Detect Me If You Can: Spam Bot Detection Using Inductive Representation Learning

Seyed Ali Alhosseini
Hasso-Plattner-Institute
University of Potsdam
Potsdam, Germany
seyedali.alhosseini@hpi.de

Raad Bin Tareaf
Hasso-Plattner-Institute
University of Potsdam
Potsdam, Germany
raad.bintareaf@hpi.de

Pejman Najafi
Hasso-Plattner-Institute
University of Potsdam
Potsdam, Germany
pejman.najafi@hpi.de

Christoph Meinel
Hasso-Plattner-Institute
University of Potsdam
Potsdam, Germany
christoph.meinel@hpi.de

ABSTRACT

Spam Bots have become a threat to online social networks with their malicious behavior, posting misinformation messages and influencing online platforms to fulfill their motives. As spam bots have become more advanced over time, creating algorithms to identify bots remains an open challenge. Learning low-dimensional embeddings for nodes in graph structured data has proven to be useful in various domains. In this paper, we propose a model based on graph convolutional neural networks (GCNN) for spam bot detection. Our hypothesis is that to better detect spam bots, in addition to defining a features set, the social graph must also be taken into consideration. GCNNs are able to leverage both the features of a node and aggregate the features of a node's neighborhood. We compare our approach, with two methods that work solely on a features set and on the structure of the graph. To our knowledge, this work is the first attempt of using graph convolutional neural networks in spam bot detection.

CCS CONCEPTS

• **Information systems** → **Social networks**; • **Security and privacy** → *Social network security and privacy*; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Social Media Analysis, Bot Detection, Graph Embedding, Graph Convolutional Neural Networks

ACM Reference Format:
Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. 2019. Detect Me If You Can: Spam Bot Detection Using Inductive Representation Learning. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3308560.3316504

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA
© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316504

1 INTRODUCTION

Online Social Networks (OSN) have provided a means of communication for individuals to share information and express their opinions in a free and simple manner. Twitter, Facebook and other social media websites have changed the way we consume news and interact with one another. The important role of these platforms has resulted in attempts by interest groups to influence users, seize their attention and ultimately change public opinion [3, 11].

Social Bots are automated user accounts operated by a computer program mimicking human behaviour with the intention of abusing the social media platform [3, 6]. They have evidently become a threat to online social networks with their malicious behavior spamming with advertisement and scam URLs, promoting a specific hashtag, spreading misinformation and impacting elections.

The research community has proposed several approaches for bot detection. The difference in these work varies depending on the definition of a bot account, the selected feature set representing accounts and the machine-learning algorithm used for classifying bot accounts from normal user accounts.

However, spam bot detection remains an open challenge for several reasons. The first reason relies in the definition of bot accounts as there is no single definition to precisely determine an account as a bot account. This is an important matter specially for building a ground truth dataset. Another issue as reported by [3, 17] is that bots have become more advanced and sophisticated in avoiding the existing proposed detection methods. In fact, bots have been evolving over time. [6] has given attention to the rise of social bots that are designed to emulate human-like behavior. The social bots are able to interact with other accounts, post tweets in different topics, and display a similar activity like humans [3].

Recent advances in deep learning for graph-structured data has led to a new method of representation learning named Graph Convolution Networks (GCNs) [9, 10]. The main idea of GCNs is to represent a node in a vector space based on its features and the features of its neighboring nodes using neural networks. The advantage of GCNs is that it captures both the node features and the graph structure to learn a low-dimensional representation of nodes [7, 8].

In this work we propose an inductive representation learning approach for bot detection based on the user profile features and the social network graph. The main contributions of this work are summarized as follows:

- We deploy graph convolutional neural networks on a well-known spam bots dataset previously used in the literature.
- We compare our approach with two algorithms by a MLP classifier and applying the Belief Propagation on the dataset.
- We show that using the graph structure in our method gains better performance in spambot detection.

The remainder of this paper is structured as follows. First we cover the previous related work in spam bot detection and graph convolutional neural networks. Section 3 describes in detail the dataset used. In section 4, we provide an overview of our methodology. We illustrate are results in section 5 and discuss the limitations of our work and suggestions for future work. Finally, we conclude this paper in section 6.

