Temporal-aware Social Bot Detection with Graph Contrastive Learning

Weiguang Wang[1,2], Tianning Zang[1,2], and Xiaoyu Zhang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{wangweiguang,zangtianning,zhangxiaoyu}@iie.ac.cn

Abstract. In recent years, AI-powered social bots have become more anthropomorphic and deceptive, posing a serious challenge to combat the spread of misinformation in social networks. However, the lack of high-quality annotated data severely limits the further development of social bot detection technology. Moreover, existing graph-based approaches analyze social networks as static graphs, overlooking the inherent dynamic nature of the evolving social network. To address the above drawbacks, we propose BotTGCL, a novel social bot detection framework that jointly utilizes the generalization patterns of the social graph structure and the dynamic nature of evolving social networks to improve the detection of social bot. Specifically, we construct the social network as a dynamic graph and employ a graph contrastive learning module to learn the topological patterns of graph structure in unlabeled social networks. We then propose a graph temporal module to integrate historical context and extract temporal patterns from the evolving graph. Finally, we fuse topological patterns and temporal patterns to classify users as social bots or humans. Extensive experiments conducted on two comprehensive social bot detection benchmarks demonstrate that BotTGCL achieves superior performance compared to state-of-the-art methods and exhibits exceptional performance in real-world scenarios with scarce labeled data. Additional studies also confirm the effectiveness of our proposed graph structure contrastive learning and graph temporal pattern learning.

Keywords: Social Bot Detection, Graph Contrastive Learning, Temporal Patterns Learning.

1 Introduction

Social bots powered by artificial intelligence and large language models (LLMs) are widely applied on various social network platforms. These advanced social bots can realistically mimic human social behavior and language habits, continuously evolve to evade detection, and are used by malicious operators to spread disinformation, manipulate public sentiment and political interference. Researchers have found that bots are used to participate in the Russia-Ukraine information war [1] and interfere with national elections [2]. Moreover, with the emergence of ChatGPT, social bots are becoming more anthropomorphic and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1

deceptive, the detection of social bots has become an urgent problem to be solved.

Earlier machine learning-based social bot detection methods generally utilize feature engineering to extract features from user profiles and employ traditional machine learning algorithms to identify social bots [4, 5]. However, feature engineering heavily relies on analytical experience, and continuously updating bots can manipulate metadata to evade detection strategies. Deep learning-based methods use the profiles and contents of the accounts post as input to the neural network, and identify social bots by building a series of convolutional and recurrent neural networks [6, 7]. These methods only consider the user profile and textual information, without utilizing the relations in social networks, making it difficult to achieve effective performance in detecting the constantly evolving social bots. Due to the superiority of graph neural networks (GNNs) in processing non-Euclidean space data, especially in social networks, more and more graph-based methods have been proposed for detecting social bots, which leverage graph neural networks to capture the structure information of the social network graph [9–11]. The above graph-based methods have achieved high recognition accuracy in the social bot detection task. Social bots powered by large language models continue to appear on social network platforms, but there is a serious lack of high-quality labeled data related to them. This makes traditional supervised models unmet training needs and vulnerable to attacks by new social bots, hindering the development of bot detection. In addition, the above models describe social networks as static graphs and only analyze their recent structural information, failing to fully utilize the rich historical context of evolving social networks and the subtle differences in dynamic behavioral characteristics of social bots and human users.

To address the challenges above, we propose a novel social bot detection framework BotTGCL (Bot Detection with Temporal-Aware and Graph Contrastive Learning). Specifically, we first propose a graph contrastive learning module to perform self-supervised learning on unlabeled social networks to capture generalization patterns in the social graph. We then describe a social network as a dynamic graph and construct snapshots of the social graph at a certain time interval. After that, we apply the graph temporal patterns learning module to integrate historical context, extract changes in behavior patterns over time, and obtain the graph temporal patterns. Finally, we fuse the graph structural patterns and the temporal patterns to distinguish social bots from genuine users. Our main contributions are summarized as follows.

- We propose to comprehensively leverage the generalization patterns of the topological structure in the social graph and the dynamic nature of temporal patterns in a constantly evolving social network to detect social bots.
- On this basis, we propose BotTGCL: a novel social bot detection framework supporting few-shot learning. We characterize a social network as a dynamic graph, obtain the underlying structural knowledge from the unlabeled social graph, and extract the valuable temporal patterns from dynamic graph.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1

- We have conducted sufficient experiments to evaluate our proposed BotTGCL. Experimental results demonstrate that our proposed method is more efficient and generalized than the state-of-the-art baseline methods and exhibits superior performance in real-world scenarios with scarce labeled data.

