Faculty of Mathematics
and Computer Science

Social Network Analysis course (SNA)

Bot Detection in Social Graphs Using Network-Based Features

Murariu Tudor Cristian

Department of Computer Science, Babeş-Bolyai University
1, M. Kogalniceanu Street, 400084, Cluj-Napoca, Romania

Abstract

A concise summary highlighting the significance of bot detection in social networks, the role of network-based features, and the key findings from recent studies.

Keywords:

1. Introduction

Social media platforms have become critical infrastructures for communication, information sharing, and social interaction. However, their growing influence has also attracted malicious actors—particularly social bots. These automated accounts are designed to mimic human behavior and can be used to manipulate public discourse, spread misinformation, inflate follower counts, or promote content for political and commercial gain.

As bots become increasingly sophisticated, the challenge of detecting them has grown more complex. Traditional bot detection methods often rely on content-based analysis or behavioral signals such as posting frequency or linguistic features. While effective in some scenarios, these methods can be bypassed by bots that carefully emulate human-like content and interaction patterns.

In contrast, network-based bot detection relies on the structure of user interactions within social graphs. Users are represented as nodes, and their connections (e.g., follows, mentions) as edges. Bots often display unusual patterns—such as high out-degree or low clustering—which can be detected using Social Network Analysis (SNA).

This paper reviews recent methods that leverage these structural features, compares detection models, and discusses key challenges and future directions.

2. Fundamentals of Social Network Analysis

Social Network Analysis (SNA) provides the mathematical and conceptual framework to study relationships among entities in a network. In the context of social graphs, **nodes** represent individuals (e.g., users), while **edges** represent interactions or relationships (e.g., following, messaging, retweeting).

Several structural metrics are commonly used in SNA:

- **Degree Centrality**: Measures how many direct connections a node has. Bots often have an unusually high out-degree and low in-degree. [13]
- **Betweenness Centrality**: Reflects how often a node appears on the shortest paths between other nodes, identifying influential users or bridges. [9]
- **Clustering Coefficient**: Indicates how densely a node's neighbors are connected to each other, often lower in bot ego-networks.
- **Eigenvector Centrality**: Extends degree by considering the influence of a node's neighbors. [13]
- **Ego Network Density**: Measures how well-connected a user's immediate neighbors are, often sparse in bot structures.

These metrics allow researchers to detect abnormal interaction patterns typical of automated accounts, without analyzing message content.

3. Review of Existing Methodologies

Numerous methodologies have been proposed for detecting bots in social networks, many of which utilize network-based features either alone or in combination with content and behavioral signals. This section reviews a selection of influential approaches, categorized into *unsupervised*, *supervised*, and *hybrid* methods, with a particular focus on those leveraging the structure of the social graph.

3.1. Unsupervised Approaches

Unsupervised methods aim to detect anomalous patterns without labeled training data. These techniques often involve clustering algorithms, graph partitioning, or anomaly detection based on structural metrics.

- **CopyCatch** by Beutel et al. [4] identifies lockstep behavior in follow or like graphs by detecting dense bipartite subgraphs. It is particularly effective for detecting coordinated bot campaigns.
- Jin et al. [7] proposed a framework based on graph entropy and clustering coefficient to detect anomalies in local structures, which often reflect fake user activity.
- Ahmed and Abulaish [1] used community detection based on modularity and intra-community density to isolate groups of suspicious users on Twitter.

These methods are useful when labeled data is unavailable, but may have higher false positive rates.

3.2. Supervised Learning Methods

Supervised approaches rely on labeled datasets to train classifiers using extracted network features.

- **Botometer** (formerly BotOrNot) developed by Varol et al. [12] combines various features, including follower/followee ratios and retweet networks, in a Random Forest model.
- **DeBot** by Chavoshi et al. [5] clusters bots based on similar temporal behavior, using mutual mention graphs for structural grouping.
- Davis et al. [6] demonstrated that structural metrics such as centrality and clustering coefficients are powerful features in bot classification.

These models offer high accuracy but may not generalize well to unseen bot strategies.

3.3. Hybrid and Graph Learning Approaches

Recent developments incorporate deep learning and graph-based embeddings for scalable detection.

- Kumar et al. [8] explored misinformation spread using graph embeddings to capture structural roles of bots in propagation networks.
- Alhosseini et al. [2] applied **Graph Convolutional Networks (GCNs)** to learn from the graph structure and improve classification of bot accounts.
- Ribeiro et al. [11] introduced **BotSpot**, a model that combines handcrafted features with graph-based embeddings, offering robust cross-platform detection.

These approaches are adaptive and powerful but often less interpretable and more computationally demanding.

