FediScan: Collaborative Social Bot Detection in the Fediverse

Min Gao
College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
State Key Laboratory of Internet Architecture, Tsinghua University
Beijing, China
mgao21@m.fudan.edu.cn

Wen Wen
Shanghai Key Lab of Intelligent Information Processing, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
wwen24@m.fudan.edu.cn

Haoran Du
Shanghai Key Lab of Intelligent Information Processing, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
hrdu24@m.fudan.edu.cn

Qiang Duan
Department of Information Sciences and Technology, The Pennsylvania State University
Abington, PA, United States
qduan@psu.edu

Yu Xiao
Department of Information and Communications Engineering, Aalto University
Espoo, Finland
yu.xiao@aalto.fi

Yupeng Li
Department of Interactive Media, Hong Kong Baptist University
Hong Kong, China
ypengl@hkbu.edu.hk

Xin Wang
Shanghai Key Lab of Intelligent Information Processing, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
xinw@fudan.edu.cn

Pan Hui
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
Hong Kong University of Science and Technology
Hong Kong, China
panhui@ust.hk

Yang Chen*
Shanghai Key Lab of Intelligent Information Processing, College of Computer Science and Artificial Intelligence, Fudan University
Shanghai, China
chenyang@fudan.edu.cn

Abstract

The growing concern for data privacy and user autonomy has led to the rise of decentralized online social networks, such as Mastodon. Unlike centralized platforms, Mastodon's federated architecture comprises a number of independent instances. Social bots, which are automated accounts that might spread misinformation and manipulate discourse, pose significant threats to platform moderation and security. Detecting these social bots in decentralized online social networks such as Mastodon is challenging due to the fragmented governance, non-IID data distributions, and diverse modalities across their different instances. Current social bot detection methods, designed for centralized systems, fail to address these challenges while preserving user privacy. To fill this gap, we propose FediScan, a decentralized federated learning framework for social bot detection in the Fediverse. FediScan introduces three key innovations: (1) a modality-specific data augmentation module integrating a feature augmentation strategy and a multimodal encoder with a gated attention mechanism to learn informative user representations for robust social bot detection; (2) a semantic-aware communication protocol incorporating an instance hypergraph built upon hashtag co-occurrence, enabling knowledge sharing without exchanging raw data; and (3) an asynchronous aggregation strategy to accelerate convergence and reduce overhead. Extensive evaluation on a representative multimodal dataset from Mastodon demonstrates that FediScan achieves a significant improvement in F1-score over existing methods. This work introduces a novel approach for privacy-preserving, collaborative detection of social bots within decentralized online social networks.

CCS Concepts

• **Security and privacy → Social network security and privacy**; • **Computing methodologies → Machine learning**.

Keywords

Decentralized Online Social Networks; Mastodon; Social Bot Detection; Decentralized Federated Learning

ACM Reference Format:
Min Gao, Wen Wen, Haoran Du, Qiang Duan, Yu Xiao, Yupeng Li, Xin Wang, Pan Hui, and Yang Chen. 2026. FediScan: Collaborative Social Bot Detection in the Fediverse. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3774904.3792690

[*] Corresponding Author: Yang Chen.

WWW '26, Dubai, United Arab Emirates

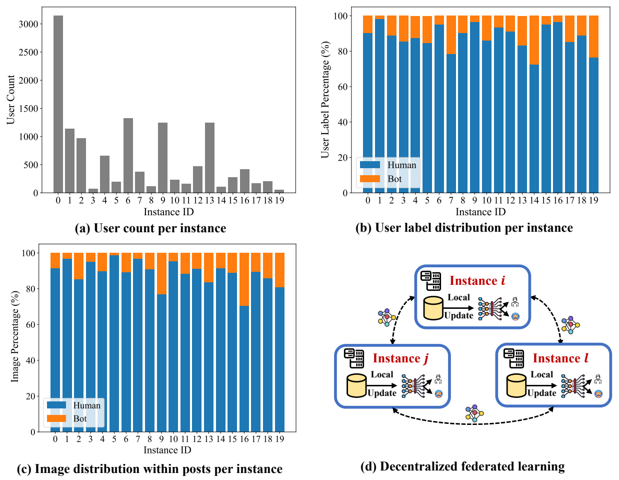
ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792690

1 Introduction

Mastodon, as part of the Fediverse [3, 4][1], has emerged as a leading alternative to centralized online social networks like Twitter (X) and Facebook [27, 67]. Its distributed architecture, where each instance operates independently with its own moderation policies, user base, and governance structures [7, 20, 28, 40, 42], empowers users with control over their data and attracts a growing number of users seeking a more private and user-centric experience. However, this decentralization creates a paradox: while it enhances privacy and autonomy, it also introduces new challenges to governance mechanisms [50]. A critical challenge in this ecosystem is the proliferation of social bots, automated accounts capable of spreading misinformation, manipulating political discourse, and degrading user experience. According to [66], approximately 5% of Mastodon users are estimated to be social bots, posing a significant threat to the platform’s integrity. Unlike centralized platforms that employ dedicated teams for content moderation, Mastodon operates as a decentralized network consisting of thousands of independently managed instances. These instances differ in values, cultures, and governance rules. Most instances are operated by volunteer administrators who often lack advanced moderation tools, unified detection policies, or access to user data from other instances [4]. This fragmented structure makes it difficult to coordinate the detection and management of potential social bots [4]. Consequently, it is imperative to develop and implement an effective approach for social bot detection to enhance the governance of the entire Mastodon platform.

Recent studies [16, 59, 60] have focused on social bot detection in centralized platforms like Twitter (X) and Weibo. Among them, graph neural networks (GNNs)-based methods have achieved leading performance by integrating multimodal data, including user metadata, textual content, and social graph structures [16, 37, 60]. However, these approaches assume complete access to user features and social connections. However, such an assumption does not hold in decentralized scenarios. In these platforms, user data is distributed across independently operated instances, each governed by distinct moderation policies [8, 65]. Even with collected data, existing models face challenges in generalizing across different instances. This is due to the variation in user behaviors, including posting preferences and language styles. Therefore, new approaches are needed to handle the decentralized nature of user data to detect social bots in decentralized platforms.

To develop an effective social bot detection framework in the Fediverse, several key challenges need to be solved. (1) *Privacy-preserving and decentralized learning.* Since user-generated content (UGC) is distributed across independently operated instances and there is no central coordinating server, aggregating all data in a central server is neither feasible nor privacy-compliant. Consequently, decentralized federated learning (DFL) [24, 53] emerges as a promising solution. However, applying DFL in this setting is challenging. Unlike traditional FL scenarios, where heterogeneity typically comes from individual users or similar organizations, the Mastodon instances exhibit a significant variety in user demographics, posting preferences, and moderation policies, leading to highly non-independent and non-identically distributed (non-IID) data [65]. Each instance serves as an independent community with its own moderation policies and content preferences. This ever-changing diversity increases distribution variations and poses significant challenges for model convergence and generalization. (2) *Label imbalance and cross-instance modality heterogeneity.* Each instance holds multimodal data such as text, images, and metadata, with distinct distributions in quantity, labeling, and semantic distribution. To analyze this heterogeneity, we use FediData [17][2], a publicly available multimodal Mastodon dataset that includes textual posts, images, user profiles, and social links. This dataset was collected via Mastodon’s official REST API[3] between October 25-30, 2024. Based on this dataset, we select the top-20 instances by user count and analyze the distribution of data heterogeneity (see Figure 1). This heterogeneity exists not only between instances but also within individual instances across modalities, making it critical to design a framework that adapts to these variations. (3) *Peer selection for semantically meaningful knowledge sharing.* In DFL settings, instances can exchange parameters with other instances (peers[4]) following a predefined network topology [33, 44, 62][5] to improve generalization and accelerate model convergence. However, using random or topology-based methods [62] might connect semantically dissimilar instances, resulting in noisy knowledge sharing and inefficient model convergence. For instance, an image-dominated instance collaborating with a text-dominated instance might introduce noisy updates due to modality mismatch, leading to suboptimal performance.