2 Related Work

2.1 Social Bot Detection

Social bots are automated software-run accounts that pose a serious threat to the authenticity and integrity of online platforms. Existing Social bots detection methods fall mainly into three categories: feature-based methods, text-based methods, and graph-based methods.

Feature-based methods generally focused on manually designed features and combined them with traditional machine learning classifiers. These methods conduct feature engineering based on handcrafted user features extracted from user profiles. Yang et al. [5] utilize minimal account metadata and labeled datasets to detect social bots. Davis et al. [7] leverage more than 1,000 user features to evaluate the extent to which a Twitter account exhibits similarity to the known characteristics of social bots. However, evolving bots can evade the detection of feature-based approaches by creating deceptive accounts with manipulated metadata.

Text-based methods adopt NLP techniques to detect Twitter bots with their historical tweets and user descriptions. Wei et al. [8] use word embeddings to encode user historical tweets and adopt a Bi-directional LSTM to distinguish Twitter bots from human accounts. David et al.[13] present a BERT (Bidirectional Encoder Representation from Transformers) based bot detection model to analyze tweets written by bots and humans. However, text-based methods are easily deceived when advanced bots post stolen tweets and descriptions from genuine users.

Graph-based methods utilize the graph structure to represent the various users and diversified relationships in social networks, and attempt to separate bots and humans based on the graph structure. Feng et al. [10] propose BotRGCN framework and use R-GNNs for social bot detection. Lei et al. [12] adopt a Twitter Bot detection framework BIC and employs a text-graph interaction module to enable information exchange across modalities in the learning process. Feng et al. [11] construct heterogeneous information networks and propose relational graph transformers to model influence intensity with the attention mechanism and learn node representations to detect social bot. However, the above methods usually analyze social networks as static graphs, fail to account for the dynamic nature of social networks, and their detection performance degrades significantly in real-world where high-quality labeled data is lacking.

2.2 Graph Contrastive Learning

Graph Contrastive Learning (GCL) aims to learn an encoding model that produces similar representations for similar nodes in a graph and significantly differ-

DOI: 10.1007/978-3-031-97564-6_1

ent representations for dissimilar nodes. The definition of positive and negative examples is crucial to graph contrastive learning methods. Thereby, based on the type of contrast examples, existing methods can be divided into two categories: instance contrast and cross-level contrast.

For the first category, instance contrast refers to the comparison between different augmented views of the same sample. Qiu et al. [14] propose the GCC framework based on self-supervised GNNs, which captures general network topological properties across multiple networks. Zhu et al. [15] propose GRACE framework which leverages a contrastive objective at the node level for unsupervised graph representation learning and proposes a hybrid scheme for generating graph views on both structure and attribute levels. The remarkable experimental results of these models show the great potential of unsupervised graph contrastive learning in node classification applications.

For the second category, these works propose to obtain node representations by leveraging local and global information of a graph [16]. Jiao et al. [17] leverage the strong correlation between anchor nodes and their neighbor subgraphs for scalable graph learning representation to obtain contextual structure information. Mavromatis et al. [18] propose a GIC framework which leverages the coarse grain information that is available in most graphs to identifies nodes that belong to the same clusters and maximizes their mutual information.

Graph contrastive learning can extract valuable knowledge without relying on annotated data, thereby effectively address the issue of scarce labeled data in real-world scenarios. Motivated by the potential of GCL, our work is to maximize the benefits of GCL and address the issue of labeled data scarcity in social bot detection tasks.

2.3 Dynamic Graph Neural Network

Social networks are dynamic networks that continuously evolve over time, exhibiting certain intrinsic dynamic characteristics. Dynamic graph neural networks are designed to capture this dynamic nature and are widely adopted in various tasks, especially in node classification task. Kim et al. [19] propose DyGRAIN framework that incrementally learns dynamic graph representations by reflecting the influential change in receptive fields of existing nodes and maintaining previous knowledge of informative nodes prone to be forgotten. Pareja et al. [20] adapt the graph convolutional network model along the temporal dimension and capture the dynamism of the graph sequence through using an RNN to evolve the GCN parameters.

Inspired by the excellent performance of the above work in downstream tasks in the graph field, we propose a novel approach that utilizes a self-attention mechanism to learn the intrinsic dynamic patterns of social networks, thereby effectively enhancing the detection and identification of social bots.

