DCGNN: Dual-Channel Graph Neural Network for Social Bot Detection

Nuoyan Lyu*
Institute of Computing Technology, University of Chinese Academy of Sciences
Beijing, China
nuoyanlyu@gmail.com

Bingbing Xu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
xubingbing@ict.ac.cn

Fangda Guo*
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
guofangda@ict.ac.cn

Huawei Shen
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
shenhuawei@ict.ac.cn

ABSTRACT

The importance of social bot detection has been increasingly recognized due to its profound impact on information dissemination. Existing methodologies can be categorized into feature engineering and deep learning-based methods, which mainly focus on static features, e.g., post characteristics and user profiles. However, existing methods often overlook the burst phenomena when distinguishing social bots and genuine users, i.e, the sudden and intense activity or behavior of bots after prolonged inter. Through comprehensive analysis, we find that both burst behavior and static features play pivotal roles in social bot detection. To capture such properties, the dual-channel GNN (DCGNN) is proposed which consists of a burst-aware channel with an adaptive-pass filter and a static-aware channel with a low-pass filter to model user characteristics effectively. Experimental results demonstrate the superiority of this method over competitive baselines.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → *Social networks.*

KEYWORDS

social bot detection; burst aware; dual channel graph neural network

ACM Reference Format:
Nuoyan Lyu, Bingbing Xu, Fangda Guo, and Huawei Shen. 2023. DCGNN: Dual-Channel Graph Neural Network for Social Bot Detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615237

1 INTRODUCTION

Nowadays, social media has undeniably emerged as the primary platform for individuals to spread information and acquire news [24, 30, 31]. However, the increasing prevalence of social bots [6, 15], which are accounts on social media operated by programs and pretending to be genuine users [16, 18, 26], significantly disrupts the order of information dissemination [5, 46] and manipulates public opinion [17, 33, 34, 38]. Effective methodologies for social bot detection holds substantial significance for society [44].

Existing methodologies can be categorized into feature engineering and deep learning-based methods [7]. Early studies predominantly focused on designing well-defined features [9, 22, 43]. With the advancement of deep learning, new methodologies have been proposed, including utilizing natural language processing (NLP) [11, 37, 40], employing generative adversarial network (GAN) models [8, 10, 29, 39], and leveraging graph neural network (GNN). Notably, GNN-based methods can capture the inherent social structure naturally existing in social media [2, 20]. For example, Feng *et al.* [14] proposed BotRGCN which considered various features and deployed relation graph convolution neural network (R-GCN) [32]. However, the aforementioned methods are heuristic in nature, capturing information without a comprehensive analysis of the distinctions between bots and genuine users.

Through comprehensive data analysis on a representative dataset, we have uncovered a significant distinction between bots and genuine users that has been largely overlooked by existing methodologies, namely the burst phenomenon that occurs during the posting process. As depicted in Fig. 1, social bots exhibit a preference for sporadic activity, while genuine users do not exhibit such behavior. We refer to this phenomenon as "burst", specifically denoting the sudden and intense activity or behavior after prolonged intervals. Such a phenomenon is reasonable since bots tend to be active only when they are engaged or prompted. Meanwhile, our data analysis reveals the significance of static account characteristics obtained from account profile information in social bot detection.

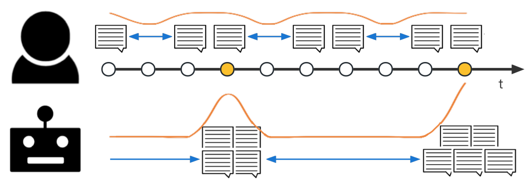

Figure 1: Burst occurs during the posting process. Genuine users tend to tweet frequently or even daily, while bots tend to post at longer intervals, as yellow lines indicate.

*Corresponding authors

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615237

For the purpose of capturing both burst and static properties, we propose Dual-Channel Graph Neural Network, namely DCGNN to detect social bots. DCGNN consists of the burst-aware channel and the static-aware channel. The former employs an adaptive-pass filter on a specific graph structure to construct users' burst property, while the latter utilizes a low-pass filter to extract users' static features. This approach comprehensively considers account features through the integration of dual perspectives. Experimental results demonstrate the superiority of DCGNN over existing methods.

2 PRELIMINARY

We introduce data analysis and problem formulation in this section.

