BotSim: LLM-Powered Malicious Social Botnet Simulation

Boyu Qiao[1,2], **Kun Li**[1*], **Wei Zhou**[1†], **Shilong Li**[1,2], **Qianqian Lu**[1], **Songlin Hu**[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{qiaoboyu, likun2, zhouwei, lishilong, luqianqian, husonglin}@iie.ac.cn

Abstract

Social media platforms like X(Twitter) and Reddit are vital to global communication. However, advancements in Large Language Model (LLM) technology give rise to social media bots with unprecedented intelligence. These bots adeptly simulate human profiles, conversations, and interactions, disseminating large amounts of false information and posing significant challenges to platform regulation. To better understand and counter these threats, we innovatively design BotSim, a malicious social botnet simulation powered by LLM. BotSim mimics the information dissemination patterns of real-world social networks, creating a virtual environment composed of intelligent agent bots and real human users. In the temporal simulation constructed by BotSim, these advanced agent bots autonomously engage in social interactions such as posting and commenting, effectively modeling scenarios of information flow and user interaction. Building on the BotSim framework, we construct a highly human-like, LLM-driven bot dataset called BotSim-24 and benchmark multiple bot detection strategies against it. The experimental results indicate that detection methods effective on traditional bot datasets perform worse on BotSim-24, highlighting the urgent need for new detection strategies to address the cybersecurity threats posed by these advanced bots.

Code — https://github.com/QQQQQQBY/BotSim
Extended version — https://arxiv.org/abs/2412.13420

Introduction

In the modern digital era, online social networks (OSNs) such as X (formerly Twitter), and Reddit have become essential mediums for shaping human interaction due to their extensive connectivity and real-time information exchange. However, the prevalence of bots on these platforms poses a significant threat to OSN security (Cresci 2020; Ferrara 2023). For example, social bots have played notable roles in major events like presidential elections (Guglielmi 2020; Pacheco 2024) and global pandemics (Gallotti et al. 2020; Himelein-Wachowiak et al. 2021), where they disseminate misinformation and sway public opinion. Previous instances of social bots primarily stem from rule-based programs, however, recent advancements have integrated large language models (LLMs) that endow bots with more sophisticated, human-like capabilities (Yang and Menczer 2024). This development has further intensified the problem of information pollution on OSNs (Sun et al. 2024). Therefore, upgrading current detection systems and understanding the characteristics of LLM-driven bots has become a critical priority.

Previous research methods have predominantly been developed using traditional bot datasets. For instance, Yang *et al.* (2020) proposed a method that exploits differences in user profiles, while Cresci *et al.* (2016) suggested identifying the longest common subsequence of user actions. With advancements in deep learning, new methods have emerged focusing on text semantic content and user interaction networks. Wei *et al.* (2019) introduced the use of recurrent neural networks (RNNs) to encode posts and detect bots based on their semantic content. More recent methods, such as RGT (Feng et al. 2022), and BECE (Qiao et al. 2024) have employed graph neural networks (GNNs) and graph-enhanced strategies to improve detection performance. However, LLM-powered bots exhibit greater logical coherence and human-like qualities in profiles, text content, and interaction strategies, posing significant challenges to these existing detection methods (Feng et al. 2024; Ferrara 2023). Therefore, collecting datasets of LLM-driven bots is essential for developing new detection techniques (Yang and Menczer 2024). Traditional dataset collection methods, however, encounter the following two major challenges:

(1) Intelligent Challenges and Decline in Labeling Quality: The intelligence of LLM-driven bots has significantly advanced, making manual annotation tasks much more challenging and leading to a notable decline in annotation quality (Zhang et al. 2024). For instance, crowdsourcing tests conducted by Cresci *et al.* (2017) revealed that manual annotators had an accuracy rate of less than 24% when labeling social spam bots. Consequently, manual annotation has become unreliable, impairing the ability of detection models to differentiate between bots and genuine users.

(2) Ethical Constraints: For ethical reasons, large-scale deployment of social bots disguised as humans in real social networks to obtain genuine annotations for research is subject to strict restrictions. This situation makes research more

[*]Corresponding Author: Kun Li.
[†]Corresponding Author: Wei Zhou.

complex and challenging.

To address these challenges, we design a scalable malicious social botnet simulation framework called BotSim, upon which we construct an accurately labeled, LLM-driven bot dataset named BotSim-24. This dataset includes both real human accounts and LLM-driven agent bot accounts. To enhance the dataset's complexity, we implement a series of disguise techniques based on detection methods proposed in previous research focusing on bot profiles (Yang et al. 2020), textual content (Qiao et al. 2023), and interaction behavior patterns (Li et al. 2023). By leveraging LLMs to analyze and simulate characteristics of real users, we construct a comprehensively disguised and highly human-like LLM-driven bot dataset to expose and challenge the limitations and weaknesses of existing detection methods. We then benchmark multiple bot detection strategies on the BotSim-24 dataset. The experimental results validate the effectiveness of the dataset and underscore the significant threat that advanced bots pose to network security.

