

Bots into the Fediverse

Francisco Moreno García

fran.moreno@upm.es

Technical University of Madrid

Pablo Perdomo Quinteiro

Technical University of Madrid

Gustavo Hernandez-Peñaloza

Technical University of Madrid

Federico Alvarez Garcia

Technical University of Madrid

Alberto Belmonte Hernandez

Technical University of Madrid

Miguel Antonio Barbero- Álvarez

Technical University of Madrid

Research Article

Keywords: Bots, Fediverse, Deep Learning, Social Media

Posted Date: September 19th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7600191/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Social Network Analysis and Mining on January 14th, 2026. See the published version at <https://doi.org/10.1007/s13278-025-01567-z>.

Bots into the Fediverse

Francisco Moreno ^{1*}, Pablo Perdomo ¹,
Gustavo Hernandez-Penalzoa ², Federico Álvarez ¹,
Alberto Belmonte ¹, Miguel Antonio Barbero-Álvarez ³

^{1*}Signals, Systems and Radiocommunications (SSR), Universidad Politécnica de Madrid UPM, Av Complutense, Madrid, 28040, Madrid, Spain.

²Escuela Técnica Superior de Ingenieros de Sistemas Informáticos, Universidad Politécnica de Madrid UPM, Alan Turing St., Madrid, 28030, Madrid, Spain.

³Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software (DLSIIS), Universidad Politécnica de Madrid UPM, C. de los Ciruelos, Boadilla del Monte, 28660, Madrid, Spain.

*Corresponding author(s). E-mail(s): fran.moreno@upm.com;

Contributing authors: p.perdomo@upm.es;
gustavo.hernandez.penalzoa@upm.es; federico.alvarez@upm.es;
alberto.belmonte@upm.es; m.balvarez@upm.es;

Abstract

Social bots are a known problem in today’s society. They are influenced by a variety of factors, ranging from the presence of bots to a lack of interaction between bots and users. This paper proposes a cross-platform approach for the detection of social bots based on profile metadata and text embeddings, applied to Twitter, Mastodon, and Bluesky user accounts. The resulting model achieves 97.15% accuracy in a four-class classification task, outperforming several established baselines, including graph-based and federated approaches while being computationally efficient. The primary contribution of this work is the demonstration that user features can support effective bot classification across heterogeneous and decentralized environments, demonstrating the feasibility of cross-domain generalization at scale. We additionally present a novel dataset that combines self-identified bot and non-bot accounts from decentralized platforms.

Keywords: Bots, Fediverse, Deep Learning, Social Media

1 Introduction

The rapid growth of decentralized and federated social networks, such as Mastodon¹ and Bluesky², has introduced new paradigms in online communication, emphasizing user autonomy, interoperability and resistance to centralized control [1]. While these platforms offer a promising alternative to mainstream social media, they are not immune to challenges, one of them being the spread of misinformation. Despite the rapid growth of decentralized platforms such as Mastodon and Bluesky, little is known about the prevalence and behavior of bots in these ecosystems.

The proliferation of misinformation on social networks has become a significant global problem, with automated accounts, or social bots, playing a pivotal role in amplifying low-credibility content [2]. In fact, social bots have been shown to amplify misinformation at scale, influencing elections, public health behaviors, and democratic processes. These bots are particularly active during the early stages of information dissemination, targeting influential users to maximize reach and engagement. Studies have shown that bots are disproportionately responsible for spreading false information, often mimicking human behavior to avoid detection and manipulate public discourse [3] [4].

Part of the persuasive power of fake news lies in their emotional content and cognitive simplicity, making it easier to comprehend and more likely to be remembered [5]. The consequences of such phenomena are seen to permeate multiple facets of society. In [6], it is shown that public health is adversely affected by the spread of misinformation, influencing public opinion on critical issues such as vaccination, smoking, and nutrition. As shown in [7], exposure to fake news can bias individuals toward more populist political parties, implying risks toward democracy. Furthermore, trust in society is declining, as more individuals begin to believe in conspiracy theories [8], which have led to several high-profile incidents such as the Pizzagate shooting (2016) [9], Capital Insurrection (2021) [10] or 5G Tower Attacks (2020) [11].

Federated social networks such as Mastodon and Bluesky, while offering decentralized alternatives to traditional platforms, face unique challenges in combating misinformation. While bots are widespread on mainstream platforms like Twitter/X and Instagram, they are harder to spot on decentralized platforms in the Fediverse. This is because harmful instances are usually banned quickly, as noted by Bono et al. [12]. In addition, the Fediverse tends to be less polarized, with 97% of instances classified as neutral according to La Cava et al. [13]. Smaller, close-knit communities help in faster moderation and response to harmful behavior. However, with the increase in the number of users on federated social media, the number of bots also increases, with almost 20% of posts being shared by bots [14].

To address the issues, various techniques have been presented to detect bots on social media platforms. These methods range from traditional machine learning algorithms to advanced deep learning and graph-based approaches. For instance, Alrubaian et al. (2021) [15] proposed a deep neural network that analyze both profile and reputation features to detect bots at the user level, achieving high classification accuracy. Graph-based methods have also gained prominence; Feng et al. (2022) [16] introduced

¹<https://joinmastodon.org>

²<https://bsky.app/>

a framework which graph interactions to enhance bot detection performance. Similarly, Tzoumanekas et al. (2024) [17] employed a Neural Architecture Search approach together with a Relational Graph Convolutional Networks, achieving superior accuracy in identifying bots. These techniques do not show how important it is to combine different types of data and use the structure of social networks to better detect and limit the influence of social bots. Nevertheless, a strong limitation remains: the majority of current models are trained and validated almost exclusively on Twitter data, raising concerns about their ability to generalize and effectiveness across other social media platforms with different dynamics, user behaviors, and network topologies. Furthermore, current deep learning approaches for bot detection often ignore the heterogeneous nature of automated accounts. In particular, they typically do not differentiate between diverse categories of bots. This distinction is crucial, since not all bots are inherently malicious as many are designed to disseminate reliable information, support content aggregation, or provide automated services of genuine value. The inability to separate harmful from benign bot activity not only reduces the precision of detection systems, but also risks undermining the utility of legitimate automated agents within online ecosystems.

Main Contributions

In this work, we propose a cross-platform approach to bot detection spanning centralized and decentralized social networks, introducing a novel dataset from Mastodon and validating models in a multiclass setting. The main contributions of this work are listed below:

- **Cross-platform bot detection.** We propose a bot detection framework explicitly adapted to both centralized (Twitter) and federated (Mastodon) social networks, demonstrating the feasibility of cross-domain generalization in heterogeneous environments.
- **Unified feature schema.** We design a consistent mapping of metadata and textual fields across Twitter, Mastodon, and Bluesky, allowing the reuse of models without extensive feature engineering or platform-specific preprocessing.
- **Heterogeneous bot classification.** Unlike previous work limited to binary detection, our approach supports a multiclass taxonomy of bots (self-declared, social, spam) and non-bots, reducing false positives on benign automation.
- **Empirical evaluation at scale.** We assemble and release a novel data set of more than 637,000 Mastodon accounts and integrate it with established benchmarks (Cresci’s corpus[18]). Experiments cover domain transfer, retraining, and multiclass classification under realistic conditions.
- **Competitive performance.** Our best model (XGBoost with metadata and multilingual embeddings) achieves 97.15% accuracy in four-class classification, outperforming or matching recent graph-based and federated approaches while being computationally efficient.
- **Interpretability and insights.** We provide analysis and visualizations of the importance of characteristics (t-SNE projections) that reveal the dominant role

of semantic embeddings, offering actionable insights for platform moderation and future research.

