Leveraging network topology for better fake account detection in social networks*

Björn Bebensee
Dept. of Comp Science and Eng.
Seoul National University
bebensee@snu.ac.kr

Nagmat Nazarov
Dept. of Comp Science and Eng.
Seoul National University
nagmat@snu.ac.kr

Abstract

Due to their popularity online social networks are a popular target for spam, scams, malware distribution and more recently state-actor propaganda. In this paper we review a number of recent approaches to fake account and bot classification. Based on this review and our experiments, we propose our own method which leverages the social graph's topology and differences in ego graphs of legitimate and fake user accounts to improve identification of the latter. We evaluate our approach against other common approaches on a real-world dataset of users of the social network Twitter.

Keywords: Fake account detection, social graph, network topology

1 Introduction

Today people all around the world use online social networks (OSNs) not only for personal connections but also for entertainment, to share opinions, to read news and inform themselves and to exchange knowledge and information. With their rise in popularity OSNs in the past decade however, they have become a target for abuse by malicious actors who are spamming the network, attempting to scam users, distribute malware, boost a legitimate user's popularity or increase the visibility of certain content. Furthermore with the 2016 United States presidential election the focus has been on social media rather than traditional media for the first time and the concern over widespread *fake news* influencing public opinion as well as social bots pushing state-actor agendas has been growing [1, 8] with some recent research focusing on identifying fake news using data mining methods [13].

Many OSNs spend a considerable amount of money and time in the form of manual labor on identifying and removing fake accounts. Our goal is to build on previous research

*working title

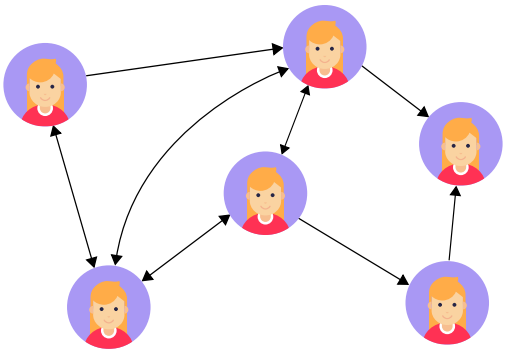

Figure 1: A social graph

and to improve the classification process for more effective and efficient identification of fake accounts. Unlike other approaches which focus only on basic profile features [6, 12] or temporal patterns in account activity [3, 7, 9], our approach belongs to a category of graph-based approaches to fake account identification.

Given the directed graph $G = (V, E)$ induced by the social network's structure as well as additional classification features m_v for every $v \in V$, we want to identify all nodes $v \in V$ that are likely to correspond to fake accounts. We call this graph a social graph (figure 1). For social networks with bidirectional connections i.e. through friend requests an undirected graph may be used to describe the network structure. It is however vital for the classification task that the false-positive rate be kept low as suspensions of legitimate user accounts can degrade the experience enormously.

In this paper we propose a novel approach to identification of fake accounts in OSNs by exploiting differences between the ego networks of legitimate users and fake accounts as well as weak links between communities of real users and communities of fake accounts. We first obtain a labelled dataset of legitimate and fake accounts on Twitter. As Twitter's developer policy limits the ways in which scraped user data may be published, these datasets typically contain a list of user IDs which we will then have to scrape to construct the social graph. We will explore this social graph and run experiments to find predictive features based on which we develop a novel approach. Finally, we evaluate our approach in experiments against other popular approaches for fake account detection in OSNs.

The rest of this paper is structured the following way: first we do a short survey of related work and their approaches (Section 2), based on the findings in these papers we will explore a dataset of Twitter users and fake accounts and their social graph, propose our own classification method, and evaluate our approach in experiments.

2 Related Work

In this section we will list papers that each member has read and reviewed as part of a first survey into the topic, along with a summary of their main ideas, how they can be of use for our own approach and possible shortcomings of the approach or important points not

addressed by the paper.

2.1 Papers read by Björn Bebensee

2.1.1 Detecting Clusters of Fake Accounts in Online Social Networks

Main idea: As opposed to previous literature which approaches the fake account classification problem on a per-account basis, Xiao, Freeman and Hwa [14] suggest a different approach which uses an approach based on clustering instead. They suggest that as more efficient way of identifying a set of spam accounts made by a single spammer, one might classify entire clusters of users to be legitimate or fake instead of single user accounts. Furthermore their approach focuses on identifying and removing fake accounts before they can interact with legitimate users and spam the network so as to prevent damaging the experience of legitimate users. As they want to stop fake accounts as early as possible and only limited information becomes available during registration Xiao et al. focus on these few features which are available at registration time.