Our contributions can be summarized as follows:

- **BotSim Framework:** We are the first to propose a scalable LLM-driven malicious social botnet simulation framework, BotSim. This environment enables researchers to continuously track the latest bot evolution strategies and generate up-to-date datasets, thereby advancing the development of new detection methods.
- **LLM-Driven Bot Dataset:** Leveraging the BotSim simulation framework, we meticulously construct a bot detection dataset based on interaction scenarios from Reddit. This dataset incorporates real Reddit users and LLM-driven bot accounts, providing a comprehensive range of interaction data that enhances existing resources for social bot detection research.
- **Experimental Evaluation:** We conduct extensive experiments on the BotSim-24 dataset to evaluate the performance of various social bot detection models. The results show that detection methods effective on traditional bot datasets perform poorly on BotSim-24, highlighting the urgent need for new detection strategies to address the cybersecurity threats posed by these advanced bots.

BotSim: Botnet Simulation Framework

The overall framework of BotSim is shown in Figure 1, and it aims to model the activity characteristics and behavior patterns of LLM-driven malicious social bots in OSNs. BotSim consists of four components: the social environment, environmental perception, action list, and agent decision center.

Preliminaries

In this paper, we aim to use a botnet simulation framework to model the activity characteristics and behavior patterns of LLM-driven malicious bots on OSNs. The BotSim framework includes two types of users: human accounts from real social ecosystems, denoted as $U_H = \{U_{h_1}, U_{h_2}, ..., U_{h_n}\}$ and LLM-driven agent bot accounts, denoted as $U_B = \{U_{b_1}, U_{b_2}, ..., U_{b_m}\}$, where n and m represent the number of humans and bots, respectively. To simulate the continuous passage of time and the dynamic changes in interaction timing in real OSNs, we set up a timeline mechanism $T = \{t_1, t_2, ..., t_n\}$. In the timeline process, the set of interactions between users is represented as $D = \{U_B, U_H, E, T\}$ with $E = \{e_1, e_2, ..., e_n\}$ denoting the set of interaction relationships among users.

Social Environment

The social environment of BotSim is built from real social media ecosystem data and consists of account collection, message feeding, timeline setup, and interaction mode.

Account Collection The account collection includes real human accounts U_H and virtual Agent bot accounts U_B. Human accounts are sourced from data collected in real social environments, while the configuration and behavior of agent bot accounts are constructed by LLM-driven agents.

Message Feeding Message feeding utilizes a dual-filtering mechanism based on timelines and recommendation functions. Initially, the message flow is filtered through the timeline, and then it is optimally ranked by the recommendation function to produce the final message stream.

Timeline Setup The timeline setup $T = \{t_1, t_2, ..., t_n\}$ ensures the environment operates according to a predefined timeline logic. Additionally, each agent bot has its dedicated timeline, which is determined by the bot's activities and interactions with other accounts to meet the need for rapid simulation of long-time-span interactions.

Interaction Mode The interaction patterns $E = \{e_1, e_2, ..., e_n\}$ must adhere to the interaction settings defined by the specific social media platform. Interactions between accounts are accompanied by message flow outputs, such as likes and comments on current messages.

Environment Perception

The environment perception mechanism is important in the operation of BotSim, which helps the agent to capture the dynamic changes of the social environment and accurately transfer the perceived multi-dimensional information to the agent decision center so that the agent can make adaptive decisions based on the environmental information.

In BotSim, account profiles, message stream updates, and complex interaction data collectively form the core elements of the social environment. To enhance the agents' understanding and responsiveness to these complex environments, we have designed clear and structured prompts to assist the LLM in comprehending environmental information. Detailed prompts can be found in Appendix B.1.

Action List

The action list integrates commonly used information dissemination interactions on social media, including the following actions: (1) **Create User:** Create a new user profile. (2) **Post:** Generate and publish original content based on background knowledge and preferences. (3) **Comment:** Reply to selected posts or comments. (4) **Repost:** Share posts to achieve targeted information dissemination. (5) **Like**: Like posts to enhance positive feedback during interactions.