2.1 Data Analysis

To gain deeper insights into social bots, we perform extensive data analysis on a comprehensive and reliable dataset. Our analysis reveals that both burst features and static account features play critical roles in differentiating between genuine users and bots.

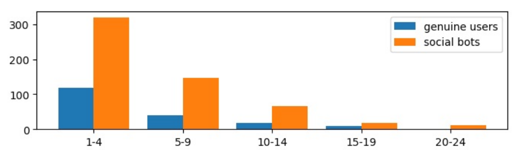

Figure 2: burst counts distribution of bots and genuine users.

2.1.1 *Burst Feature Analysis.*

In terms of account behavior, social bots may only be active when they are required to perform a role, such as controlling public opinion or promoting their agenda. This pattern results in social bot accounts featuring "frequent activity after a long time interval", which we define as a burst phenomenon. To verify this notion, we sample a group of users and count the number of bursts, assuming a release interval of 20 days as a burst. We find that social bots have an average of 2.03 bursts, while genuine users have an average of 0.618 bursts. As shown in Fig. 2, we found that social bots have more bursts than genuine users. This result proves that using bursts to detect social bots is effective.

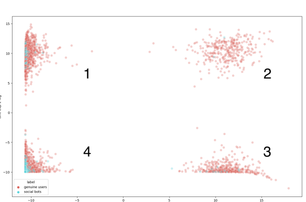

Figure 3: T-SNE results of user static features.

Cat.	Bots	Genuine
cat. 1	2.8%	18.9%
cat. 2	0.2%	3.9%
cat. 3	1.0%	4.5%
cat. 4	96%	72.7%

Table 1: Accounts' proportions in each category

2.1.2 *Static Feature Analysis.*

We randomly select 10,000 users and analyze their account features, which are considered static as they are derived from profile information and remain unchanged with user behavior. Using the t-SNE clustering algorithm [36], we visualize these features and observe that the accounts can be categorized into four distinct groups, as depicted in Fig. 3. We then compute the number of accounts and their proportions within each category that are shown in table 1. Notably, while all accounts are concentrated in the fourth category, social bots exhibit a significantly higher concentration of 96% in this particular category. This finding underscores the importance of static account features and emphasizes the need to capture them to enhance the accuracy of bot detection.

2.2 Problem Formulation

Suppose that we have N_u users and each user has a property set $U = \{U^{num}, U^{cat}, U^{des}\}$ representing the user's numeric properties, categorical properties, and account description. There are N_t corresponding tweets posted by these users and each tweet owns a property set $T = \{T^{num}, T^{cat}, T^{text}\}$, where T^{num} describes the numerical properties, T^{cat} describes the categorical properties, and T^{text} describes the content of the tweet.

Meanwhile, there are many relations between users and tweets. Let $R = \{R^{follower}, R^{following}, R^{post}, R^{reply}, R^{quote}, R^{retweet}\}$ be the relation set, which denotes the follower and following relations between users, the post relation between users and tweets, and the reply, quote, retweet relations between tweets. The task of social bot detection is to identify bots among users using the user, tweet, and relation information U, T, and R.

3 METHOD

Based on our above observations, we propose dual-channel GNN which consists of two modules: burst-aware channel and static-aware channel. The overall architecture is illustrated in Fig. 4. In the following, we introduce the details of each module.

3.1 Burst-Aware Channel

We propose to use the adaptive-pass filter to capture the burst phenomenon. In the following, we first introduce our motivation for leveraging the adaptive-pass filter and then the detailed design.

3.1.1 *Motivation to use adaptive-pass filters.*

We compute the properties of all the tweets posted by the user u_i in time slot t_j and aggregate them to describe the post behavior:

$$x_{ij} = [x_{ij}^{num}; x_{ij}^{cat}] \tag{1}$$

By connecting nodes from adjacent time slots, a user with burst behavior is described as a sequence of empty nodes followed by a node with a higher concentration of tweets. Conversely, a user without burst behavior is depicted by nodes without significant fluctuations and a proper filter should be employed to distinguish between them. To build a graph structure that can aware burst, we introduce two kinds of edges, one is based on timing relationships, connecting the same user at different time steps, and the other is based on replies, quotes, and retweets to improve the connectivity.

After constructing the post-behavior graph, the main difference between users with burst and those without it lies in the significance of feature variations on the graph. Adaptive-pass filters, which retain high-frequency and non-smooth features while filtering out low-frequency and smooth features, exhibit enhanced capability.