(a) User count per instance

(b) User label distribution per instance

(c) Image distribution within posts per instance

(d) Decentralized federated learning

Figure 1: Analysis of data heterogeneity (In subfigures (a)-(c), we display the distribution across user counts, labels, and images within UGC data for the top-20 instances from the FediData dataset, respectively. Subfigure (d) is an overview of decentralized federated learning.)

[1] The Fediverse, short for “federated universe”, is a network of decentralized online social platforms, where most of them can communicate with each other via decentralized social networking protocols such as ActivityPub.

[2] https://zenodo.org/records/15621243

[3] https://docs.joinmastodon.org/client/intro/

[4] In this paper, a peer denotes an instance that directly communicates with other instances to exchange model parameters.

[5] In DFL, the network topology defines how participants are connected. The participants involved in the learning process typically collaborate only with their neighbors.

To tackle these challenges, we propose FediScan, a decentralized federated learning framework designed for detecting social bots in decentralized online social networks such as Mastodon. FediScan features a modality-specific data augmentation technique and a semantic-aware communication protocol to facilitate efficient peer selection and perform bot detection in a decentralized, asynchronous manner. To tackle the first challenge, we introduce a decentralized federated learning (DFL) framework (see Figure 1(d)), which allows each instance to learn multimodal representations and identify social bots without exposing raw data. For the second challenge, we incorporate multiple modalities, such as text, images, and user metadata. We introduce a feature augmentation strategy and a gated attention mechanism to address the label imbalance issue and learn a unified multimodal representation for social bot detection. For the third challenge, we introduce a semantic-aware communication protocol that helps each instance select peers with semantically meaningful knowledge for parameter exchange. Furthermore, we adopt an asynchronous aggregation strategy to enable peer-to-peer communication for model updates, enhancing the efficiency of our model.

In summary, our contributions are threefold:

- To our best knowledge, we are the first to focus on social bot detection in decentralized online social networks and also the first to consider the image modality for this task. We have validated the significance of the image modality, highlighting that this modality could contain rich semantic knowledge, which was ignored by previous studies.
- We propose FediScan, a decentralized federated learning framework for social bot detection. We introduce a modality-specific data augmentation strategy to handle the label imbalance and multimodal heterogeneity issues. Furthermore, we introduce a semantic-aware communication protocol with an instance hypergraph based on the co-hashtag information across different instances. By leveraging these two designs, FediScan supports informative multimodal representation and efficient peer-to-peer communication.
- We have conducted extensive experiments on a comprehensive multimodal dataset from Mastodon. These findings indicate that FediScan outperforms existing methods by a significant improvement in F1-score for social bot detection, while maintaining an acceptable communication cost.

2 Related Work

Social bot detection. Existing social bot detection methods mainly target centralized online social networks, such as Twitter (X) and Weibo. For each platform, all user data, including user profiles, social relationships, and UGC data, is centrally stored. Therefore, early efforts in social bot detection focused mainly on extracting informative features from user data and utilizing machine learning techniques to identify bots [9, 18, 23, 29, 30, 35, 52, 57]. These studies extract statistical and textual features from user profile information, UGC data, and user behavior using statistical studies and some approaches based on natural language processing (NLP) [2, 21]. These extracted features are then fed into classical machine learning models or simple temporal learning models to identify social bots. Recent graph neural networks (GNNs)-based methods [14, 16, 38, 60] introduce GNNs for effective social bot detection due to their superiority in modeling network structures. For example, Feng et al. [16] introduced a relational graph neural network (R-GCN) to learn user representations. To adaptively fuse information from neighbors, the self-attention mechanism was utilized in the study [14]. Subsequent studies [60] considered hierarchical structural information by introducing contrastive learning for social bot detection. Ling et al. [34] considered enhancing the extracted user features and leveraging the edge similarity information to improve social bot detection. Although the above methods have achieved success in centralized scenarios, it is difficult for them to be directly applicable to decentralized online social networks like Mastodon, where user data is distributed among different instances. Moreover, previous studies [16, 34, 59] have relied heavily on textual content, user profile metadata, and social graph structures, and they have omitted the image modality, which plays a crucial role in bot behaviors. Bots often exploit coordinated visual deception, such as using identical profile pictures or memes to amplify their influence [25, 45]. The lack of image modeling might result in incomplete representations and hinder detection performance. In this work, we consider the image modality and decentralized nature of social bot detection on Mastodon.

Content moderation. Decentralizing the web is an appealing yet difficult objective. A key difficulty lies in implementing decentralized content moderation that can withstand a range of adversarial actors. Several efforts have been made to address this issue. Zia et al. [8] have characterized the spread of toxicity on the platform, confirming that the federation process allows toxic content to spread between instances. They have further explored the challenges of moderating this process by building per-instance models. They found that federated toots constitute the most significant chunk of toxic content in 26/30 of the instances. Moreover, 60% of toxic toots get more than one reblog, compared with only 16% of non-toxic toots. This trend indicates that interest and uptake in toxic materials are consistently higher. Hassan et al. [4] observed a diversity of administrator strategies, with evidence that administrators on larger instances struggle to find sufficient resources. They then proposed a tool, WatchGen, to semi-automate the process. Agarwal et al. [3] noticed that each server only has a partial view of an entire conversation because conversations are often federated across servers in a non-synchronized fashion. To address this, this work proposed a decentralized conversation-aware content moderation approach suitable for the Fediverse. Zhang et al. [64] conducted semi-structured interviews with 16 Mastodon instance administrators, including those who host instances to support marginalized and stigmatized communities, to understand their motivations and lived experiences of running decentralized online social networks. Zia et al. [65] presented FedMod, which utilized a federated learning method for collaborative content moderation. Although prior studies have explored potential solutions for content moderation on Mastodon, they have not focused on the special users, social bots. None of their solutions could be directly used to identify social bots. Unlike existing content moderation studies based on textual data, this study focuses on the social bot detection task, which is more challenging due to its multimodal nature.

Decentralized federated learning. Decentralized federated learning (DFL) [24, 53, 55, 62] is a federated learning paradigm that operates without a central server, where nodes directly exchange model updates via a peer-to-peer communication mechanism. Compared with centralized FL [31], DFL utilizes resources to aggregate model parameters across all participating nodes. Additionally, DFL enhances the robustness of the network and mitigates the risk of a single point of failure. Recent studies primarily focus on asynchronous communication mechanisms [24] and network topology algorithms [41, 62]. In this work, we have leveraged a decentralized FL framework and defined an instance hypergraph to support peer selection without employing traditional network topology for communication.

3 FediScan Framework

In this section, we first formulate the definition of the social bot detection problem in decentralized online social networks in Section 3.1. Then, we introduce the overall framework of our method, named FediScan, in Section 3.2, followed by a detailed introduction of a modality-specific data augmentation in Section 3.3 and a semantic-aware communication protocol in Section 3.4, respectively.