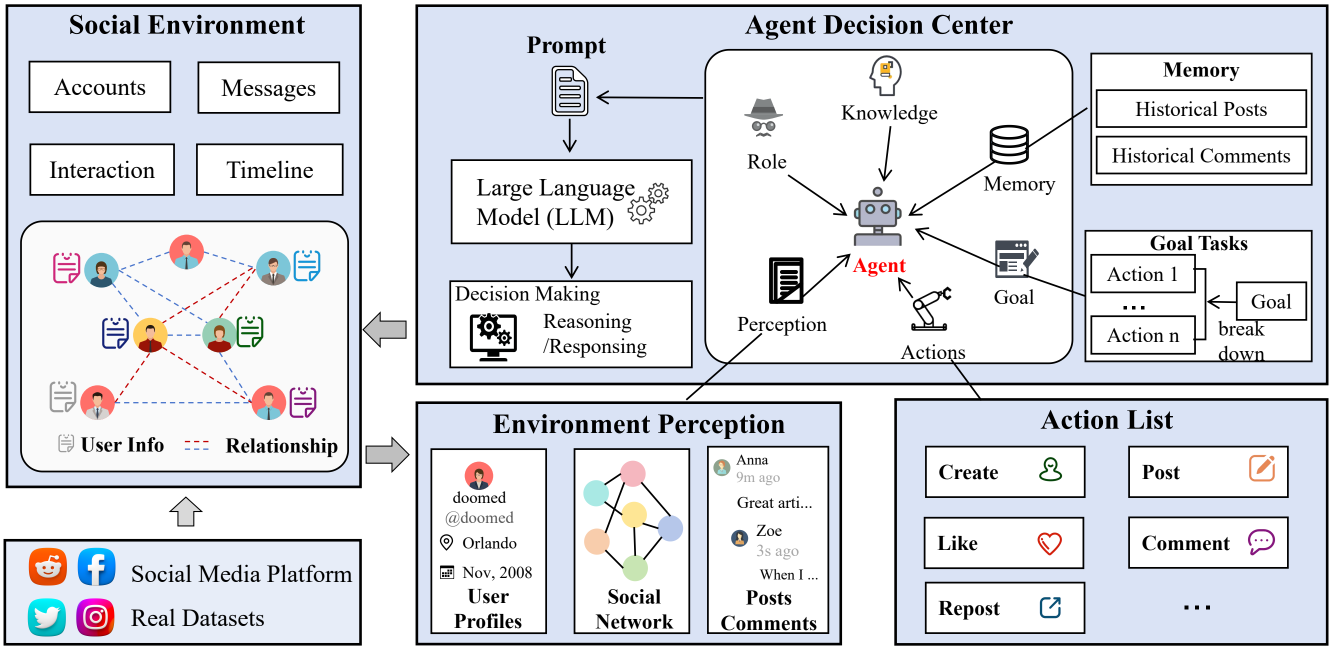


Figure 1: The overall framework of **BotSim**.

(6) **Browse:** Continue browsing the message stream based on the internal timeline if no preferred content is found. (7) **End**: Complete the mission and terminate the action.

BotSim provides a list of commonly used actions for information dissemination across various OSNs. Future research can select the appropriate actions based on specific needs and add new actions as required. Detailed description of the action list in Appendix B.2.

Agent Decision Center

The Agent Decision Center, as the core component of BotSim, integrates multidimensional information including goal tasks, role settings, background knowledge, environmental perception, action lists, and memory data. Its primary function is to accurately plan and execute action decisions, driving the comprehensive operation of BotSim.

Goal Tasks Goal tasks G define the specific needs for information dissemination and guide the agent's actions. The operators set these goals, and then the LLM decomposes the goal tasks into manageable and planned actions $PA = \{pa_1, pa_2, ..., pa_k\}$ to ensure the goals are achieved. Prompts for goal tasks are detailed in Appendix B.3.

Role Setting Role settings are crucial for the agent's decision-making process and include multidimensional attributes such as age, name, gender, preferences, education level, description, and geographic location. These attributes are applied to the profiles of created user accounts to help the agent establish a persona, enhancing both emotional expression and decision-making accuracy. More detailed information on role setup is provided in Appendix B.2.

Background Knowledge Given that LLMs may struggle to capture new social dynamics and knowledge, providing background knowledge KL can help LLMs generate relevant and novel content that aligns with goal tasks.

Memory Mechanism The memory mechanism filters relevant posts and comments related to the current task from the agent's historical records. This mechanism assists the agent in responding appropriately. An example of memory information is presented in Appendix B.4.

BotSim Execution Process

The overall execution process of the agent bots in BotSim involves the following steps: (1) **Specify the Platform:** Identify the social media platform to be simulated and gather the relevant data, including user profiles, messages, timestamps, and interaction data. (2) **Define Goal Tasks:** Clearly outline the goal tasks and compile the necessary background knowledge. (3) **Break Down Tasks:** Decompose the goal tasks into a series of executable actions, as detailed in Appendix B.3. (4) **Formulate Environment Prompts:** Perceive changes in the simulated social environment and create appropriate prompts, as detailed in Appendix B.1. (5) **Retrieve Memory Data:** Access historical posts and comments relevant to the goal task. (6) **Construct and Execute Prompts:** Build prompts using environmental perception information, memory data, planned action sequences, role settings, and background knowledge. Use these prompts to instruct the LLM, which will return the required action parameters. (7) **Update and Monitor:** Refresh the social environment and track the progress of the action sequence. If not completed, return to step (4). If completed, proceed to step (8). (8) **End:** Conclude the execution.

A complete prompt example is provided in Appendix B.4, and the algorithm for this execution process is further explained in Appendix B.5.

BotSim-24: LLM-driven Bot Detection Dataset

In this section, we present BotSim-24, a bot detection dataset powered by LLM. Building on the BotSim framework, we simulate information dissemination and user interactions across six SubReddits on Reddit. This process results in the creation of the BotSim-24 dataset, which includes 1,907 human accounts and 1,000 LLM-driven agent bot accounts.

Pre-Prepared Data

We first introduce the real OSN data information that must be pre-prepared for the BotSim simulation.

Reddit Social Environment Data Collection We choose six popular news-related SubReddits on Reddit to construct the social environment data for BotSim: “worldnews”, “politics”, “news”, “InternationalNews”, “UpliftingNews” and “GlobalTalk”. We collect posts, first- and second-level comments, timestamps, and user profiles from these six SubReddits between June 20, 2023, and June 19, 2024. We filter and annotate the collected accounts, resulting in 1,907 human Reddit accounts. More detailed data filtering and statistical information are presented in Appendix C.1.