To provide more evidence for the above analysis, we focus on the difference between high-pass filters and low-pass filters. Given a

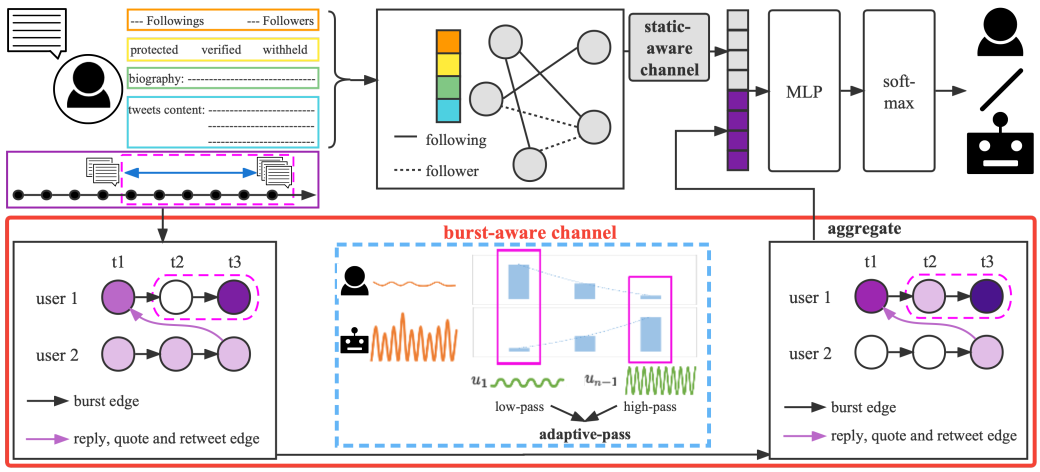

Figure 4: Architecture overview of DCGNN, which includes burst-aware channel and static-aware channel.

graph $G = (V, E)$ with adjacency matrix A, we can define its normalized Laplacian matrix as $L = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ where $D_{ii} = \sum_j A_{ij}$. The matrix L possesses a complete set of orthonormal eigenvectors $U = [u_1, u_2, ..., u_n]$.For an eigenvector u_i, its associated eigenvalue λ_i captures the smoothness, and eigenvectors corresponding to small eigenvalues exhibit smoothness in relation to the graph structure [35, 42]. Since eigenvectors are orthonormal, given a signal x defined on this graph, it can be uniquely represented by U as:

$$x = \alpha_1 u_1 + \alpha_2 u_2 + ... + \alpha_n u_n \tag{2}$$

and $u_i u_i^T$ composes a set of basic filters:

$$u_i u_i^T x = \alpha_1 u_i u_i^T u_1 + \cdots + \alpha_n u_i u_i^T u_n = \alpha_i u_i \tag{3}$$

Then various graph neural network methods employ distinct weight assignments for these filters.

As depicted in Fig. 4, human signals with few bursts consist mainly of low-frequency components, which are predominantly preserved by the low-pass filter. Conversely, bot signals with more bursts are primarily captured by the high-pass filter[23, 45]. To distinguish between these two types of signals, we utilize adaptive-pass filters that amplify low-frequency and high-frequency signals while inhibiting mid-frequency signals [41].

3.1.2 Implementation of Adaptive-Pass Filters. Therefore, we apply the Frequency Adaptation Graph Convolutional Networks (FAGCN) [4] to extract burst features. First, we map the initial post feature to hidden space:

$$x_{ij}^{(0)} = \phi(W_1 \cdot x_{ij} + b_1) \tag{4}$$

where W_1 and b_1 are learnable parameters. ϕ represents the activate function and we adopt leaky-relu as ϕ for the rest of the paper. Then we apply the convolutional layer of FAGCN:

$$x_{ij}^{(l)} = \epsilon x_{ij}^{(0)} + \sum_{k \in N_{ij}} \frac{\alpha_{ij,k}}{\sqrt{d_{ij} d_k}} x_{ij}^{(l-1)} \tag{5}$$

where ϵ is the initial layers weight defined in FAConv which we set to the default value 0.1 and $\alpha_{ij,k}$ is the learnable attention weight between node ij and its neighbor k. The representations of the same user from different time slots are aggregated to obtain the final burst feature. Specifically, we compute the absolute value to obtain the magnitude of the signal:

$$x_i = \sum_j |x_{ij}| \tag{6}$$

where x_i is the aggregated feature of user u_i. Then it is fed into a one-layer MLP to get the final user burst feature:

$$x_{i\_burst} = \phi(W_2 x_i + b_2) \tag{7}$$

where W_2 and b_2 are learnable parameters and x_{i_burst} is the final burst feature of user u_i.

