Twitter Bot Detection Using Role Discovery and Machine Learning Classification Methods

Alex Day
School of Computing
adday@clemson.edu

Maria Diaz
School of Computing
mddiaz@clemson.edu

Sadegh Sadeghi Tabas
School of Computing
sadeghs@clemson.edu

1 Introduction

1.1 Problem Specification

The online world is dominated by online social networks (OSN) such as Facebook, Twitter, and LinkedIn. OSN represents a global platform through which people share and promote products, links, opinions, and news. These massive communities possess the ability to quickly spread information from one corner of the world to the other within minutes. The data-sharing feature of social networks allows users to distribute content and links; however, this feature is also commonly exploited by spammers and fraudsters. Social bots are programs that automatically generate content, distribute it via a particular social network, and interact with its users [16]. According to a recent study by Varol et al., between 9 percent and 15 percent of Twitter accounts are bot accounts, which is the equivalent of 48 million accounts [4, 17]. A further study found that social bots are responsible for generating 35 percent of the content that is posted on Twitter [5].

In some scenarios these bots are designed with good intentions. They can protect the anonymity of members as mentioned in related works or automate and perform tasks much faster than humans, like automatically pushing news, weather updates, or sending a thank-you message to your new followers out of courtesy [17]. On the other hand, social bots can be designed with the purpose of conducting malicious tasks such as spamming, malware dissemination, impersonation, or Sybil attack launching. One of the malicious functionalities of social bots is the power of dissemination of misinformation. For example, the Syrian Electronic Army hacked the Twitter account of Associated Press and announced that the White House was under attack and Obama was injured. This fake news led to a panic and huge loss in the stock market in 2013 [10].

In order to combat the malicious intent of some of these social bots, much work has gone into sophisticated bot detection and classification algorithms. **In this work we introduce a novel classification algorithm that bridges the gap between user and graph-based algorithms. By augmenting a naive version of the BotOrNot [17] classifier with graph based features we aim to increase the accuracy of the classification**.

1.2 Related Studies

Many studies have aimed to address the problems associated with the use of automated accounts on social networks which can spread spam, worms, and phishing links or manipulate legitimate accounts by hijacking and deceiving users [7, 11, 14, 15, 16, 18, 21]. Malicious accounts typically operate under a botmaster who controls a group of social bots to distribute spam or manipulate behaviors on a given social network [20]. For example, in Syria, a social bot was employed to flood Twitter with hashtags related to the Syrian civil war with irrelevant topics that redirected the attention of users from controversial government actions [1]. Social bots have also played a significant role in the uprisings that occur in the aftermath of major events such as elections or conflicts [3]. Gupta et al. [8] studied the fake content that was proliferated via Twitter during the Boston Marathon blasts and the role such content played in spreading rumors and misinformation. They found that bot accounts were created and generated after the blasts, many of which impersonated real accounts. The malicious activities of bots during events such as these can be used to spread spam. In addition, they can also cause financial harm, as was observed in the case of Cynk, which suffered a 220-fold drop in market price as a result of the activities of automated stock trading social bots [5]. Generally, social bot detection on social networks is performed by one or more of the three common methods mentioned earlier: Graph-based, crowdsourcing, and machine learning [2]. The graph-based method involves using the social graph of a social network to understand the network information and the relationships between edges or links across accounts to detect bot activity. The crowdsourcing method involves using expert annotators to identify, evaluate, and determine social bot behaviors. Finally, the machine learning method involves developing algorithms and statistical methods that can develop an understanding of the revealing features or behavior of social network accounts in order to distinguish between human- and computer-led activity.

Role discovery was first described as a series of steps that partition a graph's nodes into classes of organizationally similar nodes, or roles [12]. Topological features such as peripheral nodes, centrality, degree, and others are used to construct the similarity features of the graph while feature-based roles are derived from the non-graph-based attributes. Feature based roles are assigned by a refinement of feature equivalence. At the nexus of this, feature and graph-based roles can be utilized in tandem to produce a hybrid approach to role discovery. The graph, initial graph-based features, and preliminary role-based features can be used as parameters for a relational learning model to construct more complex features from the preliminary graph-based role discovery techniques.