Goal Tasks Our goal task is to create agent bots designed to spread disinformation within six news-oriented SubReddits. We focus on three highly debated international news events from 2023 to 2024: the “Russia-Ukraine war,” the “Israeli-Palestinian conflict,” and “U.S. politics.” Our objective is to disseminate disinformation related to these topics while concealing our activities by posting and engaging in discussions about a broad spectrum of international news on the SubReddits.

Background Knowledge Collection To build the knowledge base for the three major news events and various international news used for our goal tasks, we collect real news from four authoritative international news sources —“BBC”, “NBC News”, “NYTimes”, and “People’s Daily”, as well as fact-checking sites “Truthorfiction” and “Snopes”. The data spans from June 2023 to June 2024. This knowledge base helps the LLM generate content that is most relevant to the goal tasks. More detailed statistics are in Appendix C.2.

User Role Role settings in BotSim are used to construct the profiles of agent bots. Usernames and descriptions are generated by LLM simulation cases, while age, gender, education level, and geographic location are randomly assigned based on weighted statistics from Reddit. Additionally, since the goal tasks involve international news, political ideology settings are included in the role settings. This information is intended to assist the agent Bots in interactions, but the BotSim-24 dataset only provides profile information relevant to Reddit.

Bot Data Construction

Previous detection methods have primarily focused on identifying bot accounts that lack sufficient anthropomorphic features in areas such as profile metadata (Value or Boolean information) (Cresci et al. 2016; Moghaddam and Abbaspour 2022; Beskow and Carley 2018), textual content (Qiao et al. 2023; Liu et al. 2023), and interaction patterns (Feng et al. 2021b; Peng et al. 2022). Our goal is to create highly human-like bot accounts, driven by LLMs and based on the BotSim framework, to challenge these detection algorithms. To achieve this, the bots must effectively disguise themselves in these key areas to evade detection.

SubReddit	Posts	Users	1-Coms	2-Coms
worldnews	14,626	1,405	15,740	859
politics	2,4074	1,744	39,704	3,155
news	8,465	1,471	11,685	441
InternationalNews	3,906	554	5,477	311
UpliftingNews	1,219	266	1,148	35
GlobalTalk	342	342	472	16
Total	52,632	2,907	74,226	4,817

Table 1: Distribution of users, posts, and comments among six SubReddits. ‘1-Coms’ means first-level comments, ‘2-Coms’ means second-level comments. The total number of users is not the sum of users participating in different SubReddits, but the number of accounts participating in the social environment.

The disguise strategies we implement for bot accounts are as follows: (1) **Metadata Disguise:** We statistically analyze six types of value-type metadata from real Reddit users, including the number of posts, the number of first-level comments, the number of second-level comments, the ratio of posts to comments, posting frequency, and the number of active SubReddits. We then use LLM to integrate this statistical information to generate human-like metadata for bot accounts, effectively achieving metadata disguise. (2) **Textual Content Disguise:** The posts and comments of bot accounts are generated by LLMs based on contextual knowledge, user role information, browsing content, and other relevant factors. Unlike traditional bots, which often produce posts and comments with inconsistent contextual semantics, LLM-driven bots utilize advanced text understanding and generation capabilities to create contextually coherent and logically sound content, effectively disguising the textual output. (3) **Interaction Disguise:** On BotSim Reddit, interactions between accounts include first-level and second-level replies. The specific posts or comments that bot accounts reply to are autonomously determined by the LLM based on the goal task and browsed information. This method leverages the LLM’s analytical capabilities, distinguishing it from previous rule-based settings, and thereby achieving interaction disguise. We present a more detailed data statistical analysis and the process of constructing bot data in Appendix C.3 and C.4.

After setting up the data information and construction strategies required for BotSim, we selected GPT4o-mini as the LLM for generating the BotSim-24 dataset. The BotSim-24 contains users’ profiles, post and comment information, and relationship information. We present the statistical information of the constructed BotSim-24 dataset in Table 1.

Dataset	Users	Human	Bot	Training	Validation	Test	Edges	Edge Types	Communities
BotSim-24	2,907	1,907	1,000	2,304	582	291	46,518	3	6

Table 2: Statistics of our BotSim-24 dataset.