2 RELATED WORK

In this section, we first review the literature on spambot detection and compare each work by their definition for spambots, the features used and the classification algorithm they employed. Next, we look at graph convolutional networks.

[12] proposed a method working as a honeypot trap for bot accounts. They created 60 twitter accounts and started posting meaningless tweets that would have no interest for humans. Despite this fact, they were able to draw some accounts' attention to follow the accounts they made. Analyzing these accounts in detail showed that they were in fact bot accounts trying to increase their following list.

[17, 18] used a conservative definition for bot accounts considering only accounts who post URLs linking to malicious content. They also introduced and considered several robust features on the BayesNet classifier to predict spam accounts. Yang and et al investigated the different approaches bots take for avoiding detection by Twitter. Their findings show that bots tend to increase the reputation of their accounts by purchasing followers and posting more tweets.

[4] introduced a DNA-inspired technique that models each account as a sequence of behavioral information and detects spambots based on similar sequences. They categorized each users' tweets into different types and based on whether a tweet contains URLs, hashtags, pictures, etc. it will be assigned a different character. The similarity of the accounts is measured by the longest common substring in their DNA sequences.

BotOrNot [5] used the random forest classifier algorithm on more than 1000 features to detect bots. The features are categorized in 6 groups: network (degree distribution, clustering coefficient, ...), users' account information, friends (number of followers, followings, ...), temporal (tweet rate, ...), content (natural language processing, ...) and sentiment features. The downside of BotOrNot is that it was trained on English tweets so its performance declines on bots which are tweeting in another language than English.

DeBot [1, 2] is an unsupervised bot detection system. The idea behind their work is that accounts with a high correlation in their activities (tweet, retweet, ...) have a high chance of being bots. DeBot monitors the activities of accounts over a specific period and creates a time series for each account. It then clusters accounts based on the similarity of their time series using a lag-sensitive hashing method. Finally, DeBot reports the accounts with a high correlation as bots.

[13] defined spam bots as content polluters that try to take over a discussion for political or advertising reasons. Their approach considers individual tweets for detecting bots. Instead of focusing on the friend and follower network, they created the event network where the nodes are the users and the edges are based on users having tweeted on the same event. They also compute the diversity of a tweet based on the URLs and hashtags it has mentioned. Results of their work indicate that spam bots operate as a group often tweeting at the same time.

2.1 Graph Convolutional Networks

Graph structures are used in many domains and applications such as social networks, recommender systems etc. The challenging task for graph structures is how to use them in machine learning algorithms. The initial works in this area considered the statistical data of the graph like the degree of the nodes, the centrality and betweenness coefficients as features for training models. In other words, they considered the graph structure as a pre-processing step to extract structural information. Therefore, these approaches do not use the graph structure in the learning phase. Another downgrade for these approaches is that computing the graph statistics has high complexity and the output of it cannot be used on unseen data.

With the recent advances in Convolutional Neural Network (CNN) there have been efforts to adapt this popular deep learning model for encoding graph structures. Two main approaches have been used for embedding the graph structure into a dimensional space. The difference between these approaches is based on how the convolution operation is defined. The first approaches aim to take a fixed-length node sequence of the graph structure and directly use it in the original CNN models that work in the Euclidean domain. Alternatively, the other methods model the graph structure to non-Euclidean domains. [10] proposed graph convolutional network (GCN); that considers spectral convolutions on graph structures. The term convolutional is used since a node's neighborhood is considered as its representation. Their method can be considered as the initial steps for graph semi-supervised classification tasks. However, the drawbacks of their approach are that it requires the full graph Laplacian to be calculated and the output embeddings of a node in each layer are dependent on all its neighbors at the previous layer.

Most recently [8] introduced GraphSage; a node embedding algorithm that uses neural networks to learn embeddings for nodes in the graph structure. Their main contribution is that they solve the limitations mentioned above and show how to aggregate information from a node's neighborhood. Their method consists of two main phases:

(1) **Defining the computation graph and training the neural nets**
The structure of a node's neighborhood will define the computation graph for training the neural networks. In this phase, the objective is to build neural networks that will

ensure nodes close to each other have similar embeddings while nodes far from one another have different embeddings.