The remainder of the paper is structured as follows: In Section 2, we review the state-of-the-art in social bot detection, highlighting approaches most closely related to our study. Section 3 describes the proposed methodology, including data collection, feature extraction, and the architectures of the AI models employed. Section 4 reports the experimental setup and presents the most representative results. Section 5 provides a critical discussion of these findings, addressing their implications, limitations, and challenges encountered. Finally, Section 6 summarizes the main contributions of the work and outlines directions for future research.

2 Related Work

2.1 Classical and Text-based Approaches

Early work on social bot detection has predominantly focused on centralized platforms such as Twitter. A large body of research has relied on metadata features including account age, posting frequency, follower/following ratios, or profile attributes combined with textual analysis of user-generated content [18–20]. Although these approaches established important baselines, they often struggled with generalization, as bots increasingly mimic human-like behavior in both activity patterns and linguistic style.

This work takes a different direction from previous research on bot detection in centralized social networks [21]. In particular, a multilingual approach for bot identification in Twitter was proposed using a combination of metadata and text-based features encoded through transformer models. Metadata included activity (e.g., account age, tweet frequency), popularity (e.g., follower count), and profile information (e.g., verified status, language). Textual fields such as user descriptions and names were embedded using multilingual models such as BERT-base-multilingual-cased [22] and RoBERTa-large-mnli [23]. These embeddings were concatenated with the metadata to form input vectors for a dense neural network architecture, Bot-DenseNet, which produced low-dimensional user representations while performing binary classification between bots and humans. The system provided robust multilingual performance and compact representations for downstream tasks.

Despite these advances, classical and text-based methods share two main limitations. First, they are heavily platform-dependent: metadata fields such as “verified status” or “retweet count” are specific to Twitter and do not directly transfer to decentralized networks like Mastodon or Bluesky, where account structures and available metadata differ substantially. Second, most models are constrained to binary classification (bot vs. human), overlooking the diversity of automated agents, some of which have non-malicious purposes such as news dissemination or content aggregation [24, 25]. As a consequence, these methods risk high false-positive rates when applied to heterogeneous ecosystems where not all automated behavior is malicious.

In contrast, our work extends beyond these classical approaches by (i) proposing a unified feature schema that harmonizes metadata across multiple platforms, including

federated networks; (ii) supporting multiclass classification to distinguish between different categories of bots; and (iii) demonstrating cross-platform generalization rather than restricting evaluation to Twitter alone.

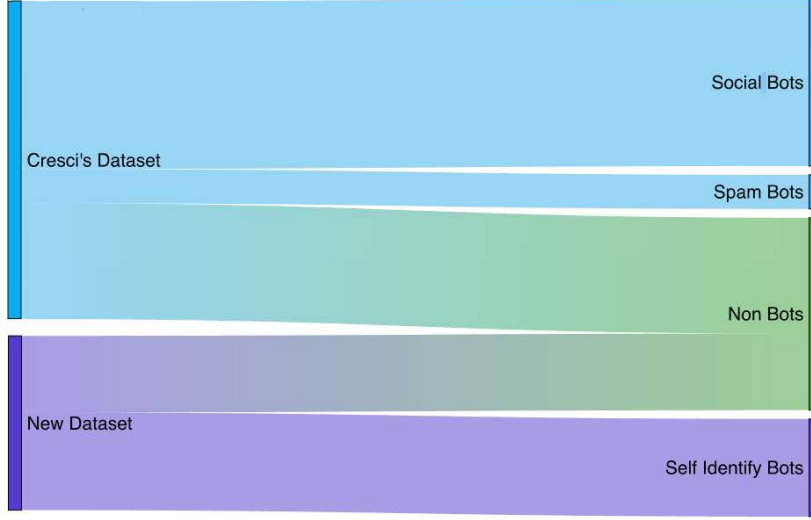


Fig. 1 Distribution of accounts across datasets and class labels. The flows illustrate how accounts from Cresci’s dataset and the new dataset are divided into social bots, spam bots, self-identified bots, and non-bot accounts.

2.2 Graph-based Methods and Federated approaches

Graph-based methods have gained prominence in social bot detection by modeling relational structures between users. Feng et al. introduced BotRGCN [26], a relational graph convolutional framework that integrates user profile metadata, tweet content, and categorical properties, propagating them through a heterogeneous graph constructed from follower–following links. It combines user profile metadata, content of the tweets and categorical properties, and then propagates them across a heterogeneous graph built from following and follower links. By explicitly capturing community structures and disguised identities, BotRGCN demonstrated strong performance on Twitter benchmarks. Similarly, other graph-based models such as BotDGT [27] exploit temporal and structural signals to uncover coordinated behaviors. Despite their effectiveness, these methods assume access to dense and centralized graph data, an unrealistic requirement in decentralized ecosystems such as the Fediverse, where graph information is fragmented across multiple instances and often only partially available.

To address data privacy and distribution constraints, recent work has explored federated learning. Wang et al. proposed FedKG [28], a knowledge distillation–based federated graph method that combines relational graph convolutional networks

(RGCN) with federated learning to enable multi-party bot detection without sharing raw data. The model mitigates class imbalance through class-level cross-entropy loss and employs a global generator to create pseudo-samples guiding local optimization. Client models also share logits to improve the global classifier while addressing data heterogeneity. However, its reliance on a central server for knowledge distillation and logits aggregation limits its adaptability to highly heterogeneous networks such as Mastodon.

Yang et al. introduced FedACK [29], a federated adversarial contrastive knowledge distillation framework for cross-lingual and cross-model bot detection. The approach employs a GAN-based federated architecture where a global generator captures data distribution and distills it into local models, while local generators enhance data with hard-to-decide samples. By combining adversarial and contrastive learning with a global generator, FedACK improves model consistency under non-IID data distributions across clients and languages. While this represents a significant step toward robustness in federated contexts, evaluations remain confined to Twitter datasets, and the approach still presupposes globally coordinated training, an assumption that does not hold in federated social networks without centralized authority.

In summary, graph-based methods excel at leveraging relational signals but are constrained by the need for dense graph structures, while federated approaches address privacy and distribution but depend on centralized coordination and remain Twitter-centric. In contrast, our work does not rely on complete social graphs or central orchestrators. Instead, we propose a unified feature schema that harmonizes meta-data and textual information across both centralized and decentralized platforms, and we evaluate cross-platform generalization in a multiclass setting that explicitly distinguishes between different categories of bots.