The authors divided their machine learning pipeline into three major parts: a cluster builder, a profile featurizer and an account scorer. The cluster builder takes a raw list of accounts and builds clusters of accounts where the clustering criteria can be simple (i.e. share a common feature) or more complex (like k-means). These clusters, along with the features needed for the profile featurizer, are then labelled as real or fake. If there are accounts of both groups in one cluster, it is labelled according to a threshold x. The featurizer extracts features from the set of accounts in one cluster to find a numerical representation (an *embedding*) which can then be used by the account scorer to score the cluster. The authors test a number of models for the account scorer, specifically logistic regression, random forests and support vector machines. They find that for their use-case random forests perform best, with a recall slightly better than SVMs. Overall the model showed good performance in tests on in-sample data as well as a newer out-of-sample dataset. The authors have sinced deployed it in production at linked in and restricted more than 250,000 accounts.

Use for our project: This paper is closely related to the approach we want to take to classification of accounts as genuine or as fake. Xiao et al. suggest classifying entire clusters of users rather than single users to leverage similarities between fake accounts. This technique could prove useful for our approach and can be used in combination with features from each user's social graph. It might be possible to cluster users based on graph features such as degree, number of triangles a node participates in and others.

Shortcomings: A classification of entire clusters as proposed by the authors may not perform as well if fake accounts are less homogeneous. Such a distribution of fake accounts will lead to clusters being more mixed and therefore to a higher number of false-negatives and false-positives. Additionally, this approach uses many features that only the social network operator has access to and that are not available in public datasets to cluster accounts.

Unfortunately, we do not have access to this type of data and different features may prove less effective for clustering.

2.1.2 Botnet detection using graph-based feature clustering

Main idea: In this paper Chowdhury et al. [4] explore the use of graph-based features for clustering in computer networks to detect botnets. As much prior literature has focused on flow-based or rule-based detection, the authors suggest using clustering to first identify clusters of suspicious nodes. The authors are using a self-organizing map (SOM) for dimensionality reduction and clustering by assigning each node to a different cluster according to the output of the SOM. The features used for clustering are node in-degree, out-degree, in-degree weight (i.e. how many packets are received), out-degree weight (i.e. number of outgoing packets), clustering coefficient, node betweenness, and eigenvector centrality. Finally they are classifying nodes in each cluster (except the largest as it is unlikely to contain bots) starting from the smallest cluster using their own bot-search algorithm which only requires examination of few nodes for classification. Chowdhury et al. show that their approach performs better than SVM classification on the CTU-13 dataset (a dataset of botnet traffic) using the same graph features.

Use for our project: Although the approach presented in the paper operates on an entirely different set of data, it is very similar to our goal in its nature. The authors want to identify a set of bad actors in a network given interactions between devices and given the network structure. As we are attempting to classify users in a social network according to the structure and topology of the social graph, we aim to use a set of graph-based features, similar to the features used in the paper, to cluster groups of users which we may subsequently classify jointly.

Shortcomings: Calculating all given graph features for all nodes in the graph will not scale very well. For the CTU-13 dataset used by the authors the computation took 30 hours on a supercomputer cluster. This is not an acceptable amount of processing power and time to detect social bots in social networks in (near) real-time in order to prevent interactions with real users. However, as the CTU-13 dataset contains much data and information that is not contained or necessary for an application on social graphs, some of the ideas from these paper may still be viable in our use-case. Further experimentation is required.

2.1.3 Aiding the detection of fake accounts in large scale social online services

Main idea: Cao, Sirivianos, Yang and Pregueiro [2] build on previous work in *sybil detection* that aims to use random walks to identify fake accounts (*sybils*) based on key observations made on the structure of social graphs. Specifically a main assumption in these fake account detection schemes is that the connectivity between real users and fake accounts is limited and lower than the number of inter-user and inter-bot connections. In this work Cao et al. propose a new algorithm called SybilRank which, unlike previous work in the field,

does not aim to make a binary classification of each user account but instead focuses on creating a ranking which allows for a measure of confidence in classifications as well as further challenges like *captchas* for suspicious accounts. The key idea behind this algorithm is that in a social network an early-terminated random walk starting from a real user account has a higher probability of landing at another real-user than at a fake account. Early termination is necessary for these random walks as the probability of landing at any node converges to a uniform distribution for random walks of sufficient length. The authors can thus use the degree-normalized landing probability of early-terminated random walks to rank nodes and leverage the fact that connections between real users and fake accounts are limited. They further propose a more effecient way of calculating the landing probability of random walks using power iteration.