3.1 Preliminaries

In Mastodon, each instance holds a subset of users and their multimodal data. Let K denote the total number of instances, and let $D_k = \{X_k^m, X_k^t, X_k^v\}$, where X_k^m, X_k^t, and X_k^v are the metadata, textual content, and images, respectively. The Y_k represents the local user labels at instance k, where 0 stands for human and 1 stands for bot. Our goal is to train a decentralized social bot detection model θ via decentralized FL learning, where each instance collaborates with selected peers to optimize a multimodal encoder and a classifier. The learning objective is

$$\min_{\theta} \sum_{k=1}^{K} L_k(\theta) = \sum_{k=1}^{K} \mathbb{E}_{(X_k, Y_k) \sim \mathcal{D}_k} \left[L\left(F_k\left(E(X_k), \theta \right), Y_k \right) \right],$$

where $E\left(X_k\right)$ is the multimodal encoder, $F_k(\cdot)$ is the classifier for instance k, and $L_k(\theta)$ is the loss function of the instance k.

3.2 Overview of FediScan

As illustrated in Figure 2, FediScan is designed to address the challenges of modality heterogeneity and privacy preservation in decentralized social bot detection. It consists of two core components, a modality-specific data augmentation (see Section 3.3) and a hypergraph-based communication protocol (see Section 3.4), which work synergistically to enable efficient and privacy-preserving collaborative learning. The former aligns heterogeneous multimodal features (text, metadata, and images) into a unified latent space, allowing the model to learn discriminative patterns between humans and bots. The latter introduces a semantic-aware communication protocol to enhance collaboration efficiency, where instances communicate only with semantically aligned peers via an asynchronous protocol based on a hypergraph structure determined by co-hashtag relationships.

3.3 Modality-Specific Data Augmentation

To effectively tackle the challenges posed by modality heterogeneity across different instances, we introduce a modality-specific data augmentation module. Specifically, for each instance, this module aims to enhance and integrate local multimodal data (user metadata, textual data, and images) into a unified latent space for robust social bot detection. We will present several key designs of this component.

3.3.1 *Multimodal Encoder.*

The multimodal encoder serves as a cross-modal feature extractor, and it maps raw multimodal data into a unified latent space. A gated attention mechanism is utilized to dynamically adjust weights across modalities to ensure robust feature alignment. We will introduce how we encode raw multimodal data.

Metadata encoder. Based on user metadata, we consider six numerical (including "follower_count", "following_count", "statuses_count", "account_name_length", "notes_length", and "activate_days") and four categorical metadata features ("locked", "discoverable", "islocal", "ismastodon") based on user profile information. After performing z-score normalization [16], we apply a two-layer Multi-Layer Perceptron (MLP) to learn representations for user metadata x^m.

Textual encoder. To make good use of textual content, we consider the original text data from user posts as textual information. We used the DistilBERT model [51][6] to learn the textual features of the users. DistilBERT is a transformer-based model and faster than BERT for inference or downstream tasks. The process can be described as

$$T_u^a = Concat(T_u), \quad x_u^t = DistilBERT(T_u^a), \tag{1}$$

where T_u^a is the combined textual data of user u with original texts T_u, and x_u^t is the learned textual representations of user u. We denote the textual representations of all users as x^t. $Concat(\cdot)$ refers to the concatenation operation, and $DistilBERT(\cdot)$ denotes the use of the DistilBERT model to learn textual embeddings.

Visual encoder. We consider both user avatars and images contained in UGC data. For these visual contents, we encode them with the DeiT-III model [56][7] individually, which can be defined as

$$x_u^v = \frac{1}{m} \text{DeiT-III}\left(Stack(v_u^1, v_u^2, \ldots, v_u^m) \right), \tag{2}$$

where v_u^k refers to the k-th image of the user u, $Stack(\cdot)$ is the stacking operation, and m is the length of the input image sequences. We then concatenate the image features obtained from the user avatars and the user UGC data and feed them into a two-layer MLP to obtain visual representations of the users x^v.

3.3.2 *Feature Augmentation.*

To alleviate the label imbalance issue in our task, we introduce a feature augmentation strategy that dynamically enriches minority class users in each modality.

Taking the user textual embedding x^t as an example, the augmentation process is formalized as follows. The first step is minority class identification. Given the user labels Y, for each class $c \in C$, we first compute its frequency $n_c = \sum_{i \in [1,\cdot,n]} \mathbb{1}\{y_i = c\}$, where $\mathbb{1}\{\cdot\}$ is the indicator function. The minority user class c_{min} is determined by $c_{min} = \text{argmin}_{c \in C} n_c$. Then, we employ a Gaussian noise augmentation. For the textual embedding $x_i^t \in X_{c_{min}}$ of user i, we generate synthetic samples by perturbing the original features with Gaussian noise, denoted as $\epsilon_i^t \sim \mathcal{N}^t\left(0, \sigma^2 I_d\right)$, where $\mathcal{N}^t$

[6] https://huggingface.co/distilbert/distilbert-base-multilingual-cased

[7] https://huggingface.co/timm/deit3_small_patch16_224.fb_in22k_ft_in1k

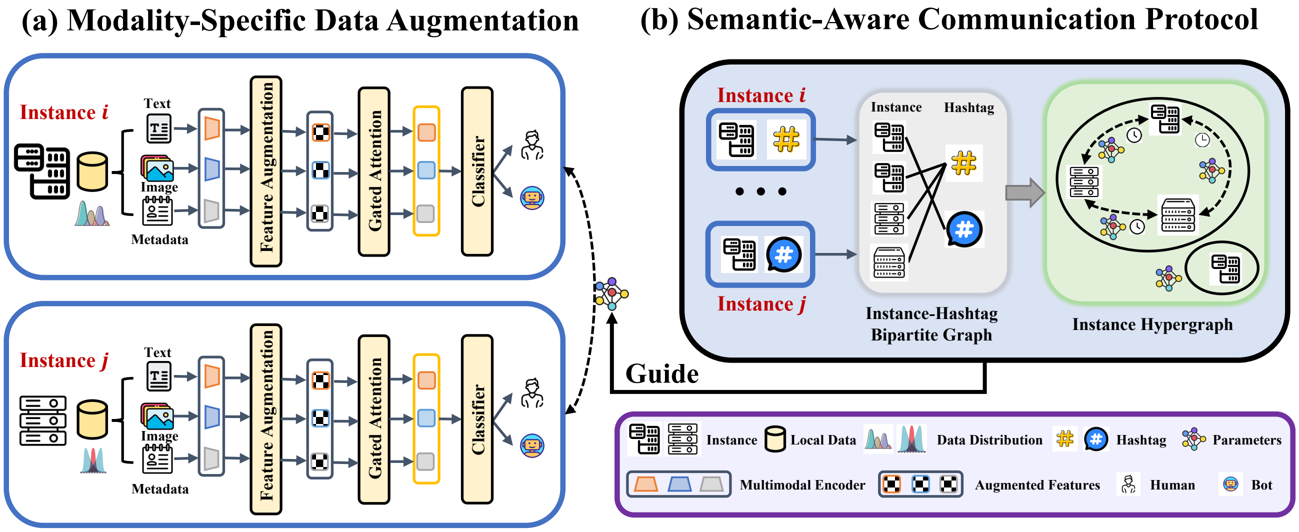


Figure 2: FediScan framework for decentralized social bot detection. It has two modules: (a) modality-specific data augmentation via feature augmentation and gated attention mechanism, and (b) semantic-aware communication protocol enabling asynchronous, semantically aligned parameter exchange to overcome modality-heterogeneity challenges.

represents the distribution with σ to control the noise scale to maintain feature consistency, and I_d denotes the identity matrix whose size matches the feature dimension. The augmented feature $x_i^{t'}$ is computed as $x_i^{t'} = x_i^t + \boldsymbol{\epsilon}_i^t$, where $x_i^{t'}$ and x_i^t are the augmented features and original features, respectively. Similarly, we obtain the corresponding labels $y_i^{t'} = y_i^t$, where y_i^t is the original user label. Finally, we concatenate the augmented textual features and labels with the original data to enhance the user data via $X_i^t = \left[x_i^t; x_i^{t'}\right]$, and $Y_i^t = \left[y_i^t; y_i^{t'}\right]$. We denote the augmented textual features of all users as X^t. Similarly, we can obtain the augmented metadata features X^m and augmented image features X^v.