3.4. Summary of Techniques

Method	**Type**	**Key Features**	**Reference**
CopyCatch	Unsupervised	Bipartite subgraph detection	[4]
Botometer	Supervised	Degree, reciprocity, centrality	[12]
DeBot	Supervised	Mention graphs, temporal patterns	[5]
GCN Models	Hybrid	Graph embeddings, topology	[2]
BotSpot	Hybrid	Structural + embedding features	[11]

Table 1. Summary of representative methodologies for bot detection.

4. CopyCatch: Detecting Lockstep Behavior in Social Graphs

CopyCatch is a system developed by Beutel et al. [4] to detect groups of accounts engaging in coordinated, malicious activities within social networks. The key innovation lies in identifying *lockstep behavior*, where multiple users perform similar actions (e.g., liking the same pages or adding the same friends) within a short time frame.

The algorithm models user actions in a bipartite graph, where one set of nodes represents users and the other represents targets (such as Facebook pages). An edge is drawn between a user and a target if an interaction occurs. CopyCatch scans this graph for near-complete bipartite cores—dense subgraphs where a group of users interacts with the same set of targets around the same time.

Key Mechanism

CopyCatch operates by:

- **Sliding Time Windows**: Interactions are grouped based on time to detect bursts of coordinated activity.
- **Seed Expansion**: The algorithm begins with a small set of suspicious users (a seed) and incrementally expands it to a larger group if evidence of lockstep behavior increases.
- **Scalable Graph Mining**: Designed to run in near real-time on large-scale social graphs.

Use Cases and Impact

Originally deployed by Facebook, CopyCatch helped detect fake accounts and spam campaigns by identifying large groups of accounts acting in synchrony. Unlike traditional methods that focus on individual account features, CopyCatch emphasizes group behavior—making it harder for malicious actors to evade detection by mimicking real user behavior individually.

Limitations

While powerful, CopyCatch assumes that malicious actions happen in dense, time-coordinated clusters. Sophisticated attackers can potentially evade detection by spacing out actions or randomizing targets. Nevertheless, the method remains one of the foundational approaches to bot and group attack detection using social graph analysis.

5. Botometer: Bot Detection via Multi-Feature Supervised Learning

Botometer, formerly known as BotOrNot, is a widely used tool for detecting bots on Twitter, developed by Varol et al. [12]. Unlike network-only approaches, Botometer evaluates a wide range of features from a user's profile, timeline, and interactions, combining them into a supervised learning framework.

System Overview

Botometer uses over 1,000 features, grouped into several categories:

- **User Features:** Metadata such as account age, number of followers, and tweet frequency.
- **Content Features:** Linguistic features derived from tweets, including sentiment, entropy, and lexical diversity.
- **Network Features:** Measures based on retweet, mention, and follower networks, including centrality and clustering.
- **Temporal Features:** Timing and regularity of activity patterns.
- **Friend Features:** Statistics derived from the accounts a user interacts with.
- **Sentiment Features:** Emotional polarity and tone of tweets.

These features are fed into ensemble classifiers—such as Random Forests—to produce a score indicating the likelihood of an account being a bot. The score ranges from 0 (likely human) to 1 (likely bot).

Strengths and Applications

Botometer's strength lies in its comprehensive feature set, allowing it to detect bots that may not exhibit obvious anomalies in any single dimension. It has been widely used in academic research and real-world investigations into social media manipulation, disinformation, and political campaigns.

Limitations

Botometer depends heavily on Twitter's API, which can impose rate limits and restrict access to historical data. Additionally, because it uses content-based features, it may struggle with multilingual data or accounts that post primarily non-textual content (e.g., images, links). It also faces challenges in detecting coordinated bots that individually mimic human behavior but act in concert.

Relation to Network-Based Approaches

While Botometer is not purely network-based, it incorporates social graph features such as retweet and mention network properties. These features are useful in detecting coordination and influence within the Twitter graph. In hybrid frameworks, combining Botometer-style features with graph-structural analysis (like CopyCatch) can improve bot detection accuracy.

6. Bot Detection Using Graph Convolutional Networks (GCNs)

Recent advances in deep learning on graphs have introduced powerful tools for analyzing social network structures. One such approach is the use of Graph Convolutional Networks (GCNs), which extend the capabilities of convolu-

tional neural networks to non-Euclidean data like graphs. Alhosseini et al. [**?**] proposed a framework for bot detection using GCNs, which leverages both the graph topology and node attributes.

Method Overview

In this approach, users are modeled as nodes in a graph, and edges represent social relationships (e.g., follower/following links). Each node is associated with a feature vector that encodes user profile data, behavioral metrics, and content-related attributes. The GCN then learns a representation for each user by aggregating information from their neighbors in the graph.

The model is trained in a semi-supervised fashion, using a small labeled subset of accounts (bots and humans), and propagates label information through the network based on learned embeddings. This allows the model to capture structural and contextual similarities that are indicative of bot behavior.

