Multi-modal Social Bot Detection: Learning Homophilic and Heterophilic Connections Adaptively

Shilong Li
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
lishilong22@mails.ucas.ac.cn

Boyu Qiao
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
qiaoboyu@iie.ac.cn

Kun Li*
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
likun2@iie.ac.cn

Qianqian Lu
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
luqianqian@iie.ac.cn

Meng Lin
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
linmeng@iie.ac.cn

Wei Zhou*
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
zhouwei@iie.ac.cn

ABSTRACT

The detection of social bots has become a critical task in maintaining the integrity of social media. With social bots evolving continually, they primarily evade detection by imitating human features and engaging in interactions with humans. To reduce the impact of social bots imitating human features, also known as feature camouflage, existing methods mainly utilize multi-modal user information for detection, especially GNN-based methods that utilize additional topological structure information. However, these methods ignore relation camouflage, which involves disguising through interactions with humans. We find that relation camouflage results in both homophilic connections formed by nodes of the same type and heterophilic connections formed by nodes of different types in social networks. The existing GNN-based detection methods assume all connections are homophilic while ignoring the difference among neighbors in heterophilic connections, which leads to a poor detection performance for bots with relation camouflage. To address this, we propose a multi-modal social **bot** detection method with learning **h**omophilic and **h**eterophilic connections adaptively (**BothH** for short). Specifically, firstly we determine whether each connection is homophilic or heterophilic with the connection classifier, and then we design a novel message propagating strategy that can learn the homophilic and heterophilic connections adaptively. We conduct experiments on the mainstream datasets and the results show that our model is superior to state-of-the-art methods.

*Corresponding author.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3612569

CCS CONCEPTS

• **Human-centered computing** → *Collaborative and social computing*; • **Computing methodologies** → *Artificial intelligence.*

KEYWORDS

Social Bot Detection, Graph Neural Networks, Homophilic, Heterophilic

ACM Reference Format:
Shilong Li, Boyu Qiao, Kun Li, Qianqian Lu, Meng Lin, and Wei Zhou. 2023. Multi-modal Social Bot Detection: Learning Homophilic and Heterophilic Connections Adaptively. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612569

1 INTRODUCTION

Social bots are automated accounts that imitate human behavior on social media platforms [23, 34, 38]. Due to their ability to disseminate false information and manipulate public opinion, social bots have attracted a lot of attention on online platforms [2, 16, 17]. Therefore, the detection of social bots is a critical problem that needs to be tackled.

To evade detection, social bots can imitate human features for disguising themselves through feature camouflage and interact with humans for establishing relations with human accounts through relation camouflage [43]. Feature camouflage involves modifying various aspects of features, thereby achieving better deception effects. Typically, social bots can camouflage their account information, behavior pattern, location, and so on.

To reduce the impact of feature camouflage, existing methods mainly utilize multi-modal user information for detection, especially GNN-based methods that utilize additional topological structure information. Early social bot detection methods extracted user semantic and property information from tweets and metadata through feature engineering for detection[9]. With the advent

of deep learning, social bot detection based on neural network has gained increasing popularity recently. [11, 26] proposed using LSTM to extract semantic information from the content of tweets. SATAR [13] extracted user features with the self-supervised method. However, these methods only focus on the sequence features of tweets, but ignore the topological structure information[7]. In recent years, graph neural networks (GNNs) [1, 12, 15] have been applied to the detection of social bots, which can improve the detection performance by leveraging more comprehensive user information, including user semantic, property, and topological structure information.

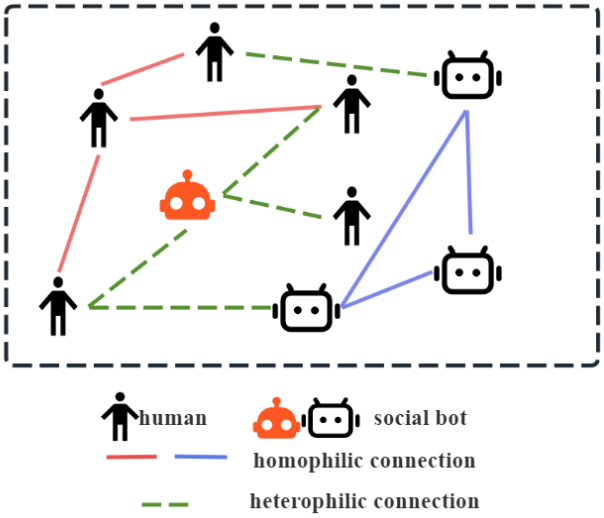

Figure 1: Three types of connections are distinguished by the type of node being connected.

However, these methods ignore the relation camouflage of social bots caused by interactions with humans. As shown in Figure 1, from the perspective of connected node types, there are actually three types of connections: the connection between two humans, the connection between two social bots, and the connection between a human and a social bot. Humans often establish connections with other humans related to themselves; social bots are interconnected to expand their discourse influence. These two types belong to the connection formed between nodes of the same type, which is called **homophilic** connection. The connection between a human and a bot which is formed between nodes of different types is called **heterophilic** connection. To verify the widespread existence of heterophilic connections in social networks, we calculate the proportion of homophilic and heterophilic connections in the mainstream datasets. The detailed statistics are presented in section 6.1.