3.3.3 Gated Attention. To effectively fuse multimodal features, we introduce a gated attention mechanism. By leveraging a learnable gating attention mechanism, the multimodal encoder could dynamically assign weights to different modalities for adapting to variations in data distribution across instances. Specifically, we compute modality attention weights via

$$\left[W^m || W^t || W^v\right] = \sigma(\mathrm{MLP}\left(\left[X^m || X^t || X^v\right]\right)), \tag{3}$$

where W^m, W^t, and W^v are learnable modality weights of metadata, textual, and visual data, respectively. X^m, X^t, and X^v are augmented metadata representation, textual representation, and visual representation, respectively. $||$ denotes the concatenation function. The unified user representation can be obtained by $X = W^m \cdot X^m + W^t \cdot X^t + W^v \cdot X^v$. This design not only maintains the complementary nature of multimodal data but also improves our model's ability to handle modality heterogeneity by learning adaptive weights. The unified user representation X enhances collaboration across different instances and enables robust social bot detection.

3.3.4 Classifier. Based on the unified latent user representation, we implement the final classification layer using a multi-layer perceptron with two fully connected layers, with the first layer followed by a ReLU activation function and a dropout layer to distinguish social bots from humans. This process can be represented as

$$\mathbf{h}_1 = \mathrm{ReLU}(W_1 X + b_1), \mathbf{h}_2 = \mathrm{Dropout}(\mathbf{h}_1), \overline{\mathbf{Y}} = W_2 \mathbf{h}_2 + b_2, \tag{4}$$

where W_1, W_2, b_1, and b_2 are learnable parameters, ReLU and Dropout are the activation function and the dropout layer. $\overline{\mathbf{Y}}$ is the final predicted user labels. Here, the dropout layer is equipped with a rate of 0.2 to mitigate overfitting. Considering the label imbalance issue, we utilize the weighted cross-entropy loss to optimize the model, denoted as $\mathcal{L}_{\mathrm{WCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot \mathbb{1}\{y_i = c\} \cdot \log \hat{p}_{i,c}$, where N and C are the number of users and classes, respectively. The w_c represents the weight assigned to user class c, and $\hat{p}_{i,c}$ refers to the predicted probability of sample i belonging to class c, and $\mathbb{1}\{y_i = c\}$ is the indicator function, where it equals 1 if $y_i = c$, otherwise 0.

3.4 Semantic-Aware Communication Protocol

In this section, we introduce a semantic-aware communication protocol, which is designed to facilitate semantic alignment and efficient collaboration across decentralized instances. This protocol allows each instance to exchange parameters with semantically similar peers during training. Inspired by [65], we first extract hashtag information from each instance and then construct an instance-hashtag bipartite graph based on the co-occurrence of hashtags among instances. This bipartite graph is then transformed into a hypergraph, where each hyperedge connects instances that share the same hashtags. This design avoids noisy updates from mismatched modality distributions and adapts to evolving data patterns. In the following, we elaborate on the key designs of this component, including the construction of the instance-hashtag bipartite graph and instance hypergraph, and the asynchronous aggregation strategy.

3.4.1 Instance-Hashtag Bipartite Graph Construction. In Mastodon, hashtags are words preceded by the "#" symbol used to categorize posts by topic. Therefore, hashtags serve as a critical source of semantic information, motivating us to leverage this knowledge in our semantic-aware communication protocol. Firstly, we collect the hashtags for each instance by aggregating them from tweets. Then, we construct a co-hashtag matrix H across all instances, where $H_{ij} = 1$ if hashtag j appears in instance i, and $H_{ij} = 0$ otherwise. Based on this matrix, we construct an instance-hashtag bipartite graph, where nodes are instances and hashtags, and edges exist between instances and hashtags.

3.4.2 Instance Hypergraph Construction. Based on the instance-hashtag bipartite graph, we construct an instance hypergraph to capture semantic relationships across different instances. In this hypergraph, hyperedges are formed by grouping instances that share semantically coherent hashtag patterns, derived from the instance-hashtag bipartite graph. This structure enables each instance to communicate and collaborate with peers exhibiting similar interests, enhancing the robustness of social bot detection while preserving privacy and reducing noisy interactions.

3.4.3 Asynchronous Aggregation. Considering the dynamic data distribution across different instances, FediScan employs an asynchronous aggregation strategy to optimize collaborative social bot detection.

Firstly, at the beginning of each round, each instance trains its personalized local model using a weighted cross-entropy loss to address class imbalance. The training order is randomized to ensure no global synchronization is required. After local training, each client aggregates shared parameters (the encoder and classifier) from its neighbors in the instance hypergraph. By avoiding sharing all parameters, FediScan reduces communication costs. We adopt an asynchronous Gossip protocol [10, 49], where instances exchange model updates independently and without waiting for global synchronization. This design aligns with the decentralized environment of Mastodon, where instances may join or leave unpredictably. The aggregation process is guided by modality-aware similarity weights derived from the instance hypergraph. Specifically, each instance i updates its shared parameters by considering both its own representations and those of its top-k neighbors using

$$\Theta_i^{\mathrm{r}} = \frac{1}{\sum_{j \in \mathcal{N}_i} W_{ij} + W_{ii}} \left(\sum_{j \in \mathcal{N}_i} W_{ij} \cdot \theta_j + W_{ii} \cdot \theta_i \right), \tag{5}$$

where r denotes the current round, and N_i is the set of top-k neighbors for instance i. W_{ij} represents the Jaccard similarity between instance i and instance j. $W_{ii} = 1$ is the self-weight to preserve local updates. The personalized local model Θ_i^r remains trained independently and is only used during the detection phase. This strategy reduces the communication overhead through lightweight parameter exchange and asynchronous updates, while similarity-based neighbor selection ensures high-quality aggregation.

4 Experiments

In this section, we comprehensively evaluate FediScan by answering the following research questions (RQs):

- **RQ1.** How does FediScan perform in social bot detection in decentralized scenarios?
- **RQ2.** How does each component contribute to the performance of FediScan?
- **RQ3.** What about FediScan's learning efficiency?
- **RQ4.** What are the effects of hyperparameter values on the detection performance?

4.1 Experimental Setup

Dataset Information. In this study, we use one representative multimodal dataset named FediData [17], which was collected from Mastodon, to evaluate the effectiveness of our method. This dataset comprises 64,345 unique users from 493 instances with 725,282 posts and 765,019 images. It also provides a sampled dataset, which contains manually annotated user labels (whether each user is a human or a bot)[8]. We adopted the sampled subgraph comprising 12,548 users, with a total of 208,395 posts and 222,850 images. The sampled dataset consists of 10,613 humans, 1,109 social bots, and 826 background users[9].