2.3 Dynamic and Heterophily-aware Models

A further line of research has focused on the dynamic nature of social networks and the challenges posed by heterophily. Dynamic graph neural networks (DGNNs) aim to capture temporal patterns of user activity and evolving relational structures. Liu et al. proposed BotDGT [27]. The architecture combines a structural module (graph attention across time snapshots) and a temporal module (self-attention on user evolution), capturing both topological and behavioral dynamics. Its emphasis on modeling evolving user behavior patterns, such as changes in clustering coefficients or reciprocal interactions, addresses limitations in traditional GNN methods that rely on static snapshots. By explicitly incorporating temporal evolution, BotDGT achieves improved robustness against sophisticated, evolving bot campaigns. However, these approaches assume continuous access to fine-grained temporal data streams, which are rarely available in federated environments where data exposure is limited and often instance-specific.

Another major challenge is heterophily, i.e., the tendency of bots to connect preferentially with human accounts rather than with each other. Classical homophily-based graph models degrade significantly under such conditions. Works on generalized GNNs [30, 31] seek to capture long-range dependencies and structural signals beyond

local homophily. While these models improve detection under adversarial connectivity, they remain reliant on dense and accurate graph data that may not be accessible in decentralized ecosystems.

He et al. presented a heterophily-aware representation learning method enhanced by prototype-guided cluster discovery [32]. Unlike traditional GCN-based models that assume homophily (similar nodes link together), this model recognizes and exploits heterophily patterns common in bot-human interactions. Cluster-based constraints further improve the separation of latent features, increasing resilience against noisy or weakly connected graphs.

Dynamic and heterophily-aware models extend the frontier of bot detection by accounting for temporal adaptations and adversarial connectivity patterns. Nevertheless, they inherit the same structural dependency as other graph-based approaches: the need for complete, centralized, and high-resolution network data. Our work diverges from this line of research by not relying on full social graphs. Instead, we design a cross-platform schema that leverages metadata and textual features consistently across centralized (Twitter) and decentralized (Mastodon, Bluesky) platforms, enabling multiclass bot classification without assuming access to dense graph connectivity.

Type	Twitter (Cresci)	Mastodon	Bluesky
<i>Profile Fields</i>			
- Username	- name	- username	- handle
- Display Name	- screen_name	- display_name	- displayName
- Post Count	- statuses_count	- statuses_count	- postsCount
- Follower Count	- followers_count	- followers_count	- followersCount
- Profile Image	- profile_image_url	- avatar	- avatar
- Banner Image	- profile_banner_url	- header	- banner
- Description	- description	- note	- description
- Following Count	- following	- following_count	- followsCount
- Account Creation Date	- created_at	- created_at	- createdAt
<i>Post Fields</i>			
- Text Content	- text	- content	- record.text
- Repost Count	- retweet_count	- reblogs_count	- repostCount
- Reply Count	- reply_count	- replies_count	- replyCount
- Like Count	- favorite_count	- favourites_count	- likeCount
- Sensitive Content	- possibly_sensitive	- sensitive	- estimated via record.text
- Hashtag Count	- num_hashtags	- len(tags)	- len(labels)
- URL Count	- num_urls	- extracted from content	- extracted from record.text
- Mention Count	- num_mentions	- len(mentions)	- len(record.facets)
- Post Date	- created_at	- created_at	- record.createdAt

Table 1 Equivalent metadata fields across Twitter (Cresci), Mastodon, and Bluesky for profile and post objects.

In summary, previous work has achieved significant advances in bot detection, but remains constrained in three ways: (i) the vast majority of studies are trained

and validated exclusively on Twitter data, (ii) methods are often limited to binary classification without accounting for benign forms of automation, and (iii) graph-based or federated models assume centralized availability of dense relational data, which is incompatible with the fragmented nature of federated platforms. To the best of our knowledge, no previous work has demonstrated cross-platform bot detection spanning centralized and decentralized social networks. This gap motivates the present study, which introduces a unified feature schema and systematically evaluate models across Twitter, Mastodon, and Bluesky.

Bot-DenseNet (Neural Network)	
<i>Architecture</i>	2 fully connected hidden layers
<i>Hidden Units</i>	256
<i>Activation Function</i>	SELU (hidden), Sigmoid (output)
<i>Loss Function</i>	Binary Cross Entropy
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	0.001
<i>Batch Size</i>	256
<i>Input Features</i>	Metadata + Description Embedding
<i>Framework</i>	PyTorch
Random Forest	
<i>Max Depth</i>	2
<i>Input Features</i>	Metadata + Description Embedding
<i>Library</i>	scikit-learn
Logistic Regression	
<i>Solver</i>	LBFGS
<i>Max Iterations</i>	500
<i>Input Features</i>	Metadata + Description Embedding
<i>Library</i>	scikit-learn
XGBoost (multiclass)	
<i>Objective</i>	multi:softmax
<i>Number of Classes</i>	4
<i>Max Depth</i>	6
<i>Learning Rate</i>	0.1
<i>Subsample</i>	0.8
<i>Colsample by Tree</i>	0.8
<i>Random State</i>	42
<i>Input Features</i>	Metadata + Description Embedding + Averaged Post Embeddings
<i>Library</i>	xgboost

Table 2 Model architectures and training parameters used across experiments

3 Methods

This study addresses the automatic identification of bot accounts in social media platforms using metadata and text-based features. The objective is to classify user accounts

as bots or non-bots through supervised learning models. The proposed approach extends our previous research where a deep learning model was developed for Twitter, integrating account metadata with transformer-based textual embeddings [21]. In this work, we adapt the model to federated platforms, specifically Mastodon and Bluesky, taking advantage of their compatible metadata structures. Additionally, using the labeled data provided by Cresci [18] and the new dataset created, we further explore a multiclass classification setting to distinguish between different types of bots.

The methodology is applied to data collected from Mastodon, a federated platform with similar account structures. Initially, the Twitter-trained model [21] is evaluated directly on Mastodon data to assess its baseline performance under domain shift. Subsequently, the model is retrained using both the original Twitter data and the newly collected Mastodon data. Comparative experiments are conducted using neural networks, random forests, and logistic regression. Finally, for the multiclass classification task, we employ an XGBoost model [33] incorporating post-level embedding features (tweets and toots).

Category	Feature	Description
<i>Username</i>	- Length	- Length of the username
	- Number of Digits	- Number of digits of the username
	- Number of Letters	- Number of letters in the username
	- Number of Special characters	- Number of special characters in the username
	- Starts with Digit	- True if the first character of the username is a digit
	- Ends with Digit	- True if the last character of the username is a digit
	- Fuzzy Score	- Fuzzy Score Username vs Display Name
<i>Account</i>	- Followers Count	- Number of followers of the user
	- Following Count	- Number of followings of the user
	- Days	- Number of days since the account creation
	- Description	- Embedding of the description of the user
	- Language	- Language of the user
<i>Activity</i>	- Statuses Count	- Total number of posts of the user
	- Average Reply Count	- Average number of replies per post
	- Average Repost Count	- Average number of reposts per post
	- Average Favorite Count	- Average number of favorites per post
	- Average Number of Hashtags	- Average number of hashtags per post
	- Average Number of URLs	- Average number of URLs per post
	- Average Number of Mentions	- Average number of mentions per post
	- Sensitive Post Percentage	- Percentage of posts tagged as sensitive

Table 3 Overview of extracted features grouped by type.