3.2 Static-Aware Channel

To introduce the static feature into the model, we follow BotRGCN [14], which is the state-of-art social bot detection method based on GNN. The initial user static feature is computed by:

$$r_i = [r_{b,i}; r_{t,i}; r_{p,i}^{num}; r_{p,i}^{cat}] \tag{8}$$

where r_i, $r_{b,i}$, $r_{t,i}$, $r_{p,i}^{num}$ and $r_{p,i}^{cat}$ correspond to the initial static features, the descriptions, the tweets, the numerical properties, and the categorical properties of user u_i, respectively. $r_{p,i}^{num}$ and $r_{p,i}^{cat}$ are directly from user property set U, while $r_{b,i}$ and $r_{t,i}$ are processed by pretrained RoBERTa [25].

First the features are transformed to derive the hidden vectors:

$$y_i^{(0)} = \phi(W_3 \cdot r_i + b_3) \tag{9}$$

where W_3 and b_3 are learnable parameters. Next we apply the R-GCN layers to aggregate feature in following and follower relations:

$$y_i^{l+1} = \Theta_{self} y_i^l + \sum_{r \in R} \sum_{j \in N_r(i)} \frac{1}{|N_r(i)|} \Theta_r y_i^l \tag{10}$$

where Θ is the projection matrix. After L layers we use MLP to get the final user static feature:

$$x_{i\_static} = \phi(W_4 y_i^L + b_4) \tag{11}$$

where W_4 and b_4 are learnable parameters. Finally, the burst and static features are concatenated to get the final representation:

$$h_i = \phi(W_5 \cdot [x_{i\_burst}; x_{i\_static}] + b_5) \quad (12)$$

where W_5 and b_5 are learnable parameters.

3.3 Loss Function

We apply a softmax layer to conduct social bot detection task based on the final user representations:

$$\hat{y_i} = softmax(W_o \cdot h_i + b_o) \quad (13)$$

where W_o and b_o are learnable parameters. We introduce the weighted cross entropy loss to address the imbalanced classification:

$$L = -\sum_{i \in Y} [\omega_+ y_i log(\hat{y_i}) + \omega_- (1 - y_i) log(1 - \hat{y_i})] + \lambda \sum_{\omega \in \theta} \omega^2 \quad (14)$$

where Y is the users with label in the dataset, y_i is the ground-truth label, θ is all learnable parameters in the model, and ω_+ and ω_- are the weights added to positive class and negative class.

4 EXPERIMENTS

To validate the proposed DCGNN, we show the experimental results of our model applied to representative TwiBot-22 dataset[13].

4.1 Experimental Setup

4.1.1 Dataset. We use TwiBot-22 [13] dataset that contains both the tweet information and various relations between users and tweets to validate DCGNN. Meanwhile, social bots are about 13%, which meets real-world scenarios. We random extract a subgraph of the raw data by applying a Louvain Community detection algorithm [3] on the following relationship among users, which involves 8694 accounts with about 11% social bots. We follow the partition of training, validation, and test set in the original benchmark.

4.1.2 Baselines. We leverage the following various baselines:

- Miller *et al.* [27] who extract 107 features and defines the social bot detection as anomaly detection.
- Kudugunta *et al.* [21] who use both account metadata and the tweet content.
- Efthimion *et al.* [12] who leverage text variation and apply the Support Vector Machine (SVM) algorithm.
- Moghaddam *et al.* [28] who use graph structure and attributes to conduct the random forest algorithm.
- Ali *et al.* [1] who select a subset of features and apply GNN.
- BotRGCN [14] which uses multiple static features and aggregates features by RGCN.

4.1.3 Evaluation Metric. We choose commonly used evaluation metrics to assess the performance, including accuracy, precision, recall, and f1-score. Considering the imbalanced nature of the dataset, we prioritize the f1-score rather than accuracy.

4.2 Performance on Social Bot Detection Task

Table 2 clearly shows the superior performance of our method. Among the evaluated methods, Miller's approach exhibits superior recall but lower precision, suggesting an inclination towards over-classifying accounts as bots. Ali's method performs the highest accuracy and precision but falls short in terms of recall, indicating its limited ability to accurately identify bots among all accounts. In contrast, our method achieves the most favorable overall performance, as reflected by the f1 score, thus underscoring its efficacy.