Use for our project: Use for our project: Cao et al. show that it is possible to leverage the topology of the social graph, specifically the weak links between fake accounts and real users, to identify these fake accounts. As we plan to use binary classification for this task, it could prove helpful to include the degree-normalized landing probability for random walks as an additional graph feature either in the machine learning algorithm or for clustering of similar nodes, given that it can be computed efficiently enough which may not be the case for large-scale social networks.

Shortcomings: The authors suggest running the SybilRank algorithm periodically, i.e. once every month, which would give fake accounts a window of time that is big enough to interact with and impact real users' experience on the social network unlike the approach introduced by Xiao et al. [14].

2.2 Papers read by Nagmat Nazarov

2.2.1 Towards a language independent Twitter bot detector

Main idea: Lundberg, Nordqvist and Laitinen [11] present a language-independent approach to classify single tweets as either auto-generated (AGT) or human-generated (HGT). Their classifier consists of 10 tweet features:

a) *isReply* $\in \{0, 1\}$ indicates if a tweet is a reply

b) *isRetweet* $\in \{0, 1\}$ indicates if a tweet is a retweet

c) *accountReputation* given by number of followers divided by number of friends and followers

d) *hashtagdensity*,*urldensity*, *mentiondensity* given by number of occurrences divided by number of words in the tweet

e) *statusesPerDay* is the number of status updates per day

f) *favoritesPerDay* is the number of tweets favorited per day

g) *deviceType* $\in$ {web, mobile, app, bot, ...}

The authors find that decision tree-based supervised learning algorithms work particularly well on this type of problem. Out of the evaluated algorithms, random forests (RF) perform best.

Use for our project: We may focus on decision tree-based supervised learning algorithms and particularly RF for a set of tweet features (or similarly basic profile features) like this for classification of fake accounts in online social networks.

Shortcomings: The proposed algorithm does not perform as well as single-language classifiers. If enough resources are available it may be more sensible to train a single-language classifier for each language one wants to identify auto-generated tweets in rather than using a multi-language model. The model has only been trained on a small dataset of tweets in two languages and may perform better if other languages are used as well. Furthermore, the authors evaluated the model in only one other language, more extensive evaluation may be necessary.

2.2.2 A network topology approach to bot classification

Main idea: Cornelissen, Barnett, Schoonwinkel, Eichstadt and Magodla are proposing a graph-based network topology approach to the bot classification problem [5]. They propose utilizing the surrounding network topology of an ego in the social graph to determine whether the user is an automated agent or human. The ego graph of node n is a K-2 graph obtained by a crawler, that is a graph with all nodes i of distance $d(n, i) \leq 2$. Using clustering on features from the ego's graph such as the density, clustering coefficient and centrality among others, they achieve an accuracy of 70%. The authors suggest using such network analysis in conjunction with other methods for better accuracy.

Use for our project: The authors suggest to use centrality graph measure, for example celebrities have high indegree and low outdegree. For instance, celebrities on Twitter tend to have more people following them than they follow themselves. The authors also propose that the bots must have high outdegree but very low indegree, since most people will not follow back. We can use this proposal to distinguish the bots with non bots. [5]

Shortcomings: The false-positive rate is very high at around $\sim$45% which is not acceptable in a real-world setting as it can potentially lead to many falsely removed accounts belonging to legitimate users, impacting the user experience negatively. However, this may be in part due to obvious outliers that have not been removed as the authors state.

2.2.3 Bot Classification for Real-Life Highly Class-Imbalanced Dataset

Main idea: Typically the research on bot detection is based on particular botnet characteristics, but in this paper Sarah Harun, Tanveer Hossain Bhuiyan, Song Zhang, Hugh Medal and Linkan Bian develop three generic features to detect different types of bots regardless of specific botnet characteristics [10]. They suggest five classification models based on those features to classify bots from a large, real-life class-imbalanced network dataset. The authors show that the generalized bot detection methods perform better than the botnet specific methods.