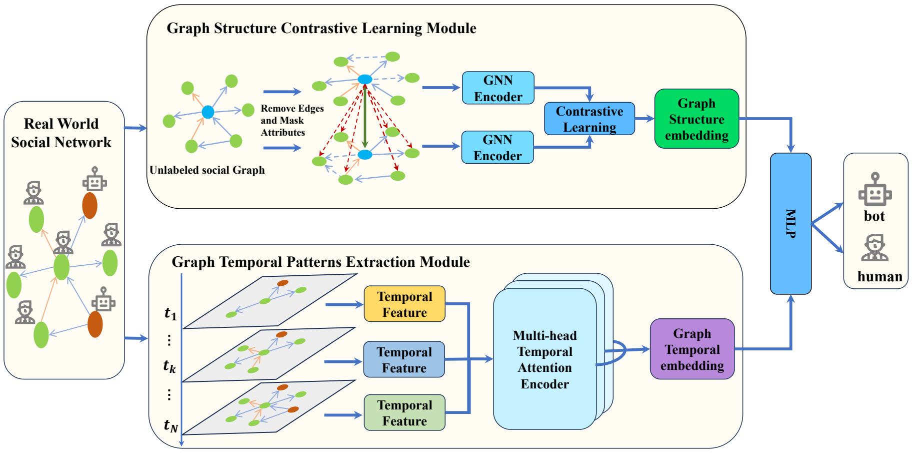
Graph Structure Contrastive Learning Module
Real World Social Network
Unlabeled social Graph
Remove Edges and Mask Attributes
GNN Encoder
GNN Encoder
Contrastive Learning
Graph Structure embedding
MLP
bot
human
Graph Temporal Patterns Extraction Module
t_1
t_k
t_N
Temporal Feature
Temporal Feature
Temporal Feature
Multi-head Temporal Attention Encoder
Graph Temporal embedding

Fig. 1. Overview of our temporal-aware with graph contrastive learning social bot detection framework BotTGCL

3 Methodology

3.1 Overview

Figure 1 presents an overview of our proposed temporal-aware with graph contrastive learning social bot detection framework BotTGCL. Our framework comprises two modules to primarily capture the deep semantic knowledge in the topological structure and temporal patterns of social networks. Specifically, we first utilize a graph contrastive learning module to perform self-supervised learning on unlabeled social network graphs, in order to obtain structural embedding that can capture the generalization patterns in social graph. We then characterize a social network as a dynamic graph and construct snapshots of the social network graph at a certain time interval. After that, we adopt graph temporal patterns learning module to integrate historical context, and extract changes in behavior patterns over time to obtain the graph temporal patterns embedding. Finally, we adopt MLP to integrate the graph structural embedding and the temporal embedding to distinguish social bots from genuine users.

3.2 Dynamic Graph Construction

We first construct a dynamic social network graph by slicing the social network at a specific time interval $\varDelta_t$, and obtain the network snapshot G^{t_k} at timestamp t_k. Similarly to the approach in the state-of-art approach BotRGCN [10], we then obtain the node representation using the information encoding procedure which use a concatenate function to fuse the numerical, textual, and categorical features of the nodes to form the initial encoding. After that, we transform the initialization value of the node encoding with a fully connected layer to serve as initial feature vectors $x_{i(init)}^{t_k}$ at timestamp t_k as follows:

$$x_{i(init)}^{t_k} = \sigma(W_0 \bullet x_{i(0)}^{t_k} \ + \ b_0) \tag{1}$$

where W_0 and b_0 are learnable parameters, σ denotes a activation function, and $x_{i(0)}^{t_k}$ denotes the initial encoding of node i in snapshot G^{t_k} at timestamp t_k.

3.3 Graph structure contrastive learning

We adopt Graph Contrastive Learning (GCL) to perform self-supervised learning on unlabeled social network data, and obtain the underlying structural information from social graph. Our GCL framework follows the common GCL paradigm where the model seeks to maximize the mutual information of representations between different views derived from the same input graph.

Graph augmentation Inspired by GCA [21], we proposed a graph augmentation model that performs edge deletion and node masking according to the importance of the edge and node attributes in the social graph. Our graph augmentation method could preserve the fundamental topological and semantic graph patterns in social network graph.

On the Topology level, we define edge centrality $\mathcal{P}_{uv}$ for *edge* (u, v) to measure its influence based on centrality of start node on directed social graph, because the starting node represents the active initiator of the relationship in social network. We apply in-degree to calculate the node centrality in social network graph. After that, we calculate the probability of each edge based on its centrality value, the probabilities can be obtained as:

$$\mathcal{P}_{uv} = \min\left(\frac{S_{max} - S_{uv}}{S_{max} - \mu_{es}} \bullet p_e, p_\tau\right) \tag{2}$$

where p_e is a hyper-parameter whose value represents the overall probability of removing edges, S_{max} and μ_{es} denotes the maximum and average of S_{uv}, and p_τ is a cut-off probability used to avoid extremely high removal probabilities that could overly damage the fundamental topology of the social network graph.