GraphRole is a library of functions related to extracting meaningful graph features of a graph [9] and expands on the previous RolX github repository [13]. The class boasts two major functions: 1. A RecursiveFeatureExtractor class intended to automatically extract recursive features that describe regional structural properties of the inputted graph parameter. This process includes accessing topological node features, such as the degree, and ego-net features, such as neighbors, and aggregates nodes across their neighbor's features until no additional information can be gleaned. The recursive process begins by converting the graph node's structural properties to binary representations before aggregating them. Dimensionality is reduced by selecting strongly correlated features to drop given a threshold. 2. The RoleExtractor function classifies each node to a role by selecting it for role assignment based on the structural roles in the graph which represent those eigenvectors associated with the largest eigenvalues. Additionally, the RoleExtractor function employs structural similarity as opposed to structural equivalence to isolate roles and classify nodes. Commonly employed graph-based techniques were not scalable enough to describe the complex roles found within the twitter data, which necessitated the hybrid approach described in this summary. These additional metrics provided greater flexibility in obtaining enough information to perform accurate classification of Twitter bots.

2 Methodology

2.1 Datasets and Pre-processing

The dataset used for the classification is *botometer-feedback-2019* [19]. This is a corpus of 529 users with labels, either human or bot, and various selected attributes from those used in the BotOrNot [17] random forest classifier.

After this dataset was loaded into the pre-processing environment the list of accounts each user follows was scraped using the Twitter API. This information was used to build the network of users. Rather than link each user based on explicit following relations the links are created if there is an overlap in the accounts two users follow. This creates a more dense graph and allows the graph-based metrics to glean more information about how two users may be related. This resulted in 510 nodes in the graph with 35477 edges between them.

2.2 Classification Methods

Machine learning and deep learning algorithms are powerful models that first consider the learning styles that an algorithm can adopt. These approaches can be divided into 3 broad categories (i) supervised learning, (ii) unsupervised learning, and (iii) reinforcement learning. Supervised learning is useful in cases where a property (label) is available for a certain dataset (training set) but is missing and needs to be predicted for other instances. Unsupervised learning is useful in cases where the challenge is to discover implicit relationships in a given unlabeled dataset (items are not pre-assigned). Reinforcement learning falls between these 2 extremes; there is some form of feedback available for each predictive step or action, but no precise label or error message. In this study, we focused on using several supervised learning models as classification-based approaches to detect bots from twitter accounts. The scikit-learn and Keras libraries have been used to implement the following classification methods. The employed methods are including:

- Nearest Neighbours (NN)
- Gaussian Process
- Decision Tree
- Random Forest
- AdaBoost
- Naive Bayes
- Quadratic Discriminant Analysis (QDA)
- Deep Neural Network
- XGBoost

The results from training the models discussed above are listed in Table 1. These models were trained on three separate feature sets. The first one is a subset of the user features from the original dataset, this includes the number of tweets, the number of followers, the number of statuses the user has liked, the account age, and the number of tweets and followers for the user as a fraction of the account age. This is denoted as the User features. The second feature set is a set of graph metrics. These metrics are eigenvector and closeness centrality along with the results from the GraphRole role detection. This set is denoted as the Graph features. The third set, denoted Hybrid features, is the union of the previous two feature sets. All of the classifiers were trained with identical hyperparamters (besides a different sized input layer on the neural network).

3 Results and Conclusion

We found that while using the Random Forest classifier (which BotOrNot currently uses) the graph features do not add enough information to increase the accuracy. This relationship holds for all non-deep learning based classification models. However, we also found that the graph based features do increase the accuracy by over 8% when using a deep neural network classifier model and by almost 0.5% when using the XGBoost classifier.

This leads to the assumption that deep learning classification techniques can glean more information for classification from graph based features than less sophisticated models, and that the graph based features should be included in these models.