(2) **Propagation** For each node, the information of its neighbors is aggregated and passed through the neural networks trained in the first phase.

3 DATASET

There are several well-known datasets collected by different research groups specifically for bot detection on Twitter. Lee et al. [12] provide a social honeypot dataset that contains approximately 22000 content polluters. They have gathered the accounts' metadata and tweets of each account. However, in their released dataset they have anonymized the Twitter account ids. Therefore collecting further information is not possible. Cresci et al. have worked on different Twitter datasets in [3] and by using a crowdsourcing platform they labeled the different types of accounts. [16] released the twitter ids of the accounts they detected as spambots.

Yang et al. 2013 collected Twitter spammers and their dataset contains each account's followers and followings. To the best of our knowledge, this is the dataset we found which has gathered this information for the Twitter accounts. The authors of that work have kindly shared their dataset and we have used the dataset in the present paper. The dataset contains 11000 nodes and 2342816 edges between them.

Table 1 shows the statistics of the dataset used in this paper. The age, tweets and neighbors columns indicate the average amount in each group. The age column is the average age of the accounts reported in days and normalized by setting the oldest day as the first day. The majority of edges between nodes are user to user connections. However, around 5.4% of the edge relations include bot accounts.

	Accounts	Age	Tweets	Neighbors
bots	1000	3023.80	220.90	1963.84
users	10000	3174.28	4658.52	21579.76
relation	bot-bot	bot-user	user-bot	user-user
	2673	73363	50153	2216627
	0.11%	3.13%	2.14%	94.61%

Table 1: Dataset statistics

Figure 1 shows the degree distribution of the accounts in the dataset. Most accounts have a small number of followers and followings and there are a few accounts which have more than 1000 accounts in their neighborhood.

Figure 2.a shows the age and the length of user account name for both bots and users accounts. As shown in figure 2.b and reported in previous work [6] bot accounts have smaller age meaning they were created more recently compared to user accounts. Also as [13] indicated there is no significant difference in length of the accounts name.

4 METHODOLOGY

We used an inductive representation learning approach similar to [8, 9] for detecting twitter bot accounts.

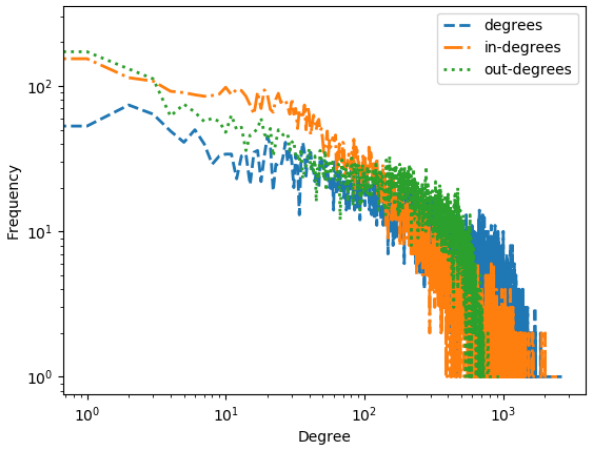


Figure 1: The degree distribution of the nodes in graph. The figure is drawn in log-log scale.

4.1 Problem definition

Let $G = (V, E)$ be a graph where for each $v \in V$ exists a feature vector X_v and a binary label $y \in \{0, 1\}$ associated with it. The goal is to find an embedding vector h_v for each node $v \in V$ such that $f(h_v)$ predicts the label of the node in the graph.

Similar to convolution filters in image processing, graph convolutional networks consider the attributes of a node's neighbors as a representation for that node. Let us define k as the depth of the neighbors of a node from which information is aggregated. If k=1 only the information from its own neighbors will be considered. For k=2 the information is gathered also from the neighbors of its neighbors and so on. The output h_v^k at each depth is calculated as follows:

$$h_{N(v)}^k = mean(\{h_u^{k-1}, \forall u \in N(v)\}) \quad (1)$$

$$h_v^k = f^k(h_v^{k-1}, h_{N(v)}^k) = \sigma(W^k \cdot concat(h_v^{k-1}, h_{N(v)}^k)) \quad (2)$$

Where $h_{N(v)}^k$ is the average of the embedding vectors from v's neighbors. h_v^k is the output which is concatenated with v's previous embedding.