On the node attribute level, we add noise to node attributes via randomly masking a fraction of dimensions with zeros in node features. Formally, we sample a random vector $\widetilde{m} = \{0,\ 1\}^F$ where each dimension of it is independently drawn from a Bernoulli distribution. And then we obtain the generated node features as $\widetilde{x}$. Similarly to the topology-level augmentation, we obtain the probability pf_i to reflect the importance of the $i-th$ dimension of node features. Given that feature dimensions frequently appearing in influential nodes should be important, we define the weights of feature dimensions as follows:

$$W_i = \sum_{u \in V} |d_{ui}| \bullet \varphi_c(u) \tag{3}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1

where the first term $|d_{ui}| \in \{0, 1\}$ indicates the occurrence of dimension i in node u, and the second term $\varphi_c(u)$ measures the node importance of each occurrence. Similarly to the topology augmentation, the probability representing the importance of node features is defined as follows:

$$\mathcal{P}_{fi} = min(\frac{W_{max} - W_i}{W_{max} - \mu_{fs}} \bullet p_f,\ p_\tau) \tag{4}$$

where W_{max} and μ_{fs} is the maximum and the average value of W_i, and p_f is a hyper-parameter that controls the overall scope of feature augmentation.

Finally, we generate two corrupted graph views G_1, G_2 by jointly performing edge-level and node attribute-level augmentations, with different probabilities p_e and p_f for generating the two views, thus providing diverse context for contrastive learning. Then the two views are fed into a GNN encoder to obtain their representations as $U = \mathcal{F}(G_1)$ and $V = \mathcal{F}(G_2)$. We chose transformerConv [26] as the GNN encoder due to it incorporates self-attention mechanism and performs well in node classification tasks.

Contrastive objective With the two view representations U and V, we apply a contrastive objective to distinguish the representations of the same node from other node representations. Specifically, for node i in the social network graph G, its representations in the two views u_i and v_i form a positive pair, while the representations of the other nodes in the two views are regarded as negative samples. Then we apply the InfoNCE loss for a positive pair, which is denoted as $\ell(u_i,\ v_i)$. Since two views are symmetric, the loss for another view is defined similarly for $\ell(v_i,\ u_i)$. The overall objective to be maximized is then defined as the average over all positive pairs, formally given by:

$$\mathcal{L} \ = \ \frac{1}{2N} \sum_{i=1}^{N} [\ell(u_i, v_i) + \ell(v_i, u_i)] \tag{5}$$

Contrastive learning encoding By conducting sufficient contrastive learning on unlabeled social network graph data using the GCL framework, we obtain a contrastive learning encoder $GCLEncoder$. We then apply this encoder to the current snapshot of the dynamic graph G^{t_N} to obtain the most recent representations of the nodes $x_i^{t_N}$ in the social network graph, as shown below:

$$x_i^{t_N} \ = \ GCLEncoder(x_{i(init)}^{t_N}) \tag{6}$$

3.4 Graph temporal patterns extraction

In order to utilize the historical contextual information of dynamic social network graphs, we propose a self-attention-based graph temporal patterns extraction model to fully make use of the dynamic nature of social networks for the detection of social bots. Specifically, the model takes a sequence composed of the initialized representations of node V_i across L length historical snapshots as input, denoted as $\left\{S_i^{t_1}, S_i^{t_2}, \cdots, S_i^{t_L},\right\}$. The module outputs a new sequence of user representations.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1

Position Embedding Since the self-attention mechanism is unaware of the nodes' ordering information, we introduce absolute and evolving temporal position embedding to obtain temporal information in the sequence of graph snapshots that can effectively reflect the dynamic nature of social networks.

First, the absolute temporal position of each snapshot was embedded as a basis to capture ordering information as :

$$p^{t_k,\ AT} = \ E_{AT}(t_k) \tag{7}$$

where $P_{Ab}^{t_k}$ denotes the absolute temporal position embedding for the timestamp t_k and E_{AT} denotes the trainable absolute temporal position embedding parameter.

Second, Yang's work [22] shows that the two temporal signals: local clustering coefficient and bidirectional links ratio have significant utility in countering the disguised social bots. We embed these two signals to reveal the evolving behavior patterns in the social graph over time.

Local Clustering Coefficient (LCC) measures the degree to which a node's neighbors are interconnected. In social networks, genuine users tend to interact with acquaintances to form highly interconnected communities, while social bots are usually associated with randomly selected neighbors who lack close connectivity. Therefore, LCC shows a significant difference between human users and social bots in social network. The position embedding of the LCC is calculated as follows:

$$p_i^{t_k,\ LCC} = E_{LCC}(\frac{2 * \left|e_{v_i}^{t_k}\right|}{k_{v_i}^{t_k} * (k_{v_i}^{t_k}\ -\ 1)}) \tag{8}$$

where $\left|e_{v_i}^{t_k}\right|$ is the number of edges between neighbors of node v_i at the timestamp t_k, and $k_{v_i}^{t_k}$ is the sum of the indegree and outdegree of node v_i at the timestamp t_k. E_{LCC} denotes the trainable local clustering coefficient embedding parameter.