Dataset	Score	Metadata-based				Text-based	Meta-Text	Homo-GNN		Heter-GNN		
		AB	RF	DT	SVM	Wei *et al.*	Roberta+NN	GCN	GAT	BotRGCN	RGT	S-HGN
Manual Labeling Strategy												
Cresci-15	Acc	95.9±0.3	97.0±0.8	96.2±1.3	96.6±0.2	96.18±1.5	95.14±0.5	98.2±0.6	98.1±0.2	98.5±0.4	**98.6±0.3**	97.5±0.5
	F1	95.5±0.3	96.7±0.9	95.9±1.4	96.3±0.3	82.65±2.2	96.19±0.4	98.0±0.4	98.0±0.1	97.3±0.5	**98.5±0.2**	97.2±0.5
Cresci-17	Acc	91.2±0.2	89.1±0.2	86.2±0.2	84.1±0.3	89.30±0.3	**96.22±0.4**	/	/	/	/	/
	F1	83.4±0.2	80.9±0.2	76.4±0.2	72.8±0.3	78.40±0.2	**97.37±0.4**	/	/	/	/	/
Twibot-20	Acc	85.7±0.4	85.0±0.5	80.1±0.5	85.2±0.3	71.26±0.1	85.11±0.3	77.2±1.2	83.2±0.4	86.8±0.5	**86.9±0.3**	85.4±0.3
	F1	85.6±0.4	84.9±0.5	80.0±0.5	84.8±0.4	75.33±0.1	87.02±0.2	76.6±0.4	81.9±0.5	86.6±0.4	**86.7±0.4**	85.3±0.2
MGTAB-22	Acc	90.1±0.9	89.5±0.4	87.1±0.5	88.7±1.4	/	84.8±1.6	85.8±1.3	87.0±1.3	89.6±0.8	**92.1±0.4**	91.4±0.4
	F1	87.7±1.1	86.8±0.5	83.7±0.7	85.3±1.7	/	68.9±4.3	78.3±1.7	82.3±2.1	87.2±0.7	**90.4±0.5**	88.7±0.6
Weak Labeling Strategy												
Twibot-22	Acc	69.3±0.5	74.3±0.7	72.6±0.8	76.4±0.9	70.2±0.1	72.6±4.0	78.3±1.3	79.3±0.8	**79.6±0.4**	76.5±0.4	76.7±1.3
	F1	34.8±0.5	30.4±0.6	51.6±0.6	54.6±0.8	53.6±1.4	47.5±0.3	54.8±1.0	55.6±1.1	**57.6±1.4**	43.1±0.5	45.7±0.5
Simulation Labeling Strategy												
BotSim-24	Acc	77.5±3.2	75.7±2.2	71.4±2.1	74.4±2.2	50.8±2.9	67.6±4.0	72.7±2.2	80.3±1.4	**89.9±1.8**	82.3±2.2	87.7±1.3
	F1	74.8±3.5	72.4±1.6	68.5±2.2	69.8±2.4	50.4±1.4	30.5±6.1	50.5±5.5	73.1±3.6	**86.7±3.0**	76.4±3.1	83.1±2.9

Table 3: Performance of the baseline method on 6 datasets. Each baseline is performed five times with different seeds and we report the average performance and standard deviation. The best and second-best results are highlighted in bold and underlined. "/" indicates that the dataset does not contain support for the corresponding method. "Homo-GNN" indicates homogeneous GNNs and "Heter-GNN" indicates GNNs. We show the labeling strategy of different datasets.

Dataset Process

In this section, we describe the construction of user features and relationships in the BotSim-24 dataset.

User Features Construction Following the user feature processing methods used in the Cresci-15 (Cresci et al. 2015) and Twibot-20 (Feng et al. 2021a) datasets, we process the user features in BotSim-24 into metadata features and text features. Metadata features include Reddit user profile information, as detailed in Appendix C.1, and additional derived features based on basic profile information, totaling 10 standardized numerical data types. Text features include both posts and comments made by users. Compared to previous bot detection datasets that contain only user posting information, BotSim-24 also incorporates bi-level comment information.

User Relationships Construction Clarifying the types of interaction relationships between users is crucial for subsequent graph-based bot detection methods. We categorize user relationships into three types: first-level comment users and post users, second-level comment users and post users, and first-level comment users and second-level comment users. We record the number of comments exchanged between users, which can be used as edge weights in future graph structures to assist in detection. Appendix C.5 provides more detailed information on user features and user relationships, as well as comparisons with other datasets.

Experiment

Experiment Settings

Parameter Settings Our experiments are conducted on four Tesla V100 GPUs with 32GB of memory. Detailed hyperparameter settings can be found in Appendix A.1.

Baseline We evaluate various commonly used methods for bot detection on BotSim-24, including feature-based, text-based, and graph-based approaches. These methods encompass the the Adaboost classifier (AB) (Hastie et al. 2009), decision tree (DT) (Lepping 2018), random forest (RF) (Yang et al. 2020), support vector machine (SVM) (Boser, Guyon, and Vapnik 1992), the approach proposed by Wei **et al.** (2019), Roberta+NN, and both homogeneous graph methods (GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017)) and heterogeneous graph approaches (S-HGN (Lv et al. 2021), BotRGCN (Feng et al. 2021b), RGT (Feng et al. 2022)). A more detailed description is provided in Appendix A.2.

Datasets We evaluate BotSim-24 alongside five publicly available bot detection datasets: Cresci-15 (Cresci et al. 2015), Cresci-17 (Cresci et al. 2017), TwiBot-20 (Feng et al. 2021a), TwiBot-22 (Feng et al. 2022), and MGTAB-22 (Shi et al. 2023). Consistent with the division used in TwiBot-20 and MGTAB-22, we randomly divide all datasets into training, validation, and test sets with a ratio of 7:2:1. Table 2 shows the division of the BotSim-24 dataset. More detailed comparisons are presented in Appendix A.3.