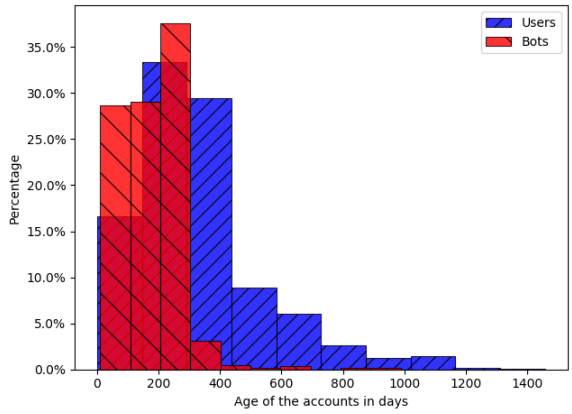
The neural networks are optimized based on the cross-entropy loss function:

$$J(f^k(h_v^{k-1}, h_{N(v)}^k), y) = \\ -\sum_{v \in V} y \log(f(X_v)) + (1 - y) \log(1 - f(X_v)) \quad (3)$$

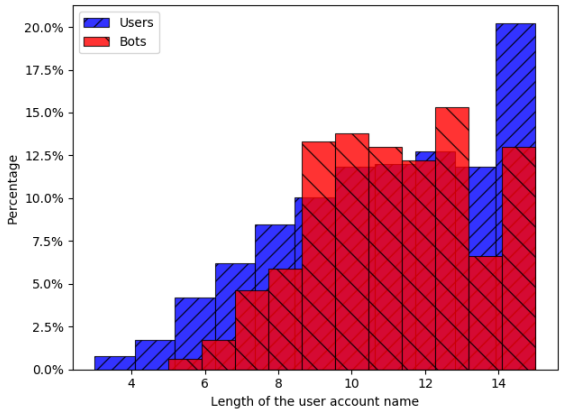
4.2 Features

The initial vector (X_v) for each user consists of the features that can be retrieved directly from the Twitter API[1]. The feature vector consists of :

[1]https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object.html.



Age (a)



Account Length Name (b)

Figure 2: Bots(red) and Users(blue) attributes

	Feature Name	Description
1	age	The created_at attribute returns the datetime that an account was created on Twitter. The age feature is computed by the number of days from the created_at date.
2	favourites_count	This feature indicates the number of tweets a user has liked.
3	statuses_count	The number of tweets including the retweets a user has posted.
4	account length name	The length of an account's name
5	followers_count	followers_count shows the number of follower an accounts has.
6	friends_count	The friends_count attribute shows the number of accounts the user is following.

Table 2: Features

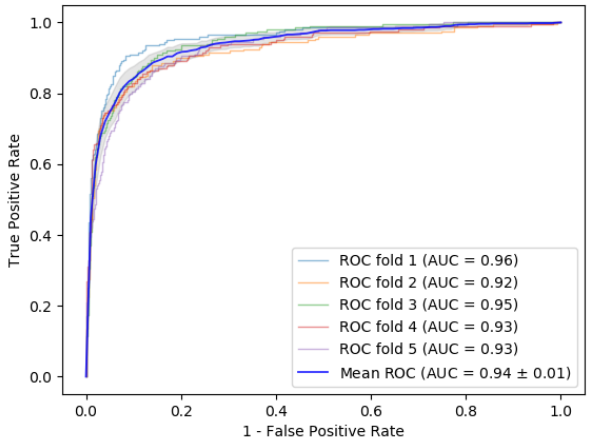


Figure 3: ROC curve over 5-fold cross-validation

5 EVALUATION

In this section, we evaluate the performance of our approach. We conducted a 5-fold cross-validation on the dataset to evaluate the accuracy of the model. Figure 3 shows the area under curve for each fold. On average the GCNN has 0.94 accuracy measured by the area under roc curve.