Baselines and Evaluation Metrics. We compare FediScan with several representative baselines. To our best knowledge, there is no existing solution that can directly detect social bots in decentralized platforms. Therefore, we consider the following methods for comparison.

- **Bot detection methods.** We consider representative social bot detection methods, including SGBot [59], BotRGCN [16], and BotBR [34]. Since they are originally designed for centralized settings, we have modified each of them to fit within a federated learning framework. In this setup, various clients (instances) collaboratively train the model without sharing raw data. We evaluate each method individually within its instance and then aggregate the results across these instances for comparison.
- **Federated learning strategies.** We consider representative FL strategies such as FedAvg [39], FedProx [31], FedPer [5], FedBN [47], and FedSea [54]. Among them, FedAvg [39] is a fundamental federated learning algorithm in which the server averages the updates of local models on client devices. FedProx [31] improves the local model training and update under heterogeneity by adding an optimization item. FedPer and FedBN are designed for personalized FL. FedSea [54] is a representative federated multimodal learning method.
- **Decentralized FL methods.** We choose two representative decentralized FL strategies, D-PSGD [32] and SGP [6] for comparison. Additionally, we consider FedMod [63] for comparison, which is the state-of-the-art decentralized content moderation method.

Because our task involves binary classification with an imbalanced label distribution, we adopt a comprehensive set of evaluation metrics, including precision, recall, and F1-score. We utilize the macro F1-score as our main evaluation metric to fairly assess the model's effectiveness in both bot and human classes. These metrics align with prior studies on social bot detection [16, 34, 37]. In addition to detection performance, computational efficiency is essential for decentralized online social networks. Therefore, we further assess training time, communication cost, and memory cost to evaluate each method's efficiency. These metrics allow us to analyze the trade-off between detection performance and system efficiency.

[8] According to [17], the labels were independently annotated by three researchers based on key criteria (e.g., automated posting, duplicate content) from [15]. The annotation quality is high, with a Cohen's kappa [11] value of 0.876.

[9] In this work, the background users refer to the unlabeled nodes that are linked to labeled users. These nodes are involved in training but are excluded from evaluation.

Implementation Details. In this study, we implemented a DFL framework for social bot detection. Following [17], we first selected instances with more than 10 accounts (including humans and bots) to ensure that each instance participating in decentralized FL has a sufficient user base. Subsequently, we adopted a stratified sampling strategy to divide users in each instance into training (50%), validation (20%), and test (30%) sets, which ensures consistency of the distribution of the users of each class across different instances. To support user alignment in our experiment, we design a global node index mapping so that nodes between different instances can be identified correctly in the federation aggregation process. We select the longest tweets for each user and consider 50 hashtags to construct the instance-hashtag bipartite graph. All features are normalized. The dimension of the user representation is 128. We use EasyGraph [19, 61] to build a hypergraph and adopt PyTorch [46] and FedML [22] to implement all models and optimize them with the AdamW optimizer [26]. We set the number of local epochs and the number of communication rounds as 5 and 10, respectively. The learning rate is set as 0.001. We report the mean test performance over 3 trials. We conducted all experiments on a Linux server equipped with the Intel Xeon CPU E5-2683 v3 @ 2.00GHz with 256GB of memory and 56 CPU cores.

4.2 Detection Performance

We compare FediScan with the baseline models in terms of precision, recall, and F1-score. The performance of different methods on the dataset is presented in Table 1. Some key findings are below:

- The compared centralized social bot detection methods leverage both multimodal user features and the social relationship information. Among these baselines, SGBot shows a relatively poor performance in centralized settings, while BotBR achieves a higher F1-score value than other methods. This could be attributed to the fact that it leverages the edge confidence information between two users. Compared with these methods, our approach achieves better performance by leveraging the feature augmentation strategy to enhance the multimodal representations without requiring complete graph structures.
- Among these representative FL strategies, we found that none of these methods achieves the best performance across all three metrics. For example, FedAvg obtains higher precision and F1-score values, while FedSea obtains higher recall values. This highlights that considering personalized FL is significant for our task, as the high heterogeneity of user data presents challenges for the model to learn global knowledge.
- Among decentralized FL methods, we observed that the FedMod method achieves higher performance in terms of precision, recall, and F1-score. These findings suggest that the text itself carries useful information, and that the decentralized FL framework helps distinguish content produced by bots.
- FediScan leverages a modality-specific data augmentation strategy to learn unified user representations and enables each instance to select semantically similar peers guided by a semantic-aware communication protocol, consistently outperforming all baselines regarding all three evaluation metrics. This indicates that augmenting multimodal representations and selecting semantically similar peers are significant for collaborative social bot detection by tackling the label imbalance and modality heterogeneity issue.

Table 1: Detection performance of all methods (Mean ± standard deviation. The best results are in bold.)

Category	Model	Precision	Recall	F1-score
Centralized	SGBot	0.1040±0.0232	0.0634±0.0317	0.0630±0.0232
	BotRGCN	0.4863±0.0572	0.5134±0.0610	0.5071±0.0256
	BotBR	0.5072±0.0788	0.5510±0.0880	0.5129±0.0013
FL	FedAvg	0.4965±0.0004	0.5667±0.0005	0.5241±0.0005
	FedProx	0.3638±0.0452	0.5003±0.0211	0.2820±0.0271
	FedPer	0.1057±0.0103	0.4287±0.0088	0.1372±0.0113
	FedBN	0.4267±0.0623	0.5256±0.0356	0.4023±0.0596
	FedSea	0.1672±0.0620	0.5914±0.0154	0.1947±0.0216
DFL	D-PSGD	0.3997±0.0509	0.5113±0.0348	0.4057±0.0583
	SGP	0.3094±0.0281	0.5083±0.0162	0.3394±0.0275
	FedMod	0.5026±0.0319	0.5704±0.0252	0.5285±0.0297
Ours	FediScan	**0.5818±0.0171**	**0.6076±0.0221**	**0.5848±0.0193**

4.3 Ablation Study

We perform an ablation study to assess the impact of various components by designing several variants. The corresponding results are reported in Figure 3.

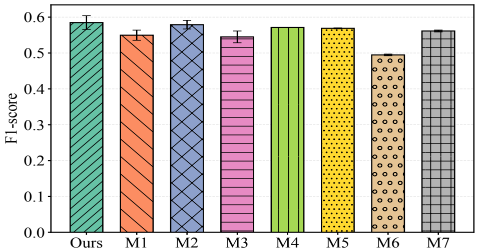


Figure 3: Ablation study

- To study the impact of the feature augmentation technique, we remove this module and utilize the original multimodal representations in FediScan. We denote this variant as M1. Removing feature augmentation leads to a sharp decrease in performance, indicating the importance of feature augmentation for social bot detection.
- To evaluate the effect of the multimodal encoder for both textual and visual data, we replace the original textual encoder DistilBERT and visual encoder DeiT-III with RoBERTa [36] and CLIP [48] models for comparison. We denote this variant as M2. One could observe that replacing these two encoders with two other encoders leads to a similar performance in F1-score, suggesting that our approach is compatible with different classic textual encoders and visual encoders.
- To investigate the importance of the gate attention mechanism, we design a variant M3 by replacing the gated attention mechanism with a simple concatenation operation followed by a single linear layer. Removing the gated attention mechanism shows a decrease in performance, highlighting the importance of this component.
- To test the influence of the semantic-aware communication protocol, we design a variant M4 by replacing the peer selection guided by the instance hypergraph with random peer selection for each instance. Variant M4 shows a decrease in F1-score, indicating the importance of selecting semantically similar peers for communication.