Experiment Results

We evaluate the performance of 11 baseline methods across 6 datasets, with each baseline executed 5 times. The average performance and standard deviation are reported. The labeling strategy, detection accuracy, and F1-scores for each dataset are presented in Table 3. Our key findings are summarized as follows:

The BotSim-24 dataset presents greater challenges for baseline detection methods. Table 3 indicates that the 11 baseline methods perform poorly on both the Twibot-22 and BotSim-24 datasets. The weak results on Twibot-22 are likely due to its reliance on low-quality weak supervision for labeling. Despite the high label reliability of BotSim-24, it still underperforms compared to other reliable datasets like Cresci-15 and Twibot-20, due to its effective camouflage of various features. Specifically, for metadata-based methods, the BotSim-24 dataset undermines the performance of traditional machine-learning approaches due to its successful camouflage of metadata features. For text-based methods, the exceptional text comprehension and generation capabilities of LLM make it nearly impossible to distinguish between bots and humans in such highly human-like content. Consequently, Wei *et al.*'s method performs almost like random guessing. Furthermore, the "Roberta + NN" neural network method, which combines text and metadata features, not only fails to improve detection performance but also shows negative gains, highlighting the effectiveness of bots' profiles and text disguises.

Graph-based Methods Perform Better in Detecting LLM-driven Bots. The detection performance after fusing relational edges is superior to that of methods only based on metadata and textual content. Although the rapid development of LLM technology has greatly enhanced the anthropomorphic nature of bot accounts, the complexity and dynamics of human-established relationships are still difficult to be fully modeled by LLM. This finding emphasizes the indispensability of inter-user relationship information in bot detection, and we believe it is a key clue for future research to distinguish human and machine behaviors.

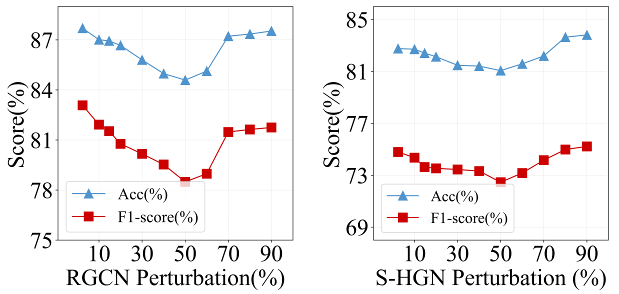
Methods Based on Heterogeneous Graphs Outperform Homogeneous Graphs. The superior performance of heterogeneous graphs is primarily due to their ability to effectively utilize different types of edge relationships within the graph. This capability reveals diverse interaction patterns between users, allowing GNNs to gather more comprehensive information and enhance detection capabilities. Additionally, we observe that RGCN and S-HGN exhibit excellent performance on the BotSim-24 dataset. This is not only due to the excellence of RGCN and S-HGN design but also affected by the dataset. In the subsequent experimental analysis, we further elucidate the key factors contributing to their superior detection performance.

Experimental Analysis

The Reason For the Excellent Performance of GNN Method. In BotSim, human data is generated in timeline order, meaning that real human users do not participate in interactions. Consequently, the BotSim-24 dataset includes interactions between humans (human ↔ human), bots and humans (bot → human), and bots (bot ↔ bot), but it does not include interactions initiated by humans towards bots (human → bot). This creates an incomplete graph structure, missing directed edges from human to bot nodes, as detailed in Appendix A.4. This structural incompleteness enables GNNs to identify the intrinsic differences between human and bot accounts, which contributes to their superior detection performance.

Dataset	Cresci-15	Twibot-20	MGTAB-22
Human → Bot	0	313	1140
Edge Num	13,130	2,133	47,907
Proportion	**0%**	**14.7%**	**2.4%**

Table 4: The interaction ratios of humans → bots. We randomly select the same number of humans and bots from these three datasets as in BotSim-24. We count the human-bot interaction edges "Human → Bot" and and the total number of edges "Edge Num".



(a) RGCN Edge Perturbation (b) S-HGN Edge Perturbation

Figure 2: The impact of different proportions of edge perturbations on RGCN and S-HGN detection performance.

Edge Perturbation Experiment. To further validate our observations and hypotheses, we design an edge perturbation experiment. This experiment randomly reverses the direction of a proportion of the original edges to simulate varying levels of interaction between humans and bots. We count the interaction ratios between humans and bots in three real-world datasets in Table 4: 0%, 14.7%, and 2.4% for Cresci-15, TwiBot-20, and MGTAB-22, respectively. The differences in interaction ratios may be due to varying proportions of bots in different events. We then visualize edge perturbations using RGCN and S-HGN at these ratios and additional ones in Figure 2. Results indicate that detection performance initially declines and then improves with increasing perturbation ratios. When the perturbation ratio reaches 50%, performance drops but then rebounds. This is because introducing more directed edges from humans to bots allows the GNN to effectively capture these structural differences, enhancing detection performance.

Discussion. LLM-driven bots are becoming increasingly difficult to detect. The BotSim-24 dataset does not include interactions between humans and bots. Statistics in Table 4

LLM	Acc(%)	F1-score(%)
LLAMA 3-8B		
Zero-Shot Text	54.82	54.22
2-Shot Text	55.11	54.40
5-Shot Text	57.79	52.89
ChatGLM 3-6B		
Zero-Shot Text	42.61	9.72
2-Shot Text	42.62	43.94
5-Shot Text	47.78	54.22
GPT-3.5-turbo-ca		
Zero-Shot Text	42.96	50.89
2-Shot Text	48.80	53.29
5-Shot Text	53.61	57.94
GPT-4-turbo-ca		
Zero-Shot Text	39.18	22.03
2-Shot Text	59.79	32.03
5-Shot Text	70.76	63.58

Table 5: LLM-based bot detectors on the text content.