We measured the precision, recall and f1 metrics as shown in Table 5 for the evaluation. Choosing a meaningful evaluation metric for the classification task is important. For example, it is possible to use the precision measure defined in equation 4 to evaluate the performance of a model. In this case, the measures are calculated over all the data disregarding the class labels.

$$Precision_{micro} = \frac{\sum_c TP}{\sum_c TP + \sum_c FP} \tag{4}$$

$$Recall_{micro} = \frac{\sum_c TP}{\sum_c TP + \sum_c FN} \tag{5}$$

$$f1_{micro} = \frac{2 * Precision_{micro} * Recall_{micro}}{Precision_{micro} + Recall_{micro}} \tag{6}$$

However, by this definition for a dataset where the majority of labels belong to one class, the precision score remains high even if the model has not detected the labels of the other class correctly. Therefore, for a better evaluation of the model, we compute the precision, recall, f1 score for each class separately and report the average score on the two classes. This is also known as macro score in the *scikit-learn* python library [15].

$$Precision_{macro} = \frac{1}{|c|} \sum_c \frac{TP}{TP + FP} \tag{7}$$

$$Recall_{macro} = \frac{1}{|c|} \sum_c \frac{TP}{TP + FN} \tag{8}$$

$$f1_{macro} = \frac{2 * Precision_{macro} * Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{9}$$

$\psi_{ij}(x_i, x_j)$	$x_j = user$	$x_j = bot$
$x_i = user$	$0.5 + w\epsilon$	$0.5 - w\epsilon$
$x_i = bot$	$0.5 - w\epsilon$	$0.5 + w\epsilon$

Table 3: Edge potentials matrices

Node	$P(user)$	$P(bot)$
User	0.99	0.01
Bot	0.01	0.99
Unknown/Validation	0.5	0.5

Table 4: Node potential based on the original state

5.1 Comparsion with MLP and Belief Propagation

We further evaluated our approach by comparing it with two other methods. As graph convolutional neural networks take both the feature set and the graph structure into consideration, we demonstrate the performance of this method by comparing it with multi layer perceptron (MLP) and belief propagation (BP).

The MLP classifier is trained based on the feature set defined in the Features section. The input layer takes the feature vectors normalized to values between 0 and 1 for each account. The hidden layers consist of two layers with 25 and 10 neurons respectively and use a rectified linear unit as the transfer function. The log loss function is optimized using stochastic gradient descent with a learning rate of 0.0001.

On the other hand, the Belief Propagation algorithm runs solely on the graph structure. The Belief Propagation (BP) algorithm originally proposed by Judea Pearl [14] infers a node's label from some prior knowledge about that node and other neighboring nodes by iteratively passing messages between all pairs of nodes in the graph. The message sent indicates nodes' beliefs regarding the state of their neighbours. For details please refer to [19]. In this experiments we adopted the original BP with the node and edge potential metrics indicated in Table 3 and 4. Furthermore, we ran the experiment with 7 iterations as the messages passed across nodes had no significant changes after 7 iterations.

We plotted the Receiver Operating Characteristic (ROC) for the different models as shown in figure 4. We observe that the area under the ROC curve is 94% for the GCNN approach which is 8% and 16% percent higher than the MLP and BP approach respectively.

While neural networks have shown to perform well in various domains, they are often considered as black boxes when it comes to why they result in such outputs. Interpreting each entry of the output and the meaning of the embedding vectors generated remains open question and topic to investigate for future research.

6 CONCLUSION AND FUTURE WORK

In this paper, we have examined a new approach for detecting malicious accounts and social bots on Twitter by using graph convolutional networks. The main idea of our method is to employ the graph structure and relationships of Twitter accounts for classifying the accounts. Each account aggregates the feature information from its neighborhood. To demonstrate the efficacy of our proposal,

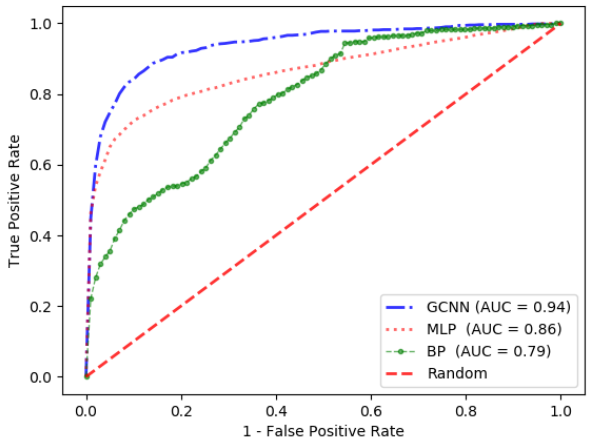


Figure 4: Comparison of the area under curve of different algorithms.

we have worked on a previous well-known dataset in bot detection. Results show that our approach outperforms the state of the art classification algorithms with 8% improvement in the area under curve accuracy.