3.1 Data Collection

A dataset of Mastodon accounts was constructed by collecting public posts using the platform’s Application Programming Interface API during January and February

2025³. Data was collected from users posting under the top ten trending topics across multiple instances, with topics obtained via the MAstodon API and updated over time. This process gathered 637,251 unique accounts, of which 23,269 were labeled as bots via Mastodon’s self-identification field.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.89	0.89	0.89	0.89
Random Forest	0.86	0.86	0.86	0.86
Bot-DenseNet (NN)	0.70	0.76	0.70	0.68

Table 4 Performance of models retrained on 44k Mastodon accounts (binary classification). All metrics are weighted averages.

During the early stages of data collection, a preliminary subset consisting of 200,000 accounts was used to evaluate the original deep learning model trained on Twitter. From this set, 14,000 self-declared bots were identified, and a balanced test set of 28,000 accounts was sampled for initial transfer evaluation.

Upon completion of the full retrieval process, the dataset was partitioned for the main binary classification experiments. A balanced subset of 44,000 accounts (22,000 bots and 22,000 non-bots) was created for training and testing purposes. All accounts were selected to have complete metadata and consistent structural features.

For the final experiment involving multiclass classification, the dataset was extended with additional labeled data from Cresci et al. [18], which includes verified instances of genuine accounts, social bots and spam bots from Twitter. The combined dataset contains 2,862 self-identified bots from Mastodon, along with 4,912 social bots and 1,000 spam bots from the Cresci corpus and 5,637 non-bots from both (Figure 1. To enrich the feature space, a total of 102,100 user posts (tweets and toots) were recovered.

To mitigate sampling bias introduced by trending topics, we rotated instances and timestamps when querying the API, and we verified that language and domain distributions remained stable across pulls. We also removed all accounts used in the early 200k subset from subsequent training to avoid leakage.

3.2 Feature Extraction

The model input consists of structured metadata features and semantic textual embeddings, extracted uniformly across platforms. Metadata features are derived from user profiles, while text embeddings are computed from descriptions and posts using a multilingual transformer model. A detailed mapping of metadata field names across Twitter, Mastodon, and Bluesky is provided in Table 1.

³<https://docs.joinmastodon.org/api/>

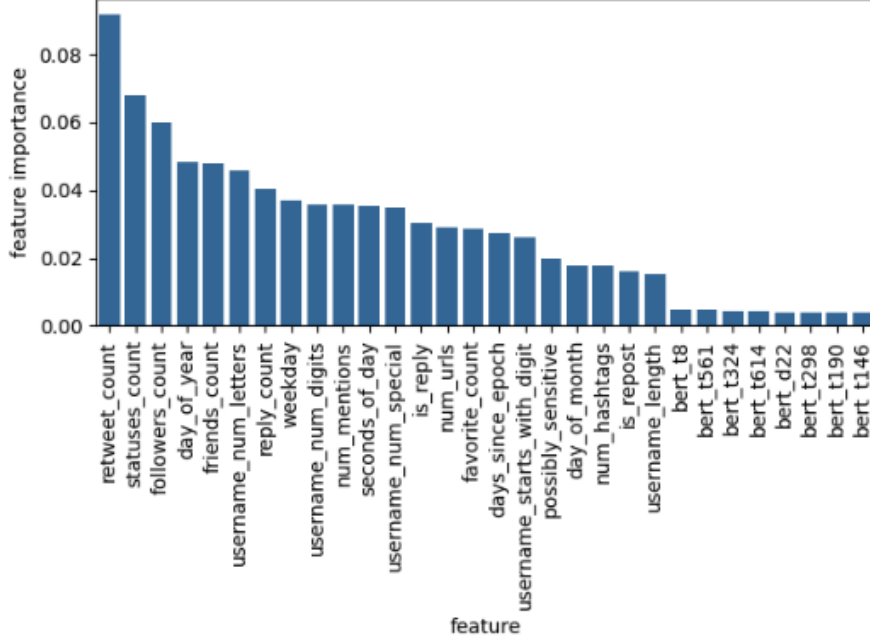


Fig. 2 Feature importance scores from the Random Forest classifier. Metadata features dominate over BERT-based embeddings.

3.2.1 Metadata Features

A common set of account-level features was constructed across all platforms. These include activity statistics (e.g., number of followers, following count, status count), temporal indicators (e.g., account age), and interaction-based averages (e.g., mean retweets, replies, likes). Fields not directly available across all platforms were discarded or approximated to maintain schema consistency. For example, the profile description on Twitter and Bluesky, correspond to note on Mastodon.

Additional engineered features were computed from usernames. The following attributes were extracted: total length, number of digits, number of letters, number of special characters, whether the username starts or ends with a digit. These were obtained via a custom function that parses the username string character by character. To capture the similarity between display names and usernames, a score feature was computed based on the normalized Levenshtein distance between the two strings. We standardize all continuous features using statistics computed on the training split only and impute missing values with the median of each feature.

For the final multiclass classification experiment, additional metadata features were introduced based on aggregated statistics from user posts. For each account, the 20 most recent posts were analyzed to compute average values of amount of replies, likes, retweets, number of hashtags, number of URLs, number of mentions, and the proportion of posts flagged as possibly sensitive. A list of all the features retrieved is shown in Table 3.

3.2.2 Text Embeddings

Semantic textual representations were obtained using the multilingual BERT model (bert-base-multilingual-cased) as introduced by Devlin et al. [22]. This model allows tokenization and embedding across multiple languages and was selected to support the multilingual content in Mastodon and Bluesky. Profile descriptions (or notes) were tokenized and embedded using the model’s output layer. For text fields we use a maximum sequence length of 128 tokens and additionally we apply basic normalization (lowercase where applicable, URL/user mention masking) and mean pooling over the last hidden layer. Post-level embeddings are averaged per user without temporal weighting, unless otherwise noted.

Finally, for multiclass classification, additional text features were included by aggregating user-level embeddings from recent activity. Specifically, the 20 most recent posts (tweets or toots) for each user were embedded individually and averaged to form a single fixed-length vector per *user*.

3.3 Model Architectures

Three classification approaches were employed across the different stages of experimentation: a deep neural network originally proposed for bot detection on Twitter, classical machine learning baselines, and a gradient boosting model for multiclass classification. Graph-based methods require dense centralized data; therefore, we adopt methods based on metadata + text, which are transferable to federated networks (see section 2).