Table 2: Performance on TwiBot-22 benchmark

Method	F1-Score	Precision	Recall	Accuracy
Kudugunta	0.4089	0.3049	0.6209	0.6083
Miller	0.3733	0.2399	**0.8405**	0.3842
Efthimion	0.2203	0.3562	0.1595	0.7537
Moghaddam	0.4058	0.4171	0.3951	0.7475
Ali	0.1939	**0.5758**	0.1166	**0.7885**
BotRGCN	0.3194	0.4200	0.2577	0.7604
DCGNN	**0.4713**	0.4653	0.4785	0.7657

4.3 Ablation studies

To demonstrate the effectiveness of dual channels, we conduct ablation studies, including using single-channel detection and changing the adaptive-pass filter to a typical low-pass one.

Table 3: Ablation studies with DCGNN

Module	F1-Score	Precision	Recall	Accuracy
BAC	0.3389	0.2190	0.7485	0.3628
SAC	0.3650	0.3739	0.2638	0.7430
low-pass	0.3094	0.3739	0.2638	0.7430
DCGNN	0.4713	0.4653	0.4785	0.7657

We conduct bot detection with a single burst-aware channel (BAC) and a single static-aware channel (SAC). As shown in table 3, both channels could complete the detection task independently, yet their performance falls short of the complete module. Notably, the BAC exhibits superior performance in terms of the recall metric, while the SAC demonstrates better precision results. This suggests that these channels focus on different aspects, emphasizing the significance of considering both. Furthermore, we validate the effectiveness of adaptive-pass filters in the BAC by replacing FAGCN with a typical low-pass filter GCN [19]. The results shown in table 3 demonstrate the effectiveness of adaptive-pass filters.

5 CONCLUSION

We perform comprehensive data analysis on a representative dataset to identify social bots and underscore the significance of burst phenomena. Therefore, DCGNN is proposed to incorporate burst features and static features. Extensive experiments on the TwiBot-22 dataset demonstrate the superiority of our method compared to competitive baselines.

6 ACKNOWLEDGEMENTS

The work was supported by National Natural Science Foundation of China (Grant No.U21B2046, No.62202448) and China Postdoctoral Science Foundation (2022M713208, 2022M713206).

REFERENCES

[1] 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*. 148–153.

[2] Samy Benslimane, Jérôme Azé, Sandra Bringay, Maximilien Servajean, and Caroline Mollevi. 2023. A text and GNN based controversy detection method on social media. *World Wide Web* 26, 2 (2023), 799–825.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[4] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3950–3957.

[5] Chun Cheng, Yun Luo, and Changbin Yu. 2020. Dynamic mechanism of social bots interfering with public opinion in network. *Physica A: statistical mechanics and its applications* 551 (2020), 124163.

[6] Stefano Cresci. 2020. A decade of social bot detection. *Commun. ACM* 63, 10 (2020), 72–83.

[7] Stefano Cresci. 2020. Detecting malicious social bots: story of a never-ending clash. In *Disinformation in Open Online Media: First Multidisciplinary International Symposium, MISDOOM 2019, Hamburg, Germany, February 27–March 1, 2019, Revised Selected Papers 1*. Springer, 77–88.

[8] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2021. The coming age of adversarial social bot detection. *First Monday* (2021).

[9] 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*. 273–274.

[10] George Dialektakis, Ilias Dimitriadis, and Athena Vakali. 2022. CALEB: A Conditional Adversarial Learning Framework to Enhance Bot Detection. *arXiv preprint arXiv:2205.15707* (2022).

[11] John P Dickerson, Vadim Kagan, and VS Subrahmanian. 2014. Using sentiment to detect bots on twitter: Are humans more opinionated than bots?. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. IEEE, 620–627.

[12] Phillip George Efthimion, Scott Payne, and Nicholas Proferes. 2018. Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Science Review* 1, 2 (2018), 5.

[13] Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al. 2022. TwiBot-22: Towards graph-based Twitter bot detection. *arXiv preprint arXiv:2206.04564* (2022).

[14] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021. BotRGCN: Twitter bot detection with relational graph convolutional networks. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 236–239.

[15] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (2016), 96–104.

[16] Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann. 2017. Social bots: Human-like by means of human control? *Big data* 5, 4 (2017), 279–293.