Since the Twitter API has a limit of 15 requests per rate limit window every 15 minutes, building the Twitter graph structure based on the follower and friend relation of the accounts is not an easy task. We are aware this may be considered as a limitation to our approach. It can thus be suggested to build the graph structure based on the retweet graph of user accounts. Finally, a specific extension for future work is to deploy this method in real time on Twitter's streaming API for spambot detection.

ACKNOWLEDGMENTS

The authors would like to thank the HPI Future SOC Lab for providing access to the resources during the period of Fall 2018.

REFERENCES

[1] N. Chavoshi, H. Hamooni, and A. Mueen. 2016. DeBot: Twitter Bot Detection via Warped Correlation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 817–822. https://doi.org/10.1109/ICDM.2016.0096

[2] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2017. Temporal Patterns in Bot Activities. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1601–1606. https://doi.org/10.1145/3041021.3051114

[3] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. 963–972. https://doi.org/10.1145/3041021.3055135

[4] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi. 2016. DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection. *IEEE Intelligent Systems* 31, 5 (Sept 2016), 58–64. https://doi.org/10.1109/MIS.2016.29

[5] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. BotOrNot: A System to Evaluate Social Bots. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*. International World Wide Web Conferences Steering Committee, 273–274. https://doi.org/10.1145/2872518.2889302

	$Precision_{macro}$	$Recall_{macro}$	$f1_{macro}$
MLP	0.81	0.73	0.77
BP	0.56	0.54	0.55
GCNN (*with features* 1, 2, 3, 4)	0.85	0.77	0.80
GCNN (*with features* 5, 6)	0.80	0.69	0.72
GCNN (*All features*)	0.89	0.80	0.84

Table 5: Comparison of different algorithms on the dataset

[6] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The Rise of Social Bots. *Commun. ACM* 59, 7 (June 2016), 96–104. https://doi.org/10.1145/2818717

[7] Palash Goyal, Homa Hosseinmardi, Emilio Ferrara, and Aram Galstyan. 2018. Embedding Networks with Edge Attributes. In *Proceedings of the 29th on Hypertext and Social Media (HT '18)*. ACM, New York, NY, USA, 38–42. https://doi.org/10.1145/3209542.3209571

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1024–1034. http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf

[9] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *CoRR* abs/1709.05584 (2017). arXiv:1709.05584 http://arxiv.org/abs/1709.05584

[10] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[11] Srijan Kumar and Neil Shah. 2018. False Information on Web and Social Media: A Survey. *CoRR* abs/1804.08559 (2018). arXiv:1804.08559 http://arxiv.org/abs/1804.08559

[12] Kyumin Lee, Brian David Eoff, and James Caverlee. 2011. Seven months with the devils: a long-term study of content polluters on Twitter. In *In AAAI IntâĂŹl Conference on Weblogs and Social Media (ICWSM)*.

[13] Mehwish Nasim, Andrew Nguyen, Nick Lothian, Robert Cope, and Lewis Mitchell. 2018. Real-time Detection of Content Polluters in Partially Observable Twitter Networks. In *Companion Proceedings of the The Web Conference 2018 (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1331–1339. https://doi.org/10.1145/3184558.3191574

[14] Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[16] Onur Varol, Emilio Ferrara, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*. AAAI Press.

[17] Chao Yang, Robert Harkreader, and Guofei Gu. 2013. Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE Transactions on Information Forensics and Security* (2013).

[18] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. 2012. Analyzing Spammers' Social Networks for Fun and Profit: A Case Study of Cyber Criminal Ecosystem on Twitter. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 71–80. https://doi.org/10.1145/2187836.2187847

[19] Jonathan S Yedidia, William T Freeman, and Yair Weiss. 2003. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8 (2003), 236–239.