Bidirectional Links Ratio (BLR) could effectively assess the reciprocity between an account and its followings in social network. A bidirectional link appears when two accounts mutually follow each other. Genuine users in social networks tend to have a higher bidirectional link counts as they follow each other within their circles of acquaintances. In contrast, social bots tend to have a relatively lower bidirectional link counts due to their indiscriminate following behavior and lack of reciprocal connections. Therefore, BLR is an important metric for distinguishing genuine users from social bots in social networks. The position embedding of the bidirectional links ratio is calculated as follows:

$$p_i^{t_k,\ BLR} \ = E_{BLR}(\ \frac{N_{blks}(v_i^{t_k})}{N_{lks}(v_i^{t_k})}) \tag{9}$$

where $N_{blks}(v_i^{t_k})$ and $N_{lks}(v_i^{t_k})$ denote the numbers of bidirectional links and all links of node v_i at the timestamp t_k. E_{BLR} denotes the trainable bidirectional links ratio embedding parameter.

Temporal Attention We utilize a multi-head temporal attention mechanism to extract the intrinsic pat-terns of dynamic social network graphs from historical snapshots. Specifically, we first aggregate the output of the positional embedding as follows:

$$S_i^{t_k} = p_i^{t_k,\ AT} \ + \ p_i^{t_k,\ LCC} + \ p_i^{t_k,\ BLR} \tag{10}$$

We then pack the representations of node v_i together across the timestamps, which is denoted as $\hat{S}_i \in R^{T \times F}$. Finally, we perform multi-head temporal attention to obtain the temporal patterns embeddings of the dynamic graph:

$$\hat{Z}_i = \ Con_{d=1}^{L}(softmax(\frac{Q_{d,i}K_{d,i}^T}{\sqrt{F}} \ + \ Mask) \cdot V_{d,i}) \tag{11}$$

where L is the count of graph snapshots, Con represents the concatenation operation, Q, K, V are the queries, keys and values transformed by trainable parameters $W_* \in R^{F \times F}$ respectively. $Mask \ \in R^{T \times T}$ is a sequence mask matrix that make the node at timestamp t_k only attends over its historical node representation.

3.5 Learning and Optimization

Our goal is to distinguish social bots from human users by learning generalization patterns and temporal patterns in the social network graph. To this end, we obtain the final node structural embedding x_i^{last} and temporal patterns embedding z_i^{last} at last snapshot of social graph, and transform them with a linear layer and a softmax layer to get social bot detection result as follows:

$$\hat{y}_i = softmax(W2 \bullet (\sigma(W1 \bullet [x_i^{last}, z_i^{last}] + b_1)) + b_2) \tag{12}$$

where $\hat{y}_i$ is out model's prediction of user i, W_* and b_* are learnable parameters, and $[,]$ is the concatenation operation. We then adopt supervised annotations and a regularization term to train out model, formulated as:

$$Loss \ = -\sum_{i \in Y}[y^i log(\hat{y}_i) + (1 - y^i) log(1 - \hat{y}_i)] + \lambda \sum_{\omega \in \theta} \omega^2 \tag{13}$$

where Y is the annotated user set, y^i is the ground-truth labels, θ denotes all trainable parameters in the model and λ is a hyperparameter.

4 Experiments

4.1 Dataset

In order to verify the effectiveness of our proposed model BotTGCL, the data set needs to have a certain graph structure type. We conducted our experiments on two public data sets with topological relationships, TwiBot-20 [27] and TwiBot-22 [28], which are more representative of the current social network environment. TwiBot-20 and TwiBot-22 are comprehensive Twitter bot detection benchmarks and provide user follow relationships to support graph-based methods. TwiBot-20 is proposed in 2020 and includes 229,573 Twitter users, which contains labeled users of 5,237 social bots and 6589 humans. TwiBot-22 is the largest public dataset to date for Twitter bot detection and includes 1,000,000 Twitter users, which contains labeled users of 139,943 social bots and 860,057 humans. It's worth noting that Twibot-22 suffers from a class imbalance issue, where the number of humans is significantly larger than that of social bots.

4.2 Baselines and experiment setting

We evaluate our proposed social bot detection framework BotTGCL along with several representative baselines on the two benchmarks as follows:

Yang et al. (2020) [5] use random forest classifier with minimal user metadata and derived features.