show that such interactions are also relatively sparse in actual OSNs. **However, as LLM-powered bots become more prevalent, their high human-like characteristics will inevitably lead to an increase in human-bot (human $\leftrightarrow$ bot) interactions. As demonstrated by our edge perturbation experiments, this trend will challenge and undermine the effectiveness of GNN-based methods.** Furthermore, Table 5 offers a detailed overview of the performance of various LLMs in account detection tasks based on textual content. Additionally, Figure 4 in Appendix A.5 visually illustrates findings on the accuracy of human annotators. These results highlight the difficulty LLMs face distinguishing between text they generate and text authored by humans. Human annotators also struggle to achieve high accuracy in this regard. For additional details, please refer to Appendix A.5. This underscores the critical challenge of detecting LLM-driven bots and emphasizes the urgent need for innovative detection strategies to keep pace with their evolving capabilities.

Related Work

Social Simulation Based on LLM. Agent-based simulation modeling plays a crucial role in social public opinion research, the most common application is to use LLM to simulate human behavior. Leveraging LLMs' human-like capabilities in perception, reasoning, and behavior, agents with unique characteristics can engage in extensive interactions, simulate real-world social phenomena, and generate rich behavioral data for in-depth social science analysis. For example, Park *et al.* (2022) proposed a simulation platform to explore social interactions beyond individual intentions. S^3 (Gao et al. 2023) utilized Markov chains and LLMs to simulate public opinion dynamics. Sotopia (Zhou et al. 2023) designed a framework for assessing social intelligence. Additionally, Mou *et al.* (2024) developed a Twitter user simulation framework to replicate the dynamic responses of user groups following trigger events. In contrast to these studies, our proposed simulation framework does not use LLMs to model human behavior. Instead, it focuses on simulating the behavior of program-driven bots within social networks, aiming to investigate the threats posed by LLM-driven bots to social media regulation platforms.

Bot Detection Dataset. Numerous bot datasets have been introduced over the years, with the Bot Repository compiling datasets from 2011 to 2022. The earliest, Caverlee-2011 (Lee, Eoff, and Caverlee 2011), was collected using honeypot techniques. Since 2015, bot detection datasets have surged, including those focused solely on user profile information, such as Gilani-2017 (Gilani et al. 2017) and PronBots-2019 (Yang et al. 2019). Additionally, there are datasets like Cresci-17, which include both profile and text information, and more comprehensive datasets like Twibot-20 and Twibot-22, which encompass profile, text, and interaction data. The advancement of LLMs has further driven the creation of bot datasets based on LLMs. Yang *et al.* (2024) constructed a dataset from bots' inadvertently self-revealing tweets, comprising 1,140 bots and 1,140 human accounts; however, this dataset contains only textual information. Li *et al.* (2023) collected data from the Chirper, an LLM-driven bot network. Since this platform lacks real human participants, the dataset consists solely of bot information, limiting its utility for detection research. In contrast, our LLM-driven bot-human interaction dataset, built on the simulation framework, includes profiles, text, and rich interaction data, supporting the development of new methods for detecting LLM-driven bots.

Conclusion

In this paper, we first introduce BotSim, a scalable framework for simulating malicious social botnets. We use BotSim to simulate interaction patterns on the Reddit social platform, creating an LLM-driven highly anthropomorphic bot detection dataset BotSim-24. Subsequently, we validate the performance of both feature-based and GNN-based detection methods on BotSim-24. The experimental results strongly affirm the contribution of the BotSim-24 dataset to advancing research in social bot detection.

Limitation

Our research has two main limitations: First, due to the cost constraints of using LLMs, we have not yet developed a large-scale bot detection dataset. Second, since the human data is pre-collected from real social networks, the simulation environment lacks actual interactions between humans and bots. To address this, we simulate human-bot interaction ratios in real datasets through edge perturbation experiments and make the perturbed edge information publicly available to support future research. Additionally, we propose two feasible approaches: (1) Engaging domain experts to further simulate real users to supplement the missing human-bot interactions in BotSim-24. (2) Crowdsourcing a large number of real human accounts, creating bot accounts, and facilitating their interactions in a virtual simulation environment to develop more comprehensive research datasets.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2022YFC3302102).

References

Beskow, D. M.; and Carley, K. M. 2018. Bot conversations are different: leveraging network metrics for bot detection in twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 825–832. IEEE.

Boser, B. E.; Guyon, I. M.; and Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152.

Cresci, S. 2020. A decade of social bot detection. *Communications of the ACM*, 63(10): 72–83.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2015. Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*, 80: 56–71.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2016. DNA-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems*, 31(5): 58–64.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion*, 963–972.

Feng, S.; Tan, Z.; Wan, H.; Wang, N.; Chen, Z.; Zhang, B.; Zheng, Q.; Zhang, W.; Lei, Z.; Yang, S.; et al. 2022. Twibot-22: Towards graph-based twitter bot detection. *Advances in Neural Information Processing Systems*, 35: 35254–35269.

Feng, S.; Wan, H.; Wang, N.; Li, J.; and Luo, M. 2021a. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 4485–4494.

Feng, S.; Wan, H.; Wang, N.; and Luo, M. 2021b. BotRGCN: Twitter bot detection with relational graph convolutional networks. In *Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining*, 236–239.