Model	Mean Accuracy	Std. Dev.	Training Time
XGBoost [33]	0.9711	0.0039	56 seconds
MLP [34]	0.9437	0.0021	5.37 minutes
TabNet [35]	0.9354	0.0085	36 minutes

Table 5 Comparison of model performance and training time

3.3.1 Deep Neural Network (Bot-DenseNet)

The neural network architecture corresponds to the model introduced in previous work on Twitter bot detection [21]. It processes a concatenated feature vector consisting of metadata and a transformer-based embedding of the user’s profile description. The same architecture, loss function, optimizer, and activation functions were used without modification. Only the metadata fields were mapped accordingly, and the same multilingual BERT model was reused to ensure the input compatibility. Hyperparameters were selected via stratified 3-fold cross-validation on the training split. For XGBoost we used early stopping with a 10% validation hold-out. The hyperparameters and training configuration are summarized in Table 2.

3.3.2 Classical Machine Learning Models

Random Forest and Logistic Regression classifiers were used as baselines. These models were trained using metadata features combined with profile description embeddings, excluding post-level content. Implementations were based on standard scikit-learn libraries. All relevant configuration parameters are listed in Table 2.

3.3.3 XGBoost for multiclass Classification

For the final experiment, a four-class classification task was defined using self-identified bots, non-bots, social bots, and spam bots. The input of the model consisted of meta-data features, description embeddings, and average embeddings of the most recent posts. Classification was performed using an XGBoost model configured for multiclass output. The training parameters are reported in Table 2.

3.4 Experimental Protocol

Three stages of experiments were conducted to evaluate model performance across classification settings. The evaluation focused on both binary and multiclass classification using fixed dataset partitions and standard performance metrics.

For the binary classification experiments, two datasets were used: a 28,000 account subset balanced between self-identified bots and non-bots, and a 44,000 account dataset balanced in the same way. The same training and testing splits were applied consistently across all models in each experiment phase to make the results comparable. Stratified sampling was used to preserve class balance in both cases. No cross-validation was performed in these stages.

Fold	Accuracy	F1
1	0.9705	0.9782
2	0.9722	0.9733
3	0.9719	0.9725
4	0.9695	0.9712
5	0.9726	0.9742
Mean	0.9715	0.9739
Std. Dev.	0.0030	0.0024

Table 6 Cross-validation accuracy and F1 scores for multiclass classification using XGBoost.

In the final multiclass experiment, the dataset included four classes: non-bots (5,637), social bots (4,912), self-identified bots (2,862), and spam bots (1,000). Due to the inherent class imbalance, no undersampling or oversampling was applied. Instead, model robustness was evaluated using five-fold cross-validation. The dataset was randomly partitioned into five folds using KFold cross-validation with shuffling and a fixed random seed. Each fold was used once as the validation set while the remaining four

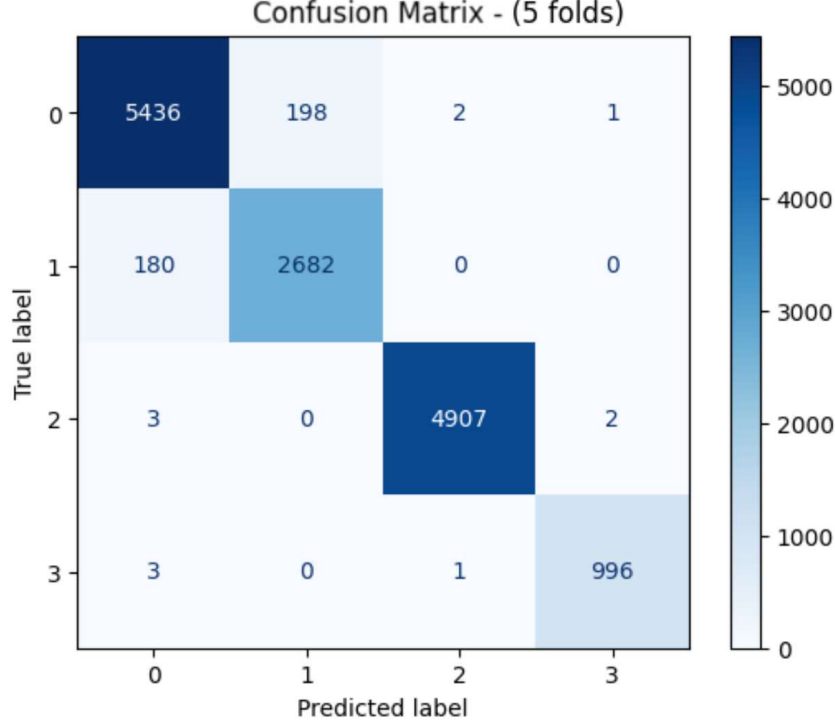


Fig. 3 Global confusion matrix obtained by aggregating predictions across all folds in the 5-fold cross-validation.

served for training, and accuracy was computed on the held-out fold. Mean accuracy and standard deviation across folds were reported to quantify model consistency.

Performance was measured using accuracy, precision, recall, and F1-score. In binary classification tasks, these metrics were reported per class and overall accuracy. In the multiclass setting, only accuracy was used to summarize fold-level performance, while detailed per-class metrics were computed separately during analysis.

4 Results

4.1 Binary Classification on Mastodon

The first experiment assessed the performance of the original deep learning model, trained on Twitter data, when applied directly to Mastodon accounts without retraining. The evaluation was performed using a balanced dataset of 28,000 Mastodon accounts, composed of 14,000 self-identified bots and 14,000 non-bots. The dataset was randomly partitioned into training (80%) and testing (20%) subsets using a stratified split.

The Bot-DenseNet model, introduced in prior work [21], achieved an F1-score of 0.71 on this task. This represents a notable drop in performance compared to the F1-score of approximately 0.77 reported in the original Twitter-based evaluation. No

additional fine-tuning or parameter adaptation was performed at this stage; the model was evaluated as-is on the new domain. The results suggest that the model’s performance does not generalize directly to Mastodon data, likely due to structural and behavioral differences across platforms. Differences in follower dynamics, activity levels, or linguistic norms may contribute to this degradation, highlighting the challenges of cross-platform model transfer.