Kudugunta et al. (2018) [6] propose to jointly leverage user tweet semantics and user metadata.

Botometer (2016) [7] is a bot detection service that leverages more than 1,000 user features.

Alhosseini et al. (2019) [9] use graph convolutional networks to learn user rep-resentations and conduct spam bot detection.

BotRGCN (2021d) [10] constructs a heterogeneous graph to represent the Twittersphere and adopts relational graph convolutional networks for representation learning and bot detection.

BotDGT (2024) [25] is the state-of-the-art dynamic graph-based model, which characterizes a social network as a dynamic graph and proposes a temporal module to integrate historical context and model the evolving behavior patterns exhibited by social bots and legitimate users.

RGT (2022) [12] proposes relational graph transformers to model heterogeneous influence between users and use semantic attention networks to aggregate messages across users and relations and conduct heterogeneity-aware Twitter bot detection.

We use pytorch [23], torch geometric [24] for an efficient implementation of our proposed social bot detection framework. We conduct all experiments on a server with 2 Tesla V100 GPUs with 32 GB memory, 32 CPU cores, and 300GB CPU memory. According to the excellent performance of He et al.[25] in social bot detection on dynamic graph, we set the time interval granularity to 12 months. To directly and fairly comparing with previous works, we follow the same train, valid and test splits provided in the benchmark.

4.3 Main Results

We evaluate our proposed social bot detection framework BotTGCL along with 8 representative baselines on the two benchmarks and present the results in Table 1. It is demonstrated that graph-based methods for social bot detection, such as BotTGCL(Ours), botDGT [25], RGT [12], generally obtain higher classification effectiveness compared to traditional methods like Yang et al [5], Kudugunta et al [6] and Botometer [7]. This under-scores the critical importance of leveraging the topological structure for node classification tasks in social networks.

Table 1. Accuracy and binary F1-score of different social bot detection methods on TwiBot-20 and TwiBot-22. Bold indicates the best performance, underline the second-best. Our method is significantly better than the second-best baseline.

Model	**TwiBot-20**		**TwiBot-22**	
	Accuracy	F1-score	Accuracy	F1-score
Yang et al.	81.91	85.46	75.08	36.59
Kudugunta et al.	81.74	75.15	65.78	51.67
Botometer	48.01	62.66	49.9	42.75
Alhosseini et al.	68.13	73.18	47.72	38.1
BotRGCN	84.62	87.07	78.87	54.99
BotDGT	87.25	88.87	79.33	58.15
RGT	86.64	88.2	76.5	42.94
BotTGCL (Ours)	**87.72**	**90.51**	**79.45**	**58.23**

Our proposed BotTGCL outperforms the state-of-art graph-based baseline models in terms of accuracy and F1-score on both two comprehensive social bot detection benchmarks. In comparison with the previous dynamic graph-based methods, BotTGCL not only incorporates the temporal patterns of the dynamic social network captured by a temporal module, but also leverages the topological structure of the social network extracted from unlabeled social graphs. The superior performance of our approach BotTGCL could be attributed to its ability to integrate the generalization patterns in the structure of unlabeled social graph and the behavioral patterns in historical context of evolving social network, which can effectively distinguish the social bot accounts that disguise themselves by imitating human behavior in real-world with scarce labeled data.

4.4 Ablation Study

Table 2. Ablation study for BotTGCL

Ablation Settings	**TwiBot-20**	
	Accuracy	**F1-score**
BotTGCL	**87.72**	**90.51**
w/o graph structure contrastive learning	85.87	87.32
replace graph structure contrastive learning w/ RGCN	86.41	87.88
replace graph structure contrastive learning w/ RGT	86.28	86.91
w/o Graph temporal patterns extraction	86.18	88.17
w/o Position Embedding	86.89	88.57
w/o Temporal Attention	86.72	88.53
replace Graph temporal patterns extraction w/ RNN	86.81	88.46
replace Graph temporal patterns extraction w/ LSTM	87.15	88.72

In order to prove the effectiveness of each part of BotTGCL, we conduct an ablation study on BotTGCL by removing or replacing one specific component at a time to access its significance. The components validated in this section are the graph structure contrastive learning module and the temporal patterns extraction module. The experimental results of these ablation models on TwiBot-20 are shown in Table 2.

In the graph structure contrastive learning module, we propose a graph augmentation model that performs edge deletion and node masking according to the importance of the edge and node attributes on social graph. We first remove the graph augmentation model, the performance experiences significant degradation compared to the original BotTGCL, indicating the importance of graph structure contrastive learning for effective bot detection. We then replace the graph augmentation model with other static graph-based methods proposed to detect social bot, including BotRGCN [10] and RGT [12]. The performance degradation observed indicates that our proposed graph augmentation model better captures the underlying topological structure of the social network.