Feng, S.; Wan, H.; Wang, N.; Tan, Z.; Luo, M.; and Tsvetkov, Y. 2024. What Does the Bot Say? Opportunities and Risks of Large Language Models in Social Media Bot Detection. *arXiv preprint arXiv:2402.00371*.

Ferrara, E. 2023. Social bot detection in the age of ChatGPT: Challenges and opportunities. *First Monday*.

Gallotti, R.; Valle, F.; Castaldo, N.; Sacco, P.; and De Domenico, M. 2020. Assessing the risks of 'infodemics' in response to COVID-19 epidemics. *Nature human behaviour*, 4(12): 1285–1293.

Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J.; Wang, H.; Jin, D.; and Li, Y. 2023. S $\hat{3}$: Social-network Simulation System with Large Language Model-Empowered Agents. *arXiv preprint arXiv:2307.14984*.

Gilani, Z.; Farahbakhsh, R.; Tyson, G.; Wang, L.; and Crowcroft, J. 2017. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 349–354.

Guglielmi, G. 2020. The next-generation bots interfering with the US election. *Nature*, 587(7832): 21–21.

Hastie, T.; Rosset, S.; Zhu, J.; and Zou, H. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3): 349–360.

Himelein-Wachowiak, M.; Giorgi, S.; Devoto, A.; Rahman, M.; Ungar, L.; Schwartz, H. A.; Epstein, D. H.; Leggio, L.; and Curtis, B. 2021. Bots and misinformation spread on social media: Implications for COVID-19. *Journal of medical Internet research*, 23(5): e26933.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lee, K.; Eoff, B.; and Caverlee, J. 2011. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the international AAAI conference on web and social media*, volume 5, 185–192.

Lepping, J. 2018. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Li, S.; Qiao, B.; Li, K.; Lu, Q.; Lin, M.; and Zhou, W. 2023. Multi-modal social bot detection: Learning homophilic and heterophilic connections adaptively. In *Proceedings of the 31st ACM International Conference on Multimedia*, 3908–3916.

Li, S.; Yang, J.; and Zhao, K. 2023. Are you in a masquerade? exploring the behavior and impact of large language model driven social bots in online social networks. *arXiv preprint arXiv:2307.10337*.

Liu, Y.; Tan, Z.; Wang, H.; Feng, S.; Zheng, Q.; and Luo, M. 2023. Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 485–495.

Lv, Q.; Ding, M.; Liu, Q.; Chen, Y.; Feng, W.; He, S.; Zhou, C.; Jiang, J.; Dong, Y.; and Tang, J. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1150–1160.

Moghaddam, S. H.; and Abbaspour, M. 2022. Friendship preference: Scalable and robust category of features for social bot detection. *IEEE Transactions on Dependable and Secure Computing*, 20(2): 1516–1528.

Mou, X.; Wei, Z.; and Huang, X. 2024. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. *arXiv preprint arXiv:2402.16333*.

Pacheco, D. 2024. Bots, Elections, and Controversies: Twitter Insights from Brazil's Polarised Elections. In *Proceedings of the ACM on Web Conference 2024*, 2651–2659.

Park, J. S.; Popowski, L.; Cai, C.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–18.

Peng, H.; Zhang, Y.; Sun, H.; Bai, X.; Li, Y.; and Wang, S. 2022. Domain-aware federated social bot detection with multi-relational graph neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Qiao, B.; Li, K.; Zhou, W.; Yan, Z.; Li, S.; and Hu, S. 2023. Social bot detection based on window strategy. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 2201–2206. IEEE.

Qiao, B.; Zhou, W.; Li, K.; Li, S.; and Hu, S. 2024. Dispelling the Fake: Social Bot Detection Based on Edge Confidence Evaluation. *IEEE Transactions on Neural Networks and Learning Systems*.

Shi, S.; Qiao, K.; Chen, J.; Yang, S.; Yang, J.; Song, B.; Wang, L.; and Yan, B. 2023. Mgtab: A multi-relational graph-based twitter account detection benchmark. *arXiv preprint arXiv:2301.01123*.

Sun, Y.; He, J.; Cui, L.; Lei, S.; and Lu, C.-T. 2024. Exploring the Deceptive Power of LLM-Generated Fake News: A Study of Real-World Detection Challenges. *arXiv preprint arXiv:2403.18249*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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

Yang, K.-C.; and Menczer, F. 2024. Anatomy of an AI-powered malicious social botnet. *Journal of Quantitative Description: Digital Media*, 4.

Yang, K.-C.; Varol, O.; Davis, C. A.; Ferrara, E.; Flammini, A.; and Menczer, F. 2019. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies*, 1(1): 48–61.

Yang, K.-C.; Varol, O.; Hui, P.-M.; and Menczer, F. 2020. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 1096–1103.

Zhang, Y.; Sharma, K.; Du, L.; and Liu, Y. 2024. Toward Mitigating Misinformation and Social Media Manipulation in LLM Era. In *Companion Proceedings of the ACM on Web Conference 2024*, 1302–1305.

Zhou, X.; Zhu, H.; Mathur, L.; Zhang, R.; Yu, H.; Qi, Z.; Morency, L.-P.; Bisk, Y.; Fried, D.; Neubig, G.; et al. 2023. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*.