- To evaluate the effect of different types of modality data, we remove each modality data, including metadata, textual content, and visual content, leading to variants M5, M6, and M7, respectively. Compared with the complete model, it is obvious that excluding any single data modality could lead to reduced performance, highlighting the importance of each type of information for social bot detection.

4.4 Learning Efficiency Analysis

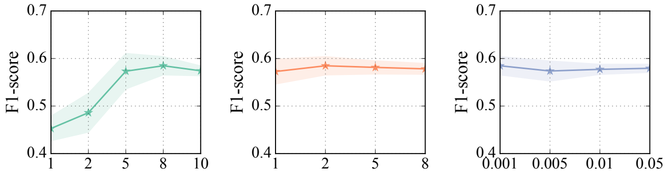
As shown in Table 2, we compare the learning efficiency of different methods in social bot detection. FediScan takes 1.59 seconds per round for training, which is acceptable for our task. Regarding the memory cost, our method requires 399.99 MB of memory, which could be due to the fact that FediScan introduces a relatively lightweight multimodal encoder. In terms of the communication cost, our method demonstrates an acceptable efficiency performance. Specifically, it exchanges an average of 115.59 MB per round, which is substantially lower than several other approaches, like FedAvg.

Table 2: Communication efficiency of all methods (Mean ± standard deviation. The notation "-" indicates that the method operates without requiring communication.)

Category	Model	Training Time (s)	Memory Usage (MB)	Communication Cost (MB)
Centralized	SGBot	4.96±0.12	1,167.11±32.71	-
	BotRGCN	17.83±0.42	926.21±19.67	-
	BotBR	0.17±0.03	14.85±9.38	-
FL	FedAvg	2.16±0.37	255.43±2.71	390.53±0.00
	FedProx	8.171±1.04	183.12±0.45	22.78±0.00
	FedPer	5.58±0.50	80.92±8.90	16.406±0.00
	FedBN	12.91±0.96	203.23±5.00	302.43±0.00
	FedSea	20.65±0.16	262.62±2.01	292.90±0.00
DFL	D-PSGD	6.37±0.37	181.36±0.87	62.42±0.00
	SGP	4.57±0.01	46.582±1.95	28.36±0.00
	FedMod	3.80±0.21	1,782.25±5.05	33.88±0.00
Ours	FediScan	1.59 ±0.03	399.99±0.10	115.59±0.00

4.5 Hyperparameter Sensitivity Analysis

In this section, we investigate the hyperparameter sensitivity of FediScan. We focus on three critical parameters, including (1) the number of local training epochs, (2) the top-k neighbor selection in the communication process, and (3) the noise scale of the feature augmentation strategy. Figure 4 presents the model sensitivity with respect to F1-score under varying settings.



(a) Epochs (b) Top-k neighbors (c) Noise scales

Figure 4: Sensitivity analysis of FediScan w.r.t (a) different number of epochs, (b) different number of top-k neighbors, and (c) different noise scales

Effect of local training epochs. We analyze how changing the number of local training epochs influences the detection performance of FediScan. As shown in Figure 4(a), local training epochs are set within the range $[1, 2, 5, 8, 10]$. Increasing local epochs improves the F1-score, as more local updates allow the model to better adapt to instance-specific data distributions. However, further increasing the local training epochs beyond 8 leads to a gradual decline in performance due to overfitting on local data within the high heterogeneity environment. In our experiment, the optimal value is chosen as 8, achieving a balance between local adaptation and global generalization.

Effect of top-k neighbor selection. We examine how varying the number of neighbors (top-k), ranging from $[1, 2, 5, 8]$, affects the detection performance of FediScan. All other hyperparameters are fixed, except for the number of clients. The results are shown in Figure 4(b). When k is too small, the model lacks sufficient cross-instance knowledge sharing, resulting in suboptimal performance. Conversely, a larger k introduces noisy or irrelevant neighbors, degrading performance due to misaligned modality distributions. FediScan achieves the best performance at $k = 2$, where each instance selects two suitable peers for communication.

Effect of different noise scales. The noise scale determines the augmentation scale of multimodal representations in the feature augmentation strategy. The optimal noise scale of 0.001 is chosen to guide our model. As depicted in Figure 4(c), a lower noise scale reflects that we generate more similar embeddings to the original multimodal representations. This could introduce more stable augmentation and help our model handle the label imbalance issue, thereby achieving robust social bot detection.

5 Conclusion and Future Work

In this paper, we have investigated the problem of social bot detection in decentralized online social networks. We presented FediScan, a decentralized federated learning method to address this problem. We identified three key challenges, including label imbalance, modality heterogeneity, and peer selection. We introduced a modality-specific data augmentation strategy and a semantic-aware communication protocol to tackle them. FediScan is flexible enough and could be utilized in other similar applications in decentralized online social networks [1, 12, 43]. Extensive experiments demonstrate that FediScan not only sets a new benchmark in this domain but also significantly advances the state-of-the-art in a Mastodon dataset featuring comprehensive multimodal information across all evaluation metrics.

For future work, we aim to adapt FediScan for broader practical application scenarios within the Fediverse. Also, we will further evaluate our framework in other decentralized online social networks. In addition, we plan to incorporate other modalities, such as video data, to further enhance our detection.

Acknowledgments

This work is sponsored by National Natural Science Foundation of China (No. 62072115, No. 62472101), Shanghai Science and Technology Innovation Action Plan Project (No. 22510713600), Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme (No. 2025A0505000001), the Early Career Scheme (ECS) from the Research Grants Council of HKSAR (HKBU 22202423), the Initiation Grant for Faculty Niche Research Areas 2023/24 (No. RC-FNRA-IG/23-24/COMM/01), the General Research Fund (GRF) from the Research Grants Council of HKSAR (HKBU 12203425), Research Cluster Matching Scheme of Hong Kong Baptist University (No. RCMS/24-25/01).

References

[1] Roel Roscam Abbing and Robert W Gehl. 2024. Shifting your research from X to Mastodon? Here's what you need to know. *Patterns* 5, 1 (2024).

[2] Aseel Addawood, Adam Badawy, Kristina Lerman, and Emilio Ferrara. 2019. Linguistic Cues to Deception: Identifying Political Trolls on Social Media. In *Proc. of ICWSM*, Vol. 13. 15–25.

[3] Vibhor Agarwal, Aravindh Raman, Nishanth Sastry, Ahmed M Abdelmoniem, Gareth Tyson, and Ignacio Castro. 2024. Decentralised Moderation for Interoperable Social Networks: A Conversation-based Approach for Pleroma and the Fediverse. In *Proc. of ICWSM*, Vol. 18. 2–14.

[4] Ishaku Hassan Anaobi, Aravindh Raman, Ignacio Castro, Haris Bin Zia, Damilola Ibosiola, and Gareth Tyson. 2023. Will Admins Cope? Decentralized Moderation in the Fediverse. In *Proc. of WWW*. 3109–3120.

[5] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. *arXiv preprint arXiv:1912.00818* (2019).

[6] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael G. Rabbat. 2019. Stochastic Gradient Push for Distributed Deep Learning. In *Proc. of ICML*. 344–353.