Table 1: Classification model accuracy (percent) results

Classification Methods	User Features	Graph Features	Hybrid Features
Nearest Neighbors	71.57	68.63	68.63
Gaussian Process	69.61	69.61	69.61
Decision Tree	74.50	70.59	69.61
Random Forest	82.35	66.67	67.65
AdaBoost	75.49	68.63	68.63
Naive Bayes	73.53	68.63	68.63
QDA	72.55	67.65	67.65
Deep Neural Network	68.63	70.20	**76.86**
XGBoost Classifier	78.43	69.12	**78.92**

Code and Data Availability

The code and data used for this project is available on GitHub[1]. The repo is comprised of the employed datasets and three Jupyter Notebooks which contain the code required to scrape additional twitter data, create the features, and train the classification models used.

References

[1] N. Abokhodair, D. Yoo, and D. W. McDonald. *Dissecting a social botnet: Growth.* content and influence in Twitter, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work Social Computing, 2015.

[2] E. Alothali, N. Zaki, E. A. Mohamed, and H. Alashwal. *Detecting social bots on twitter: A literature review.* in: 2018 International Conference on Innovations in Information Technology (IIT), 2018.

[3] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. *The socialbot network: when bots socialize for fame and money.* in: Proceedings of the 27th Annual Computer Security Applications Conference, 2011.

[4] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Detecting automation of twitter accounts: Are you a human. *bot, or cyborg? IEEE Trans. Dependable Secur. Comput*, 9, 2012.

[5] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The rise of social bots. *Commun. ACM*, 59, 2016.

[6] C. Freitas, F. Benevenuto, S. Ghosh, and A. Veloso. *Reverse engineering socialbot infiltration strategies in twitter, in: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* IEEE, 2015.

[7] C. Grier, K. Thomas, V. Paxson, and M. Zhang. *@ spam: the underground on 140 characters or less, in: Proceedings of the 17th ACM Conference on Computer and Communications Security.* 2010.

[8] A. Gupta, H. Lamba, and P. Kumaraguru. *$1.* 00 per rt bostonmarathon prayforboston: Analyzing fake content on twitter, in: 2013 APWG ECrime Researchers Summit, 2013.

[9] D. Kaslovsky. *"Dkaslovsky/GraphRole." GitHub.* github. com/dkaslovsky/GraphRole, 2019.

[10] D. Mail. *Syrian Electronic Army linked to hack attack on AP Twitter feed that 'broke news' Obama had been injured in White House blast and sent Dow Jones plunging.* 2013.

[11] S. Rathore, P. K. Sharma, V. Loia, Y. S. Jeong, and J. H. Park. Social network security: Issues. *challenges, threats, and solutions. Inf. Sci. (Ny)*, 421, 2017.

[12] R. A. Rossi and N. K. Ahmed. "role discovery in networks. *" ArXiv. org*, 3, Nov. 2016.

[13] B. Rozenberczki. *"Dkaslovsky/GraphRole." GitHub.* github. com/dkaslovsky/GraphRole, 2018.

[14] M. Shafahi, L. Kempers, and H. Afsarmanesh. *Phishing through social bots on Twitter.* in: 2016 IEEE International Conference on Big Data (Big Data), 2016.

[15] G. Stringhini, C. Kruegel, and G. Vigna. *Detecting spammers on social networks.* in: Proceedings of the 26th Annual Computer Security Applications Conference, 2010.

[16] V. S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer. The DARPA Twitter bot challenge. *Computer (Long. Beach. Calif)*, 49, 2016.

[17] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection. *estimation, and characterization. arXiv Prepr. arXiv1703*, page 03107., 2017.

[18] A. H. Wang. *Detecting spam bots in online social networking sites: a machine learning approach, in: IFIP Annual Conference on Data and Applications Security and Privacy.* Springer, 2010.

[19] K. Yang, O. Varol, C. A. Davis, E. Ferrara, A. Flammini, and F. Menczer. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies 1, no. 1*, 1.

[1] `https://github.com/AlexanderDavid/NetworkScienceFinalProject`

[20] J. Zhang, R. Zhang, Y. Zhang, and G. Yan. The rise of social botnets: Attacks and countermeasures. *IEEE Trans. Dependable Secur. Comput*, 15, 2016.

[21] X. Zhang, S. Zhu, and W. Liang. *Detecting spam and promoting campaigns in the twitter social network*. in: 2012 IEEE 12th International Conference on Data Mining, 2012.