In the graph temporal patterns extraction module, we propose a self-attention based temporal patterns learning model to fully utilize the dynamic behavior of social network graphs for social bot detection. We first remove the graph temporal patterns extraction module, positional embeddings and temporal attention respectively, the performance experiences have decreased to varying degrees compared with the original BotTGCL, which shows the importance of our temporal module for social bot detection. We then replace the self-attention based temporal patterns learning model with general recurrent neural networks for temporal analysis, including RNN and LSTM, the observed performance degradation demonstrates that our proposed model better capture the deep temporal patterns of dynamic social networks.

To sum up, both our proposed graph structure contrastive learning module and graph temporal patterns extraction module are effectiveness and contribute to our model's outstanding performance.

4.5 Label Robustness Study

In order to validate the effectiveness of our model BotTGCL in real-world with a lack of high-quality labeled data, we conduct label robustness experiments. To simulate the scenario with few labels, we choose the dataset TwiBot-20 and only randomly select 1% - 10% labels from the training set to train the models and verify the performance on the test set.

The result is presented in Figure 2. when trained with only 1% of the labeled data, our model exhibits a significant drop in accuracy and F1-score, but it still substantially outperforms the baseline models. As the proportion of labeled data gradually increases, the performance of our model also steadily rises. By the time the ratio of labeled data reaches 10%, the accuracy of our model in identifying social bots has reached 86.73%, surpassing the most baseline models when trained on the full set of labeled data. This shows that even with very little labeled data, our model still outperforms other baselines. Experimental results also demonstrate that our BotTGCL model could learn more generalized underlying knowledge about the topological structure and dynamic behavior in networks for social bot detection than baseline methods under the same amount of labeled data.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1

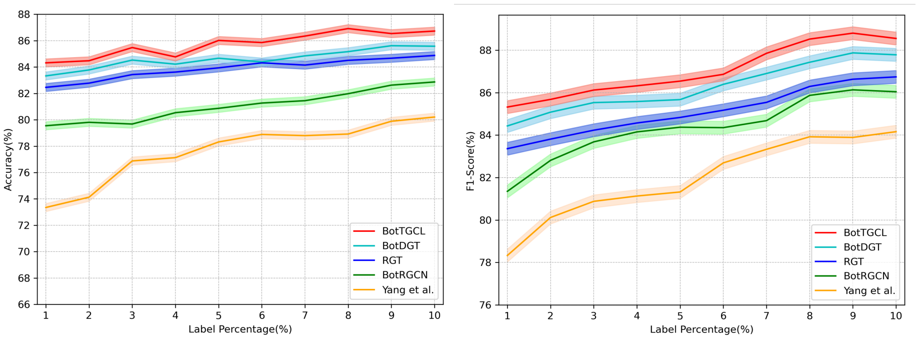

Fig. 2. Label Robustness study, even with very little labeled data, our model BotTGCL still outperforms other baselines

Table 3. Few-shot detection study for BotTGCL

K-shot	1 shot	5 shot	10 shot	15 shot	20 shot	25 shot
Accuracy	57.66	64.27	74.38	76.43	77.63	78.46
F1-Score	48.31	49.82	55.18	56.36	56.42	57.13

4.6 Few-shot Detection Study

Newly emerging social bots often lack adequate labeled data for training, making them difficult to detect using state-of-the-art social bot detection models. To evaluate our model to identify these novel social bots, we conducted a few-shot detection study. Specifically, we first divide the TwiBot-22 dataset into 10 relatively independent subcommunity categories following the method described by Feng et al.[10], and then randomly sample k labels per category from the training set to fine-tune our model with the graph structure contrastive learning module, and then test it in the test set. As shown in Table 3, when k is reduced to 1, one-shot detection, the performance has decreased significantly. However, as k increases, the detection performance of BotTGCL rapidly improves, and when $k = 15$, its performance exceeds that of most baseline models.

The experimental results demonstrate that our model acquires valuable structure knowledge from the unlabeled social graph and integrates the dynamic behavior patterns of the social network. These factors collectively reduce the model's dependence on labeled data and enhance its generalization capability in detecting new novel social bots.