[7] Haris Bin Zia, Jiahui He, Ignacio Castro, and Gareth Tyson. 2024. Fediverse Migrations: A Study of User Account Portability on the Mastodon Social Network. In *Proc. of IMC*. 68–75.

[8] Haris Bin Zia, Aravindh Raman, Ignacio Castro, Ishaku Hassan Anaobi, Emiliano De Cristofaro, Nishanth Sastry, and Gareth Tyson. 2022. Toxicity in the Decentralized Web and the Potential for Model Sharing. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 2 (2022), 1–25.

[9] Meng Cai, Han Luo, Xiao Meng, Ying Cui, and Wei Wang. 2023. Network distribution and sentiment interaction: Information diffusion mechanisms between social bots and human users on social media. *Information Processing & Management* 60, 2 (2023), 103197.

[10] Daniel Cason, Nenad Milosevic, Zarko Milosevic, and Fernando Pedone. 2021. Gossip consensus. In *Proc. of Middleware*. 198–209.

[11] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.

[12] Christina Dunbar-Hester. 2024. Showing your ass on Mastodon: Lossy distribution, hashtag activism, and public scrutiny on federated, feral social media. *First Monday* 29, 3 (2024).

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *Proc. of USENIX Security Symposium*. 1605–1622.

[14] Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022. Heterogeneity-aware Twitter Bot Detection with Relational Graph Transformers. In *Proc. of AAAI*, Vol. 36. 3977–3985.

[15] 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. *Advances in Neural Information Processing Systems* 35 (2022), 35254–35269.

[16] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021. BotRGCN: Twitter bot detection with relational graph convolutional networks. In *Proc. of ASONAM*. 236–239.

[17] Min Gao, Haoran Du, Wen Wen, Qiang Duan, Xin Wang, and Yang Chen. 2025. FediData: A Comprehensive Multi-Modal Fediverse Dataset from Mastodon. In *Proc. of CIKM*. 6372–6376.

[18] Min Gao, Qiang Duan, Boen Liu, Yu Xiao, Xin Wang, and Yang Chen. 2025. Higher-Order Information Matters: A Representation Learning Approach for Social Bot Detection. In *Proc. of CIKM*. 675–685.

[19] Min Gao, Zheng Li, Ruichen Li, Chenhao Cui, Xinyuan Chen, Bodian Ye, Yupeng Li, Weiwei Gu, Qingyuan Gong, Xin Wang, and Yang Chen. 2023. EasyGraph: A Multifunctional, Cross-Platform, and Effective Library for Interdisciplinary Network Analysis. *Patterns* 4, 10 (2023), 100839.

[20] Luke Gassmann, Ryan McConville, and Matthew Edwards. 2024. Leading the Mastodon Herd: Analysing the Traits of Influential Leaders on a Decentralised Social Media Platform. In *Proc. of the IEEE BigData*. 2939–2948.

[21] Samar Haider, Luca Luceri, Ashok Deb, Adam Badawy, Nanyun Peng, and Emilio Ferrara. 2023. Detecting Social Media Manipulation in Low-Resource Languages. In *Companion Proceedings of the ACM Web Conference 2023*. 1358–1364.

[22] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. In *Proc. of NeurIPS 2020 SpicyFL Workshop*.

[23] Loukas Ilias, Ioannis Michail Kazelidis, and Dimitris Askounis. 2024. Multimodal detection of bots on X (Twitter) using Transformers. *IEEE Transactions on Information Forensics and Security* 19 (2024), 7320–7334.

[24] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and Hamid R Tizhoosh. 2023. Decentralized federated learning through proxy model sharing. *Nature Communications* 14, 1 (2023), 2899.

[25] Tuja Khaund, Baris Kirdemir, Nitin Agarwal, Huan Liu, and Fred Morstatter. 2021. Social Bots and Their Coordination During Online Campaigns: A Survey. *IEEE Transactions on Computational Social Systems* 9, 2 (2021), 530–545.

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

[27] Lucio La Cava, Sergio Greco, and Andrea Tagarelli. 2021. Understanding the growth of the Fediverse through the lens of Mastodon. *Applied Network Science* 6 (2021), 1–35.

[28] Lucio La Cava, Sergio Greco, and Andrea Tagarelli. 2022. Network Analysis of the Information Consumption-Production Dichotomy in Mastodon User Behaviors. In *Proc. of ICWSM*, Vol. 16. 1378–1382.

[29] Shilong Li, Boyu Qiao, Kun Li, Qianqian Lu, Meng Lin, and Wei Zhou. 2023. Multimodal Social Bot Detection: Learning Homophilic and Heterophilic Connections Adaptively. In *Proc. of MM*. 3908–3916.

[30] Shudong Li, Chuanyu Zhao, Qing Li, Jiuming Huang, Dawei Zhao, and Peican Zhu. 2023. BotFinder: a novel framework for social bots detection in online social networks based on graph embedding and community detection. *World Wide Web* 26, 4 (2023), 1793–1809.

[31] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proc. of MLSys*, Vol. 2. 429–450.

[32] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *Proc. of NIPS*. 5336–5346.

[33] Xuan Liang, Jianhua Tang, and Marie Siew. 2025. Decentralized Federated Learning Framework for Social IoT With Dynamic Network Topology. *IEEE Internet of Things Journal* 12, 17 (2025), 35001–35013.

[34] Qilong Lin and Jingya Zhou. 2025. BotBR: Social Bot Detection with Balanced Feature Fusion and Reliability-Enhanced Graph Learning. In *Proc. of SIGIR*. 392–402.

[35] Feng Liu, Chunfang Yang, Zhenyu Li, Daofu Gong, Rui Ma, and Fenlin Liu. 2023. Accou2vec: A Social Bot Detection Model Based on Community Walk. *IEEE Transactions on Dependable and Secure Computing* (2023), 1–17.

[36] 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).

[37] Yuhan Liu, Zhaoxuan Tan, Heng Wang, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. BotMoE: Twitter Bot Detection with Community-Aware Mixtures of Modal-Specific Experts. In *Proc. of SIGIR*. 485–495.

[38] Nuoyan Lyu, Bingbing Xu, Fangda Guo, and Huawei Shen. 2023. DCGNN: Dual-Channel Graph Neural Network for Social Bot Detection. In *Proc. of CIKM*. 4155–4159.

[39] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. of AISTATS*. 1273–1282.

[40] Shaojie Min, Shaobin Wang, Yaxiao Luo, Min Gao, Qingyuan Gong, Yu Xiao, and Yang Chen. 2025. FediLive: A Framework for Collecting and Preprocessing Snapshots of Decentralized Online Social Networks. In *Companion Proceedings of the ACM on Web Conference 2025*. 765–768.

[41] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. 2018. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization. *Proceedings of the IEEE* 106, 5 (2018), 953–976.

[42] Matthew N Nicholson, Brian C Keegan, and Casey Fiesler. 2023. Mastodon Rules: Characterizing Formal Rules on Popular Mastodon Instances. In *Proc. of CSCW Companion*. 86–90.

[43] Lisong Ou and Zhixin Li. 2025. Multi-modal Sarcasm Detection on Social Media via Multi-Granularity Information Fusion. *ACM Transactions on Multimedia Computing, Communications and Applications* 21, 3 (2025), 1–23.

[44] Luigi Palmieri, Lorenzo Valerio, Chiara Boldrini, and Andrea Passarella. 2023. The effect of network topologies on fully decentralized learning: a preliminary investigation. In *Proc. of NetAISys*. 1–6.