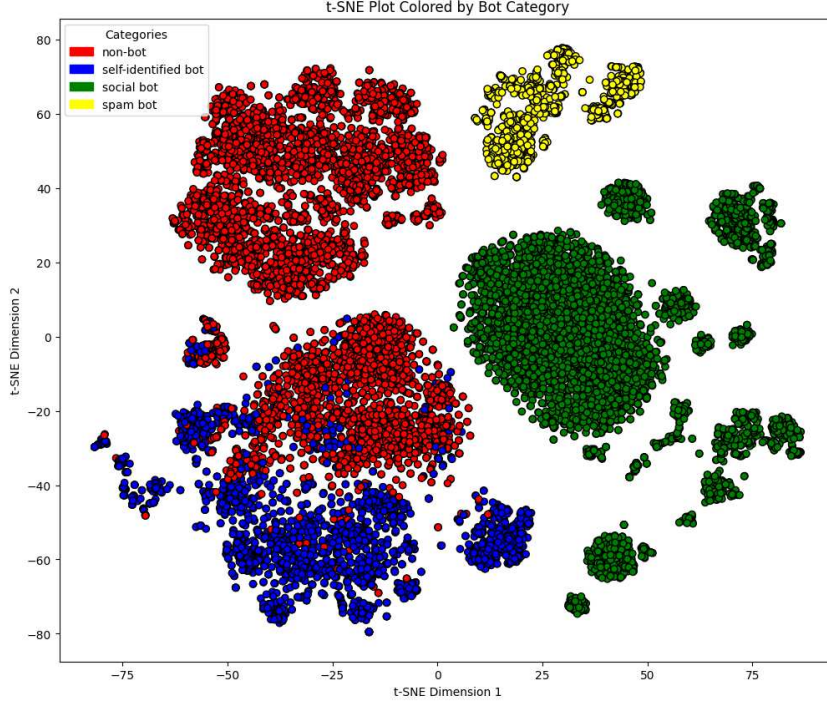


Fig. 4 t-SNE projection of user feature vectors in the multiclass experiment. Each point represents a user colored by class label: self-identified (blue), not bot (red), social bot (green), and spam bot (yellow). While Cresci-derived classes are well separated, overlap is visible between Mastodon non-bots and self-identified bots.

4.2 Retraining on Mastodon Data

In the second experiment, all models were retrained from scratch using a balanced dataset of 44,000 Mastodon accounts, with equal numbers of self-identified bots and non-bots [34]. Three classifiers were evaluated: the Bot-DenseNet neural model, a Random Forest, and a Logistic Regression model. All models received the same input feature set, combining metadata and profile description embeddings.

Performance results are summarized in Table 4. The logistic regression classifier achieved the best overall results with an accuracy of 0.89 and an F1-score of 0.89. Random Forest followed closely with an F1-score of 0.86. Surprisingly, the retrained

deep neural model performed significantly worse, reaching only 0.68 in F1 and 0.70 in accuracy. These results suggest that, in this context, simpler models not only matched but improve the performance of the more complex neural network architecture.

A feature importance analysis conducted using the Random Forest model (Figure 2) revealed that metadata features related to user activity, particularly average retweet count, status volume, and follower/following counts, strongly influenced predictions. Username-based structural features also contributed to the decision process of the model. While individual BERT embedding dimensions appeared less informative in isolation, the sum of all BERT dimensions, treated as a single aggregate feature, was identified as the most important individual predictor in the dataset.

These findings suggest that in the absence of high-quality fine-tuning or large training volumes, classical models may offer more robust and interpretable performance on real-world federated social data.

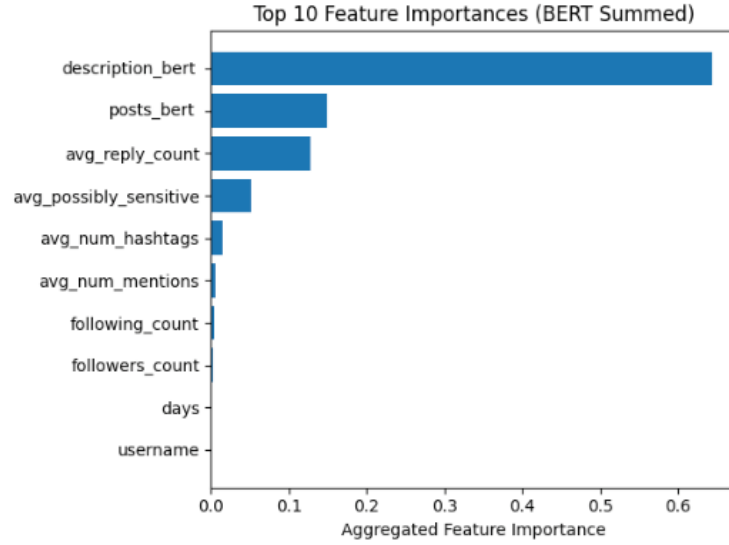


Fig. 5 Top 10 aggregated feature importances in the multiclass classification experiment. The summed BERT embedding from the profile description (`description_bert`) dominates the signal, followed by post-level embeddings (`posts_bert`) and behavioral metadata features such as average reply count and sensitivity.

4.3 multiclass Classification

In the final experiment, a four-class classification task was conducted using a combined dataset consisting of 14,411 user accounts. The classes were defined as: non-bots (3394 Twitter + 2243 Mastodon), self-identified bots (2,862), social bots (4,912), and spam bots (1,000). Non-bots included accounts from both Mastodon and Cresci’s dataset [18]. Each user was represented by a feature vector combining metadata, a profile

description embedding, and an average embedding of up to 20 of their most recent posts.

We evaluated three candidate models for this task: XGBoost [33], a Multi-Layer Perceptron (MLP)[34], and TabNet [35]. After comparing their accuracy and training times, we selected XGBoost as the final model due to its superior performance and significantly faster training. The experiment used a five-fold cross-validation with stratified sampling to account for class imbalance. Accuracy was computed for each fold. A detailed comparison of the three models is presented in Table 5.

XGBoost per-fold and global results are listed in Table 6 and in Figure 3. No resampling or class weighting was applied, indicating that the classifier performed robustly even under moderate imbalance.

A t-SNE [36] projection of the embedded feature space (Figure 4) reveals clear separation between the Cresci-derived classes (social and spam bots), while the distinction between Mastodon non-bots and self-identified bots was more diffuse. This suggests that semantic similarity and behavioral patterns overlap more strongly between organic and bot-like accounts within the same platform, while cross-source bot types remain more distinguishable.

Feature importance analysis (Figure 5) based on aggregated contributions confirms that text-based features were dominant in this setting. The summed BERT embedding of the profile description alone accounted for 64.4% of the total importance, followed by the average embedding of recent posts and average reply count. Metadata features such as follower counts and username structure contributed minimally in comparison.

These results indicate that fine-grained bot classification benefits strongly from semantically rich textual features and that platform-specific distinctions may challenge traditional metadata-based separation strategies.

Method	Platform(s)	Features	Type	Performance
Bot-DenseNet[21]	Twitter	Metadata + Desc. Embeddings	Binary	Acc.=Not Provided F1 = 77%
FedACK [29]	Twitter (TwiBot-20)	Graph + Contrastive + Meta	Federated Binary	Acc. = 95.1% F1=Not Provided
FedKG [28]	Twitter (TwiBot-20)	Graph + Distillation + Meta	Federated Binary	Acc. = 93.4% F1=Not Provided
Cresci et al. [18]	Twitter	Graph + Behavior (manual labels)	Binary	Acc. = 97.6% F1=Not Provided
BoostBot [32]	Twitter15, TwiBot-20	GNN + Heterophily + Clustering	Semi-sup. Binary	Acc. $\approx$ 89–91% F1. $\approx$ 83–89%
BotRGCN [26]	Twitter (TwiBot-20)	Metadata + Tweet + Category + Graph	Binary	Acc. = 84.6% F1= 87%
BotDGT [37]	Twitter (TwiBot-20/22)	Dynamic Graph + Temporal Attention	Binary	F1 = 88.87% F1 =88.7%
Ours (Final)	Mastodon, Bluesky, Cresci	Metadata + Desc. + Posts	Multiclass	Acc. = 97.15% F1 =97.3%

Table 7 Comparison of bot detection models: features, platforms, learning settings, and performance.