5 Conclusion and Future Work

Due to the lack of high quality labeled data in the real world, advanced supervised social bot detection methods face restrictions in detection performance. These methods also typically describe social networks as static graphs, overlooking their dynamic nature. To address the challenges above, we propose a novel social bot detection framework BotTGCL. We represent a social network as a dynamic graph and construct snapshots of the social network graph at a certain time interval. Then we obtain the underlying knowledge from unlabeled data by graph structure contrastive learning, and extract the valuable temporal patterns from dynamic graph by a temporal self-attention mechanism. After that, we fuse the graph structural patterns and the temporal patterns to differentiate social bots from genuine users. Extensive experiments demonstrate that our model acquires valuable structure knowledge from unlabeled social graph and integrates the dynamic behavior patterns of the social network, and is more efficient and generalized than the state-of-the-art baseline methods in real-world scenarios with scarce labeled data. In the future, we plan to experiment with more diversified ways to face the dynamic update scenario of social networks and extend our social bot detection approach.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (NSFC) (Grant 62376265).

References

1. Jarynowski, A.: Conflicts driven pandemic and war issues in social media via multi-layer approach of german twitter. Interdisciplinary Research pp. 1–9 (2022)
2. Woolley, S.C.: Automating power: Social bot interference in global politics. First Monday (2016)
3. Ferrara, E.: Social bot detection in the age of chatgpt: Challenges and opportunities. First Monday (2023)
4. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: Dna-inspired online behavioral modeling and its application to spambot detection. IEEE Intelligent Systems **31**(5), 58–64 (2016)
5. Yang, K.C., Varol, O., Hui, P.M., Menczer, F.: Scalable and generalizable social bot detection through data selection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 1096–1103 (2020)
6. Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Information Sciences **467**, 312–322 (2018)
7. Davis, C.A., Varol, O., Ferrara, E., Flammini, A., Menczer, F.: Botornot: A system to evaluate social bots. In: Proceedings of the 25th international conference companion on world wide web. pp. 273–274 (2016)
8. Wei, F., Nguyen, U.T.: 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). pp. 101–109. IEEE (2019)
9. Ali Alhosseini, S., Bin Tareaf, R., Najafi, P., Meinel, C.: Detect me if you can: Spam bot detection using inductive representation learning. In: Companion proceedings of the 2019 world wide web conference. pp. 148–153 (2019)
10. Feng, S., Wan, H., Wang, N., Luo, M.: Botrgcn: Twitter bot detection with relational graph convolutional networks. In: Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining. pp. 236–239 (2021)
11. Feng, S., Tan, Z., Li, R., Luo, M.: Heterogeneity-aware twitter bot detection with relational graph transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3977–3985 (2022)

12. Lei, Z., Wan, H., Zhang, W., Feng, S., Chen, Z., Li, J., Zheng, Q., Luo, M.: Bic: Twitter bot detection with text-graph interaction and semantic consistency. arXiv preprint arXiv:2208.08320 (2022)
13. Dukić, D., Keča, D., Stipić, D.: Are you human? detecting bots on twitter using bert. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA). pp. 631–636. IEEE (2020)
14. Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., Wang, K., Tang, J.: Gcc: Graph contrastive coding for graph neural network pre-training. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1150–1160 (2020)
15. Yanqiao, Z., Yichen, X., Feng, Y., Qiang, L., Shu, W., Liang, W.: Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131 (2020)
16. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: International conference on machine learning. pp. 4116–4126. PMLR (2020)
17. Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., Zhu, Y.: Sub-graph contrast for scalable self-supervised graph representation learning. In: 2020 IEEE international conference on data mining (ICDM). pp. 222–231. IEEE (2020)
18. Mavromatis, C., Karypis, G.: Graph infoclust: Maximizing coarse-grain mutual information in graphs. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 541–553. Springer (2021)
19. Kim, S., Yun, S., Kang, J.: Dygrain: An incremental learning framework for dynamic graphs. In: IJCAI. pp. 3157–3163 (2022)
20. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C.: Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 5363–5370 (2020)
21. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the web conference 2021. pp. 2069–2080 (2021)
22. Yang, C., Harkreader, R., Gu, G.: Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Transactions on Information Forensics and Security **8**(8), 1280–1293 (2013)
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
24. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428 (2019)
25. He, B., Yang, Y., Wu, Q., Liu, H., Yang, R., Peng, H., Wang, X., Liao, Y., Zhou, P.: Dynamicity-aware social bot detection with dynamic graph transformers. arXiv preprint arXiv:2404.15070 (2024)
26. Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., Sun, Y.: Masked label prediction: Unified message passing model for semi-supervised classification. arXiv preprint arXiv:2009.03509 (2020)
27. Feng, S., Wan, H., Wang, N., Li, J., Luo, M.: Twibot-20: A comprehensive twitter bot detection benchmark. In: Proceedings of the 30th ACM international conference on information & knowledge management. pp. 4485–4494 (2021)
28. Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., Zheng, Q., Zhang, W., Lei, Z., Yang, S., et al.: Twibot-22: Towards graph-based twitter bot detection. Advances in Neural Information Processing Systems **35**, 35254–35269 (2022)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97564-6_1