Strengths and Contributions

The use of GCNs provides several advantages:

- **Contextual Learning:** Accounts are evaluated not just in isolation, but in the context of their network neighborhood.
- **High Accuracy:** The model outperforms traditional classifiers in bot detection tasks on benchmark datasets such as Cresci-2015 and Twibot-20.
- **Scalability:** Once trained, GCNs can be efficiently applied to large-scale graphs using mini-batch or sampling strategies.

Limitations

GCNs require access to both node features and graph structure, which may be limited due to API constraints or data privacy. Moreover, label imbalance and evolving bot behavior can affect generalizability. Interpretability of deep learning models on graphs remains an open challenge compared to rule-based or graph-theoretic methods.

Relevance to Network-Based Detection

The GCN-based method complements classical SNA approaches by incorporating both local and global structure in the classification process. It represents a significant step forward in automated and adaptive bot detection in complex social graphs.

7. Comparative Analysis of Network-Based Bot Detection Methods

A wide array of bot detection techniques has been proposed, each with distinct strengths and limitations depending on the nature of the social network, the available data, and the evolving strategies of malicious agents. In this section, we compare key approaches discussed in the literature to highlight their effectiveness, scalability, and adaptability.

7.1. Effectiveness

When evaluating detection accuracy and precision, supervised learning methods such as Botometer by Varol et al. [12] and DeBot by Chavoshi et al. [5] generally outperform unsupervised approaches due to access to labeled data and advanced classification algorithms.

Botometer integrates diverse features, including friend/follower ratios and network centrality metrics, achieving high classification accuracy on known bots. However, it struggles to generalize across platforms and evolving bot behaviors. DeBot performs well in detecting bots that exhibit correlated behaviors, such as synchronized posting or mutual mentions, making it effective for identifying coordinated campaigns.

By contrast, CopyCatch [4], an unsupervised method, excels at identifying groups of bots acting in lockstep but may miss more subtle or individual bot behaviors.

7.2. Scalability

Scalability is essential for real-world applications where millions of users interact. In this regard:

- **CopyCatch** is highly scalable due to its reliance on efficient bipartite graph mining algorithms, making it suitable for large-scale systems like Facebook.
- **GCN-based approaches** used by Alhosseini et al. [3] offer good scalability with GPU acceleration, although training remains computationally expensive.
- **BotSpot** [10] balances scalability and performance by combining handcrafted features with graph embeddings, although its use of multiple data sources can be a bottleneck.

7.3. Adaptability to Evolving Bots

Modern bots adapt their behavior to evade detection, making robustness against evasion crucial:

- Kumar et al. [8] highlight adaptability by modeling misinformation spread and network roles using dynamic graph embeddings.
- GCN models [3] show strong adaptability by learning directly from structure, reducing the need for manual feature engineering.
- Traditional models like Botometer and DeBot may require retraining and feature updates to remain effective.

7.4. Interpretability

Interpretability is important for practical deployments:

- **Botometer** provides interpretable scores based on well-known features like follower ratios and activity timing.
- **GCN-based methods** and deep learning models like BotSpot [10] function as black boxes, limiting interpretability.
- **CopyCatch** is highly interpretable due to its graph-based clustering approach.

7.5. Summary Comparison

Method	Type	Accuracy	Scalability	Adaptability	Interpretability
CopyCatch	Unsupervised	Medium	High	Low	High
Botometer	Supervised	High	Medium	Medium	Medium-High
DeBot	Supervised	High	Medium	Medium	Medium
GCN (Alhosseini)	Hybrid	High	Medium-High	High	Low
BotSpot	Hybrid	High	Medium	High	Low
Kumar et al.	Hybrid	Medium	Medium	High	Medium

Table 2. Comparison of network-based bot detection methods.

7.6. Conclusion of Comparison

GCN-based models and BotSpot provide high accuracy and adaptability to new bot strategies. CopyCatch is ideal for large-scale platforms where interpretability and scalability are essential. Botometer strikes a balance between performance and explainability, although it may require frequent updates. Future research should focus on hybrid models that retain interpretability while leveraging structural graph learning to enhance adaptability.

8. Challenges in Network-Based Bot Detection

Despite the growing effectiveness of network-based techniques for detecting bots in social media, several critical challenges remain. These challenges limit the scalability, accuracy, and real-world applicability of current methodologies.