5 Discussion

This work builds on previous bot classification research by adapting and extending deep learning methods to federated social media platforms [21]. The original Bot-DenseNet model achieved an F1 score of 0.80 on Twitter using a combination of metadata and transformer-based embeddings. Our final model, using XGBoost on a multiclass dataset that includes both description and posts embeddings, achieved 97.15% accuracy.

The 2017 approach of Cresci et al. [18], which reached 97.6% precision on a manually labeled Twitter dataset, is comparable in terms of overall performance but relies heavily on graph features and curated labels, while our method adapts directly to user-declared identity in a decentralized environment.

Federated models such as FedACK [29] and FedKG [28] reached 95.1% and 93.4% accuracy on TwiBot-20. These methods use more complex designs, such as adversarial learning or knowledge sharing between models. In contrast, our model uses basic text and account features, yet performs better and is easier to apply to different platforms.

Graph-based models such as BotDGT [37] and BotRGCN [26] use network structure and account connections to detect bots. BotDGT looks at how users change over time using attention layers, and BotRGCN builds user graphs from Twitter data. These models reached 88.87% and 84.6% accuracy. Our method does not use social graphs, which makes it easier to apply to platforms like Mastodon and Bluesky, where user connections are harder to collect.

Boosting Bot detection [32] improves detection using cluster patterns in the data, reaching up to 91% accuracy. Although this approach offers robustness in specific contexts, our model achieves higher accuracy without requiring test-time learning, cluster labels, or specialized graph heuristics.

In addition, Table 7 reports the performance of representative approaches from the related literature discussed in this section, providing a direct comparison between our method and prior state-of-the-art models.

6 Conclusion

This work has proposed a cross-platform method for the detection of social bots based on profile metadata and text embeddings, applied to Twitter, Mastodon, and Bluesky user accounts. A novel dataset was assembled, combining self-identified bot and non-bot accounts from decentralized platforms with manually labeled data from prior resources. The resulting model achieved 97.15% accuracy in a multiclass classification task, outperforming several established baselines, including graph-based and federated learning approaches.

The primary contribution of this work is the demonstration that user features, including aggregated profile and post embeddings, can support effective bot classification across heterogeneous and decentralized environments. To the authors' knowledge, this represents the first approach incorporating real user data from the Fediverse with aligned cross-platform metadata.

Several limitations are noted. The model was not explicitly evaluated on Bluesky accounts, despite the compatibility of metadata. Explainability is limited to basic feature importance scores, and platform-specific variation in language use or moderation policies has not been systematically assessed.

Future work may include the integration of large language models to improve semantic understanding, expansion to a broader range of Fediverse instances, and the application of post hoc interpretability techniques to enhance transparency and trust in classification outcomes.

Acknowledgements. The authors acknowledge the use of the Cresci et al. (2017) dataset with permission. This dataset provided manually labeled Twitter accounts for bot classification and served as an external benchmark for comparative evaluation.

Declarations

Funding. This work was supported in part by AI-CODE, an initiative funded by the European Union Horizon Europe Research and Innovation program under grant agreement N° 101135437

Conflict of interest. The authors declare no conflict of interest.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The data set created for this study is available upon reasonable request. Interested researchers may contact the corresponding author at fran.moreno@upm.es. The data set does not include components derived from Cresci et al. (2017).

Materials availability. Not applicable. In this study, no new materials were generated.

Code availability. All scripts used for data processing, model training, and evaluation are available upon request. Public release is under consideration and may be provided in a future repository.

Author contributions. F.M.G Conceptualization, Methodology, Software, Formal analysis, Validation, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. P.P.Q Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Writing - Review & Editing. G.H.P. Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision. F.A.G. Funding acquisition, Review & Editing, Visualization, Supervision. A.B.H. Review & Editing, Visualization, Supervision. M.A.B. Project administration, Review & Editing, Visualization, Supervision.

References

- [1] Cava LL, Greco S, Tagarelli A. Understanding the growth of the Fediverse through the lens of Mastodon. *Applied Network Science*. 2021 12;6:1–35. <https://doi.org/10.1007/S41109-021-00392-5/FIGURES/10>.
- [2] Aimeur S E ; Amri, Brassard G. Fake news, disinformation and misinformation in social media: a review. *Soc Netw Anal*. 2023;13(11). <https://doi.org/https://doi.org/10.1007/>.
- [3] Shao C, Ciampaglia GL, Varol O, Yang KC, Flammini A, Menczer F. The spread of low-credibility content by social bots. *Nature Communications* 2018 9:1. 2018 11;9:1–9. <https://doi.org/10.1038/s41467-018-06930-7>.
- [4] Luo H, Meng X, Zhao Y, Cai M. Rise of social bots: The impact of social bots on public opinion dynamics in public health emergencies from an information ecology perspective. *Telematics and Informatics*. 2023;85:102051. <https://doi.org/https://doi.org/10.1016/j.tele.2023.102051>.
- [5] Carrasco-Farré C. The fingerprints of misinformation: how deceptive content differs from reliable sources in terms of cognitive effort and appeal to emotions. *Humanities and Social Sciences Communications*. 2022 12;9:1–18. <https://doi.org/10.1057/S41599-022-01174-9;SUBJMETA=4000,4003,4007,4008,4009,4014,4045;KWRD=CULTURAL+AND+MEDIA+STUDIES,INFORMATION+SYSTEMS+AND+INFORMATION+TECHNOLOGY,LANGUAGE+AND+LINGUISTICS,SCIENCE>.
- [6] Suarez-Lledo V, Alvarez-Galvez J. Prevalence of Health Misinformation on Social Media: Systematic Review. *J Med Internet Res*. 2021 Jan;23(1):e17187. <https://doi.org/10.2196/17187>.
- [7] Cantarella M, Fraccaroli N, Volpe R. Does fake news affect voting behaviour? *Research Policy*. 2023;52(1):104628. <https://doi.org/https://doi.org/10.1016/j.respol.2022.104628>.
- [8] Hanley HWA, Kumar D, Durumeric Z. A Golden Age: Conspiracy Theories' Relationship with Misinformation Outlets, News Media, and the Wider Internet. *Proc ACM Hum-Comput Interact*. 2023 Oct;7(CSCW2). <https://doi.org/10.1145/3610043>.
- [9] Bleakley P. Panic, pizza and mainstreaming the alt-right: A social media analysis of Pizzagate and the rise of the QAnon conspiracy. *Current Sociology*. 2021;<https://doi.org/doi.org/10.1177/00113921211034896>.
- [10] Antonio RJ. Democracy and capitalism in the interregnum: Trump's failed self-coup and after. *Critical Sociology*. 2022;48(6):937–965.

