability to construct fingerprints for users on the basis of their behavior has wide ranging implications for privacy and security. Broadly speaking, behavioral tracing is applicable in any domain where individuals take actions consistent with a set of interests, habits, or tasks. It could be used for example, to identify someone on the basis of online purchases. Equivalently, it could also be used to disambiguate between multiple individuals using a single account (e.g. on Netflix). At its core, behavioral tracing offers a way of uniquely identifying individuals on the basis of their behavior. Because behavior is hard to mask or alter, behavioral tracing is especially potent.

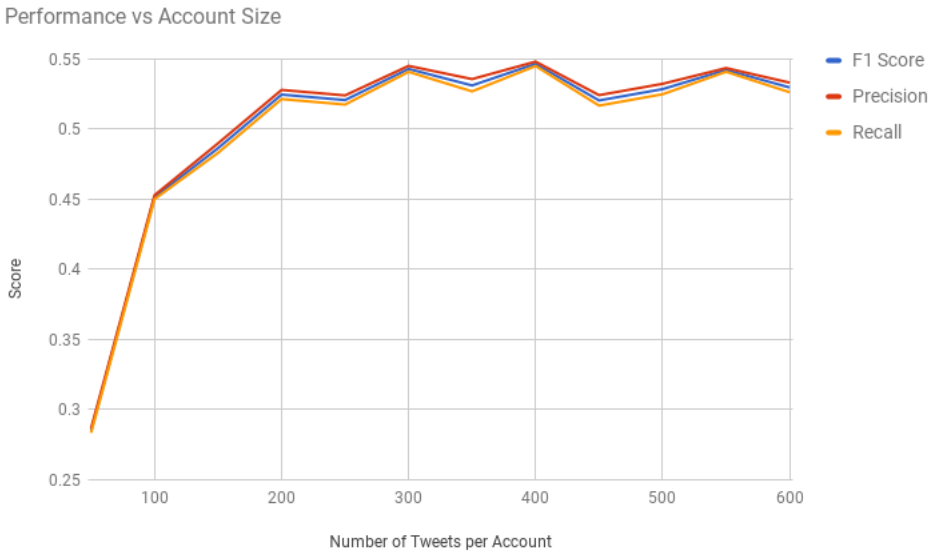

Fig. 4. Algorithm performance as the tweet sample size varies

In summary, the primary contribution of our work is *behavioral tracing*, a topical authorship model for Twitter. In framing a user's tweets as samples from their interest distribution, we demonstrate how users can be fingerprinted on the basis of a semantic signature. Validating our approach on real world Twitter data, we demonstrate how it can find success at identify users across different accounts.

7 Acknowledgments

We'd like to thank Anand Shukla, Ramakrishnan Srikant, Dan Boneh, Ramanathan Guha, Mehran Sahami, Lea Kissner, Scott Ellis, and Jonathan Mayer for their advice and guidance on this project.

References

1. Deep learning with paragraph2vec. https://radimrehurek.com/gensim/models/doc2vec.html.
2. Bhargava, M., Mehndiratta, P., and Asawa, K. *Stylometric Analysis for Authorship Attribution on Twitter*. Springer International Publishing, Cham, 2013, pp. 37–47.
3. Cheng, J., Danescu-Niculescu-Mizil, C., and Leskovec, J. Antisocial behavior in online discussion communities. *CoRR abs/1504.00680* (2015).

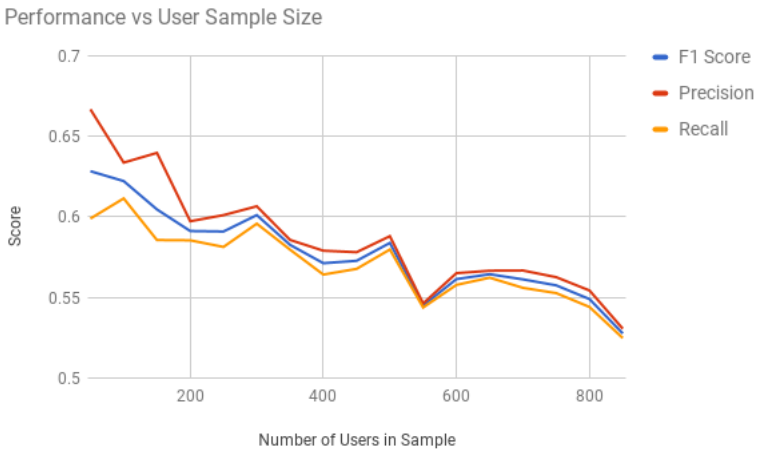

Fig. 5. Algorithm performance (for train and test sets) as the number of users in the sample varies

4. Galán-García, P., de la Puerta, J. G., Gómez, C. L., Santos, I., and Bringas, P. G. *Supervised Machine Learning for the Detection of Troll Profiles in Twitter Social Network: Application to a Real Case of Cyberbullying.* Springer International Publishing, Cham, 2014, pp. 419–428.
5. Gibbs, S. The Problem With Twitters New Abuse Strategy. https://www.theguardian.com/technology/2015/mar/04/twitters-new-bid-to-end-online-abuse-could-endanger-dissidents-analysis.
6. Gross, R., and Acquisti, A. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2005), WPES '05, ACM, pp. 71–80.
7. Guha, N. Semantic identification of web browsing sessions. *CoRR abs/1704.03138* (2017).
8. Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow. 1*, 1 (Aug. 2008), 102–114.
9. Houvardas, J., and Stamatatos, E. *N-Gram Feature Selection for Authorship Identification.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 77–86.
10. Kunegis, J., Lommatzsch, A., and Bauckhage, C. The slashdot zoo: Mining a social network with negative edges. In *Proceedings of the 18th International Conference on World Wide Web* (New York, NY, USA, 2009), WWW '09, ACM, pp. 741–750.
11. Layton, R., Watters, P., and Dazeley, R. Authorship attribution for twitter in 140 characters or less. In *2010 Second Cybercrime and Trustworthy Computing Workshop* (July 2010), pp. 1–8.
12. Le, Q. V., and Mikolov, T. Distributed representations of sentences and documents. *CoRR abs/1405.4053* (2014).

13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. Distributed representations of words and phrases and their compositionality. *CoRR abs/1310.4546* (2013).
14. Narayanan, A., and Shmatikov, V. De-anonymizing social networks. In *2009 30th IEEE Symposium on Security and Privacy* (May 2009), pp. 173–187.
15. Ortega, F. J., Troyano, J. A., Cruz, F. L., Vallejo, C. G., and Enrquez, F. Propagation of trust and distrust for the detection of trolls in a social network. *Computer Networks 56*, 12 (2012), 2884 – 2895.
16. Su, J., Shukla, A., Goel, S., and Narayanan, A. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17, International World Wide Web Conferences Steering Committee, pp. 1261–1269.
17. Thomas, K., Grier, C., Song, D., and Paxson, V. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (New York, NY, USA, 2011), IMC '11, ACM, pp. 243–258.
18. Zheng, R., Li, J., Chen, H., and Huang, Z. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology 57*, 3 (2006), 378–393.
19. Zhou, B., and Pei, J. Preserving privacy in social networks against neighborhood attacks. In *2008 IEEE 24th International Conference on Data Engineering* (April 2008), pp. 506–515.