1. **Evasion and Adaptation by Bots**
   Modern bots are increasingly sophisticated, often designed to mimic the behavior and network structures of real users. They may actively form reciprocal connections, engage in interactions to boost centrality, or embed themselves in communities to appear authentic. These tactics reduce the effectiveness of detection systems that rely solely on structural anomalies.
2. **Limited Access to Comprehensive Data**
   Most academic research relies on publicly available or sampled datasets, which may not reflect the full scope or complexity of real-world social networks. Major platforms impose restrictions on API access, user metadata, and full graph structures, hindering the ability to test detection algorithms at scale or across diverse populations.
3. **Ground Truth and Labeling Issues**
   Supervised learning approaches require high-quality labeled data, which is difficult to obtain. Manual labeling is time-consuming, error-prone, and may vary across annotators. Moreover, there is no universally accepted definition of a "bot," which further complicates the annotation process and the evaluation of detection performance.
4. **Scalability of Algorithms**
   Social networks often consist of millions of nodes and edges. Many graph-based algorithms, such as k-core decomposition or betweenness centrality, become computationally expensive at large scales. Efficient, scalable implementations are necessary for real-time or near-real-time detection in practice.
5. **False Positives and Misclassification**
   Some legitimate users—such as influencers, brand accounts, or highly active individuals—may exhibit bot-like behavior in their structural metrics (e.g., high degree, low reciprocity). This overlap increases the risk of false positives, which can undermine trust in detection systems and result in unfair account suspensions.
6. **Interpretability and Trust**
   Advanced methods such as Graph Neural Networks (GNNs) may provide high performance, but often at the cost of transparency. In sensitive applications like account moderation or disinformation tracking, interpretability is critical to ensure trust, accountability, and ethical usage.

9. Conclusion

Social bots pose a growing threat to the integrity of online platforms, spreading misinformation, amplifying manipulation campaigns, and distorting online discourse. While many detection approaches rely on content or behavioral analysis, network-based methods offer a promising alternative by focusing on the structural characteristics of user interactions.

This paper has explored how social network analysis (SNA) metrics—such as centrality, clustering coefficient, and community structure—can be used to uncover bots based solely on graph topology. We reviewed several methodologies, both supervised and unsupervised, that leverage these features to identify anomalous behavior within social graphs.

Although network-based detection techniques have demonstrated strong potential, they face challenges including evasion tactics by bots, limitations in data access, and scalability to large networks. Addressing these issues will be critical for developing robust and interpretable bot detection systems.

Future research should focus on combining structural features with temporal and behavioral signals, improving ground truth labeling processes, and designing algorithms that balance accuracy with transparency and fairness. As social platforms evolve, adaptive and ethical detection strategies will be essential for maintaining trust in digital ecosystems.

References

[1] Ahmed, F., Abulaish, M., 2013. Anomaly detection in online social networks using graph-based features, in: IEEE 9th International Conference on Collaborative Computing, IEEE. pp. 1–6.

[2] Alhosseini, A., Cresci, S., Lerman, K., 2022a. Detecting social bots using graph neural networks. Journal of Complex Networks 10.

[3] Alhosseini, A., Cresci, S., Lio, P., Tesconi, M., 2022b. A graph-based approach to social bot detection, in: Proceedings of the 2022 International Conference on Web Intelligence, ACM. pp. 56–65.

[4] Beutel, A., Xu, W., Guruswami, V., Palow, C., Faloutsos, C., 2013. Copycatch: Stopping group attacks by spotting lockstep behavior in social networks, in: Proceedings of the 22nd international conference on World Wide Web, ACM. pp. 119–130.

[5] Chavoshi, N., Hamooni, H., Mueen, A., 2016. Debot: Twitter bot detection via warped correlation, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE. pp. 817–822.

[6] Davis, C.A., Varol, O., Ferrara, E., Flammini, A., Menczer, F., 2016. Botornot: A system to evaluate social bots, in: Proceedings of the 25th International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee. pp. 273–274.

[7] Jin, F., Dougherty, E., Saraf, P., Cao, Y., Ramakrishnan, N., 2013. Epidemiological modeling of news and rumors on twitter. Proceedings of the 7th Workshop on Social Network Mining and Analysis (SNA-KDD) .

[8] Kumar, S., West, R., Leskovec, J., 2018. False information on web and social media: A survey, in: Proceedings of the 25th International World Wide Web Conference (WWW), ACM. pp. 417–441.

[9] Newman, M.E., 2010. Networks: An Introduction. Oxford University Press, Oxford, UK.

[10] Ribeiro, M.E., Gonalves, M.A., Almeida, V.A., Benevenuto, F., 2021a. Detecting social bots in online networks: Taxonomy, methods, and open challenges, in: ACM Computing Surveys (CSUR), pp. 1–36.

[11] Ribeiro, M.H., Henrique, L., Benevenuto, F., Gummadi, K.P., 2021b. Botspot: A system for bot detection in the wild. ACM Transactions on the Web (TWEB) 15, 1–33.

[12] Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A., 2017. Online human-bot interactions: Detection, estimation, and characterization, in: Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM), pp. 280–289.

[13] Wasserman, S., Faust, K., 1994. Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, UK.