- [11] Beckes L SD. Social baseline theory: State of the science and new directions. *Current Opinion in Psychology*. 2021;43:65–71. <https://doi.org/10.1016/j.copsyc.2021.06.004>.
- [12] Bono CA, La Cava L, Luceri L, Pierri F. An Exploration of Decentralized Moderation on Mastodon. In: *Proceedings of the 16th ACM Web Science Conference. WEBSCI '24*. New York, NY, USA: Association for Computing Machinery; 2024. p. 53–58. Available from: <https://doi.org/10.1145/3614419.3644016>.
- [13] La Cava L, Mandaglio D, Tagarelli A. Polarization in Decentralized Online Social Networks. In: *Proceedings of the 16th ACM Web Science Conference. WEBSCI '24*. New York, NY, USA: Association for Computing Machinery; 2024. p. 48–52. Available from: <https://doi.org/10.1145/3614419.3644013>.
- [14] Zia HB, Raman A, Castro I, Tyson G.: Collaborative Content Moderation in the Fediverse. *arXiv*. Available from: <https://arxiv.org/abs/2501.05871>.
- [15] Alrubaian M, Al-Qurishi M, Omar S, Mostafa MA. DeepTrust: A Deep Learning Approach for Measuring Social Media Users Trustworthiness. *ArXiv*. 2021;abs/2101.07725.
- [16] Feng S, Tan Z, Li R, Luo M. Heterogeneity-aware Twitter Bot Detection with Relational Graph Transformers. In: *AAAI Conference on Artificial Intelligence*; 2021. p. 1–8. Available from: <https://api.semanticscholar.org/CorpusID:237428699>.
- [17] Tzoumanekas G, Chatzianastasis M, Ilias L, Kiokes G, Psarras J, Askounis D. A graph neural architecture search approach for identifying bots in social media. *Frontiers in Artificial Intelligence*. 2024;7. <https://doi.org/10.3389/frai.2024.1509179>.
- [18] Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M. 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*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee; 2017. p. 963–972. Available from: <https://doi.org/10.1145/3041021.3055135>.
- [19] Ferrara E, Varol O, Davis C, Menczer F, Flammini A. The rise of social bots. *Commun ACM*. 2016 Jun;59(7):96–104. <https://doi.org/10.1145/2818717>.
- [20] Subrahmanian VS, Azaria A, Durst S, Kagan V, Galstyan A, Lerman K, et al. The DARPA Twitter Bot Challenge. *Computer*. 2016 Jun;49(6):38–46. <https://doi.org/10.1109/MC.2016.183>.
- [21] Martín-Gutiérrez D, Hernández-Peñaloza G, Hernández AB, Lozano-Diez A, Álvarez F. A Deep Learning Approach for Robust Detection of Bots in Twitter

Using Transformers. IEEE Access. 2021;9:54591–54601. <https://doi.org/10.1109/ACCESS.2021.3068659>.

- [22] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers); 2019. p. 4171–4186.
- [23] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR. 2019;abs/1907.11692. [1907.11692](https://arxiv.org/abs/1907.11692).
- [24] Grimme C, Assenmacher D, Adam L. Changing Perspectives: Is It Sufficient to Detect Social Bots? In: Social Computing and Social Media. User Experience and Behavior: 10th International Conference, SCSM 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part I. Berlin, Heidelberg: Springer-Verlag; 2018. p. 445–461. Available from: https://doi.org/10.1007/978-3-319-91521-0_32.
- [25] Varol O, Ferrara E, Davis C, Menczer F, Flammini A. Online human-bot interactions: Detection, estimation, and characterization. In: Proceedings of the International AAAI Conference on Web and Social Media. AAAI Press; 2017. p. 280–289. Available from: <https://arxiv.org/abs/1703.03107>.
- [26] Feng S, Wan H, Wang N, Luo M. 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. ASONAM '21. New York, NY, USA: Association for Computing Machinery; 2022. p. 236–239. Available from: <https://doi.org/10.1145/3487351.3488336>.
- [27] He B, Yang Y, Wu Q, Liu H, Yang R, Peng H, et al. Dynamicity-aware social bot detection with dynamic graph transformers. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. IJCAI '24; 2024. p. 5844–5852. Available from: <https://doi.org/10.24963/ijcai.2024/646>.
- [28] Wang X, Chen K, Wang K, Wang Z, Zheng K, Zhang J. FedKG: A Knowledge Distillation-Based Federated Graph Method for Social Bot Detection. Sensors. 2024;24(11). <https://doi.org/10.3390/s24113481>.
- [29] Yang Y, Yang R, Peng H, Li Y, Li T, Liao Y, et al. FedACK: Federated Adversarial Contrastive Knowledge Distillation for Cross-Lingual and Cross-Model Social Bot Detection. In: Proceedings of the ACM Web Conference 2023. WWW '23. New York, NY, USA: Association for Computing Machinery; 2023. p. 1314–1323. Available from: <https://doi.org/10.1145/3543507.3583500>.

- [30] Pei H, Wei B, Chang KCC, Lei Y, Yang B. Geom-GCN: Geometric Graph Convolutional Networks. In: International Conference on Learning Representations (ICLR 2020); 2020. p. 1–10. Available from: <https://openreview.net/forum?id=S1e2agrFvS>.
- [31] Zhu J, Yan Y, Zhao L, Heimann M, Akoglu L, Koutra D. Beyond homophily in graph neural networks: current limitations and effective designs. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20. Red Hook, NY, USA: Curran Associates Inc.; 2020. p. 7793 – 7804.
- [32] He B, Jiang X, Wu Q, Liu H, Yang Y, Liao Y. Boosting Bot Detection via Heterophily-Aware Representation Learning and Prototype-Guided Cluster Discovery. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2. KDD '25. New York, NY, USA: Association for Computing Machinery; 2025. p. 860–871. Available from: <https://doi.org/10.1145/3711896.3736862>.
- [33] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. ACM; 2016. p. 785–794. Available from: <http://dx.doi.org/10.1145/2939672.2939785>.
- [34] Rumelhart DE, Widrow B, Lehr MA. The basic ideas in neural networks. Communications of the ACM. 1994;37(3):87–93.
- [35] Arif S, Pfister T. TabNet: Attentive Interpretable Tabular Learning. Proceedings of the AAAI Conference on Artificial Intelligence. 2021 5;35:6679–6687. <https://doi.org/10.1609/AAAI.V35I8.16826>.
- [36] van der Maaten L, Hinton G. Visualizing Data using t-SNE. Journal of Machine Learning Research. 2008;9(86):2579–2605.
- [37] He B, Yang Y, Wu Q, Liu H, Yang R, Peng H, et al. Dynamicity-aware social bot detection with dynamic graph transformers. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. IJCAI '24; 2024. p. 1–9. Available from: <https://doi.org/10.24963/ijcai.2024/646>.