They first filter out unnecessary data and then extract features from the rest of data by computing the previously developed three generic features for each source-destination pair of IP addresses. In the filtering step the authors remove any IPs which never act as a source and those which perform only single communication with another device. In the feature extraction step the following features are computed: 1) Falling rate of communication frequency, 2) median communication frequency and 3) source bytes per packet for highest communication frequency. Using the extracted features the authors explore a number of different supervised learning algorithms like Quadratic Discriminant Analysis (QDA), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), K-Nearest neighbors (KNN) and Random Forests (RF) for this highly class-imbalanced dataset. In their experiments they find that RF, KNN and even SVM perform poorly on imbalanced data and that QDA and GNB perform best for imbalanced datasets.

Use for our project: QDA and GNB perform much better than the other supervised learning algorithms on class-imbalanced data. Given such an imbalanced data distribution we should avoid using RF and KNN as these perform extremely poorly. Another important takeaway from this paper is that for accurate training of supervised models a balanced dataset is important and we should thus use such a dataset if possible.

Shortcomings: The biggest shortcoming of this paper is that it ignores passive or less active bots from the beginning. Furthermore, it is not a real-time detection system and works primarily by observing activity patterns. Although a similar approach is possible in social networks (i.e. accounts posting very frequently, sending spam links or only replies are likely to be fake accounts) but such activities can only be observed after the fact which makes this type of approach less useful in preventing fake account interactions with real users.

References

[1] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[2] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX*

conference on Networked Systems Design and Implementation, pages 15–15. USENIX Association, 2012.

[3] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. Temporal patterns in bot activities. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1601–1606. International World Wide Web Conferences Steering Committee, 2017.

[4] Sudipta Chowdhury, Mojtaba Khanzadeh, Ravi Akula, Fangyan Zhang, Song Zhang, Hugh Medal, Mohammad Marufuzzaman, and Linkan Bian. Botnet detection using graph-based feature clustering. *Journal of Big Data*, 4(1):14, 2017.

[5] Laurenz A Cornelissen, Richard J Barnett, Petrus Schoonwinkel, Brent D Eichstadt, and Hluma B Magodla. A network topology approach to bot classification. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 79–88. ACM, 2018.

[6] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71, 2015.

[7] Alceu Ferraz Costa, Yuto Yamaguchi, Agma Juci Machado Traina, Caetano Traina Jr, and Christos Faloutsos. Rsc: Mining and modeling temporal activity in social media. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–278. ACM, 2015.

[8] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378, 2019.

[9] Supraja Gurajala, Joshua S White, Brian Hudson, and Jeanna N Matthews. Fake twitter accounts: profile characteristics obtained using an activity-based pattern detection approach. In *Proceedings of the 2015 International Conference on Social Media & Society*, page 9. ACM, 2015.

[10] Sarah Harun, Tanveer Hossain Bhuiyan, Song Zhang, Hugh Medal, and Linkan Bian. Bot classification for real-life highly class-imbalanced dataset. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 565–572. IEEE, 2017.

[11] Jonas Lundberg, Jonas Nordqvist, and Mikko Laitinen. Towards a language independent twitter bot detector. In *DHN*, pages 308–319, 2019.

[12] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1065–1070. IEEE Computer Society, 2012.

[13] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[14] Cao Xiao, David Mandell Freeman, and Theodore Hwa. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pages 91–101. ACM, 2015.

A Appendix

A.1 Labor Division

During our work we will perform the following tasks

- Find a fitting dataset to test our approach, est. ∼3h [Bebensee]
- Scrape the social graph of accounts containted in the dataset and clean the data, est. ∼5h [Bebensee]
- Read more to our approach related work, est. ∼2h [Nazarov]
- Explore the graph data to find possible discrepancies between subgraphs of bots and subgraphs of real users, est. ∼6h [Nazarov]
- Run experiments on the data, est. ∼12h [Bebensee, Nazarov]
- Develop a classification approach based on our findings, est. ∼8h [Bebensee, Nazarov]
- Evaluate our approach in experiments, determine precision, recall, AUC (area under ROC), est. ∼10h [Bebensee, Nazarov]

A.2 Full disclosure w.r.t. dissertations/projects

Bebensee: His research and thesis is not related to this project in any way. He is working on neural machine translation and visual question answering.

Nazarov: His research and thesis is not related to this project in any way. He is working on designing an open framework for designing, implementing, and evaluating hardware and software components for solid-state drives.

Contents