[17] Loni Hagen, Stephen Neely, Thomas E Keller, Ryan Scharf, and Fatima Espinoza Vasquez. 2022. Rise of the machines? Examining the influence of social bots on a political discussion network. *Social Science Computer Review* 40, 2 (2022), 264–287.

[18] Arzum Karataş and Serap Şahin. 2017. A review on social bot detection techniques and research directions. (2017).

[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[20] David Knoke and Song Yang. 2019. *Social network analysis*. SAGE publications.

[21] Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences* 467 (2018), 312–322.

[22] Kyumin Lee, Brian Eoff, and James Caverlee. 2011. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the international AAAI conference on web and social media*, Vol. 5. 185–192.

[23] Shouheng Li, Dongwoo Kim, and Qing Wang. 2021. Beyond low-pass filters: Adaptive feature propagation on graphs. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*. Springer, 450–465.

[24] Shu-Hsien Liao, Retno Widowati, and Yu-Chieh Hsieh. 2021. Investigating online social media users' behaviors for social commerce recommendations. *Technology in Society* 66 (2021), 101655.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[26] Nathalie Maréchal. 2016. Automation, algorithms, and politics| when bots tweet: Toward a normative framework for bots on social networking sites (feature). *International Journal of Communication* 10 (2016), 10.

[27] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. 2014. Twitter spammer detection using data stream clustering. *Information Sciences* 260 (2014), 64–73.

[28] Samaneh Hosseini Moghaddam and Maghsoud Abbaspour. 2022. Friendship Preference: Scalable and Robust Category of Features for Social Bot Detection. *IEEE Transactions on Dependable and Secure Computing* (2022).

[29] Shaghayegh Najari, Mostafa Salehi, and Reza Farahbakhsh. 2022. GANBOT: a GAN-based framework for social bot detection. *Social Network Analysis and Mining* 12 (2022), 1–11.

[30] Esteban Ortiz-Ospina and Max Roser. 2023. The rise of social media. *Our world in data* (2023).

[31] Neha Puri, Eric A Coomes, Hourmazd Haghbayan, and Keith Gunaratne. 2020. Social media and vaccine hesitancy: new updates for the era of COVID-19 and globalized infectious diseases. *Human vaccines & immunotherapeutics* 16, 11 (2020), 2586–2593.

[32] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 593–607.

[33] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. 2017. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592* 96 (2017), 104.

[34] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications* 9, 1 (2018), 1–9.

[35] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[37] Feng Wei and Uyen Trang Nguyen. 2019. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA)*. IEEE, 101–109.

[38] Samuel C Woolley. 2016. Automating power: Social bot interference in global politics. *First Monday* (2016).

[39] Bin Wu, Le Liu, Yanqing Yang, Kangfeng Zheng, and Xiujuan Wang. 2020. Using improved conditional generative adversarial networks to detect social bots on Twitter. *IEEE Access* 8 (2020), 36664–36680.

[40] Yuhao Wu, Yuzhou Fang, Shuaikang Shang, Jing Jin, Lai Wei, and Haizhou Wang. 2021. A novel framework for detecting social bots with deep neural networks and active learning. *Knowledge-Based Systems* 211 (2021), 106525.

[41] Xuanting Xie, Wenyu Chen, Zhao Kang, and Chong Peng. 2023. Contrastive graph clustering with adaptive filter. *Expert Systems with Applications* 219 (2023), 119645.

[42] Bingbing Xu, Huawei Shen, Qi Cao, Keting Cen, and Xueqi Cheng. 2020. Graph convolutional networks using heat kernel for semi-supervised learning. *arXiv preprint arXiv:2007.16002* (2020).

[43] 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* 8, 8 (2013), 1280–1293.

[44] Kai-Cheng Yang, Onur Varol, Clayton A Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies* 1, 1 (2019), 48–61.

[45] Wenhan Yang and Baharan Mirzasoleiman. 2023. Contrastive Learning under Heterophily. *arXiv preprint arXiv:2303.06344* (2023).

[46] Mattia Zago, Pantaleone Nespoli, Dimitrios Papamartzivanos, Manuel Gil Perez, Felix Gomez Marmol, Georgios Kambourakis, and Gregorio Martinez Perez. 2019. Screening out social bots interference: Are there any silver bullets? *IEEE Communications Magazine* 57, 8 (2019), 98–104.