[45] Javier Pastor-Galindo, Félix Gómez Mármol, and Gregorio Martínez Pérez. 2022. Profiling users and bots in Twitter through social media analysis. *Information Sciences* 613 (2022), 161–183.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: an imperative style, high-performance deep learning library. In *Proc. of NIPS*. 8026–8037.

[47] Zhenwen Peng, Yingjie Song, Qiong Wang, Xiong Xiao, and Zhuo Tang. 2024. FedBN: A Communication-Efficient Federated Learning Strategy Based on Blockchain. In *Proc. of CSCWD*. 754–759.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models

From Natural Language Supervision. In *Proc. of ICML*. 8748–8763.

[49] Zhe Ren, Xinghua Li, Yinbin Miao, Zhuowen Li, Zihao Wang, Mengyao Zhu, Ximeng Liu, and Robert H Deng. 2023. Intelligent Adaptive Gossip-Based Broadcast Protocol for UAV-MEC Using Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Mobile Computing* 23, 6 (2023), 6563–6578.

[50] Yoel Roth and Samantha Lai. 2024. Securing Federated Platforms: Collective Risks and Responses. *Journal of Online Trust and Safety* 2, 2 (2024).

[51] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[52] Mohsen Sayyadiharikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2020. Detection of Novel Social Bots by Ensembles of Specialized Classifiers. In *Proc. of CIKM*. 2725–2732.

[53] Tao Sun, Dongsheng Li, and Bao Wang. 2022. Decentralized Federated Averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2022), 4289–4301.

[54] Min Tan, Yinfu Feng, Lingqiang Chu, Jingcheng Shi, Rong Xiao, Haihong Tang, and Jun Yu. 2024. FedSea: Federated Learning via Selective Feature Alignment for Non-IID Multimodal Data. *IEEE Transactions on Multimedia* 26 (2024), 5807–5822.

[55] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. 2022. GossipFL: A Decentralized Federated Learning Framework With Sparsified and Adaptive Communication. *IEEE Transactions on Parallel and Distributed Systems* 34, 3 (2022), 909–922.

[56] Hugo Touvron, Matthieu Cord, and Hervé Jégou. 2022. DeiT III: Revenge of the ViT. In *Proc. of ECCV*. 516–533.

[57] Lanjun Wang, Xinran Qiao, Yanwei Xie, Weizhi Nie, Yongdong Zhang, and Anan Liu. 2023. My Brother Helps Me: Node Injection Based Adversarial Attack on Social Bot Detection. In *Proc. of MM*. 6705–6714.

[58] Xianghua Xie, Chen Hu, Hanchi Ren, and Jingjing Deng. 2024. A survey on vulnerability of federated learning: A learning algorithm perspective. *Neurocomputing* 573 (2024), 127225.

[59] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and Generalizable Social Bot Detection through Data Selection. In *Proc. of AAAI*. 1096–1103.

[60] Yingguang Yang, Qi Wu, Buyun He, Hao Peng, Renyu Yang, Zhifeng Hao, and Yong Liao. 2024. SEBot: Structural Entropy Guided Multi-View Contrastive learning for Social Bot Detection. In *Proc. of KDD*. 3841–3852.

[61] Bodian Ye, Min Gao, Xiu-Xiu Zhan, Xinlei He, Zi-Ke Zhang, Qingyuan Gong, Xin Wang, and Yang Chen. 2025. EasyHypergraph: an open-source software for fast and memory-saving analysis and learning of higher-order networks. *Humanities and Social Sciences Communications* 12, 1291 (2025).

[62] Liangqi Yuan, Ziran Wang, Lichao Sun, Philip S Yu, and Christopher G Brinton. 2024. Decentralized Federated Learning: A Survey and Perspective. *IEEE Internet of Things Journal* 11, 21 (2024), 34617–34638.

[63] Xifan Zhang, Zhenyu Yan, and Guoliang Xing. 2024. FedMod: Towards Cross-modal Training for Heterogeneous Federated Learning Systems. In *Proc. of SenSys*. 828–829.

[64] Zhilin Zhang, Jun Zhao, Ge Wang, Samantha-Kaye Johnston, George Chalhoub, Tala Ross, Diyi Liu, Claudine Tinsman, Rui Zhao, Max Van Kleek, et al. 2024. Trouble in Paradise? Understanding Mastodon Admin's Motivations, Experiences, and Challenges Running Decentralised Social Media. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW2 (2024), 1–24.

[65] Haris Bin Zia, Aravindh Raman, Ignacio Castro, and Gareth Tyson. 2025. Collaborative Content Moderation in the Fediverse. *arXiv preprint arXiv:2501.05871* (2025).

[66] Matteo Zignani, Sabrina Gaito, and Gian Paolo Rossi. 2018. Follow the "Mastodon": Structure and Evolution of a Decentralized Online Social Network. In *Proc. of ICWSM*, Vol. 12. 541–550.

[67] Diana Zulli, Miao Liu, and Robert Gehl. 2020. Rethinking the "social" in "social media": Insights into topology, abstraction, and scale on the Mastodon social network. *New Media & Society* 22, 7 (2020), 1188–1205.

Appendix

A Convergence Analysis

We examined the convergence trend of FediScan and other methods. As depicted in Figure 5, our approach converges rapidly, requiring only about 2 rounds to reach an F1-score ≥ 0.58 and 10 rounds to achieve the final performance. These results indicate that FediScan achieves a favorable trade-off between detection performance and efficiency cost, making it suitable for deployment in decentralized online social networks.

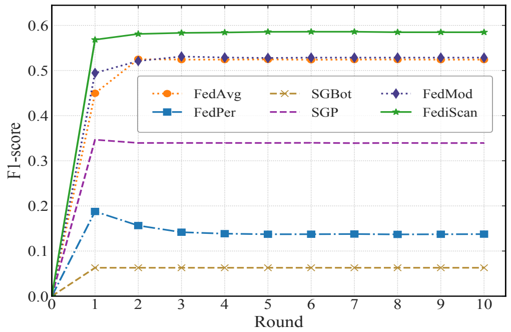


Figure 5: Rounds v.s. F1-score across different methods. FediScan converges fast due to its augmented multimodal representations and semantic-aware communication.

B Model Robustness Analysis

To evaluate the impact of backdoor injection on the performance of FediScan, we construct a set of malicious instances under varying multiple malicious ratio settings (10%, 20%, 30%, 40%, 50%). We employ backdoor injection only for malicious instances. During the training process, we randomly select users from each batch of malicious instances. These users are embedded with the trigger and modified to the target class. We assess model performance using two metrics, F1-score and attack success rate (ASR) [13, 58]. ASR is widely used to evaluate the effectiveness of an attack. As shown in Figure 6, increasing the backdoor ratio leads to a small decrease in F1-score and a corresponding slight rise in ASR, indicating that our model maintains considerable robustness under backdoor attacks.

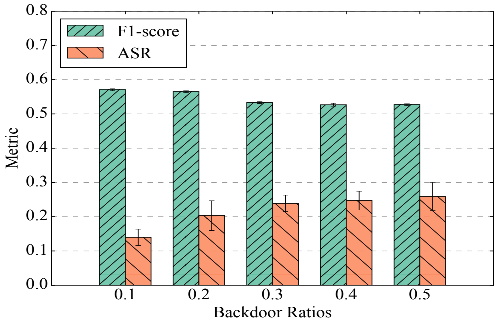


Figure 6: Model robustness across different backdoor ratios