The existing GNN-based social bot detection methods treat all connections as homophilic connections while ignoring the difference of neighbors in heterophilic connections [44]. In the GNN aggregating process, existing GNN-based social detection methods smooth the features of neighbor nodes and preserve the commonality of the neighboring node features regardless of the node types[40]. The existence of heterophilic connections would limit the learning ability of GNNs, leading to a significant decline in the detection performance. It is particularly challenging for some wily social bots such as the bot marked in orange in Figure 1 which are trapped in neighborhoods that only have humans connected to them, further increasing the difficulty of detection. Therefore, developing effective methods that can learn homophilic and heterophilic connections adaptively is crucial for social bot detection.

In order to solve this problem, we propose a multi-modal social **bot** detection method with learning **h**omophilic and **h**eterophilic connections adaptively (**BothH** for short). Specifically, firstly we construct a combination graph which consists of the original graph and node similarity graph to transmit similar information to some wily bots. Then we determine whether each connection is a homophilic or heterophilic connection with the connection classifier. Finally, we design a novel message propagating strategy that can learn the homophilic and heterophilic connections adaptively on the combination graph.

The following section provides an overview of our work's main contributions:

- As far as we know, we are the first to consider the relation camouflage of heterophilic connections for social bot detection.
- We design a novel message propagating strategy that can learn homophilic and heterophilic connections adaptively.
- We conduct experiments on the mainstream datasets, and the results show that our method is superior to the state-of-the-art methods.

2 RELATED WORKS

2.1 Social Bot Detection

Traditional social bot detection methods mainly rely on feature engineering and machine learning classifiers. [3, 4, 42] used the metadata information on the user's profile. [9, 18, 25] utilized the textual features of users by analyzing their tweets. As deep learning later shows great promise and gains popularity, recent research has shifted towards applying deep learning methods to detect bots. [26] utilized recurrent neural networks to capture the tweet features. [13] employed a self-supervised framework to identify social bots. [27] raised the issue of semantic inconsistencies between tweets posted by bots and those from humans. However, these methods only focus on the sequence features of tweets, but ignore the topological structure information[7]. To capture the topological structural information, the graph neural networks (GNNs) [1] have been used in social bot detection based on the social network structure. [15] considered the differences in social relations by constructing a heterogeneous graph. [12] identified and extracted user relationship and influence heterogeneity and achieve the state-of-the-art performance on social bot detection.

The existing GNN-based methods leverage multi-modal user information for detection while ignoring the relation camouflage[43] of social bots. In order to solve this problem, we propose a social bot detection method with learning homophilic and heterophilic connections adaptively.

2.2 Heterophily-based Graph Neural Network.

Considering the limitations of most GNNs based on the assumption of homophilic connections, some GNN models also consider heterophilic connections. There are two ways to solve this problem:(1) By changing the graph structure, the original heterophilic graph is transformed into a homophilic graph. [29, 33] reconstructed a homophilic graph through the structure information. [6, 10] removed heterophilic edges to form a homophilic graph. (2) Improving the message propagating mechanism of GNNs to enhance their expression ability. [45] extended GNNs to heterophilic graph by separating the node features and the neighbor nodes. [5, 35] conveyed opposite information with heterophilic connections. However, the existing social bot detection methods ignore the heterophilic connection caused by relation camouflage.

3 PROBLEM FORMULATION

In this section, we introduce the task of social bot detection with multi-modal user information, and the concept of homophilic and heterophilic connection.

Multi-modal social bot detection. We use a set $T = \{t_i\}_{i=0}^{K}$ to represent a user's profile description and tweets, where the first text content is the user's description and K denotes the number of tweets posted by the user. A single text content can be represented by $t_i = \{w_1^i, \cdots, w_Q^i\}$. The user's metadata is represented by M, which is divided into value type data denoted by V and Boolean type data denoted by B. The user information X consists of the text information and the metadata information. We use R to represent the different relations between users, E to represent the connections between users, and Y to represent the labels of users. Besides, humans are considered as positive nodes with the label 0, while social bots are considered as negative nodes with the label 1.

Therefore, given a graph $G = \{V, X, E_r|_{r=0}^{R}, Y\}$, the social bot detection model aims to the labels of the test set $\hat{Y}_{test}$ based on the training set : $f(G) \rightarrow \hat{Y}_{test}$.

Homophilic and heterophilic connection. An edge is considered homophilic if the two nodes connected by it are of the same type, whereas an edge is considered heterophilic if the two nodes are of different types. We use $\{E_r^+\}|_{r=0}^{R}$ to represent the homophilic connections and $\{E_r^-\}|_{r=0}^{|R|}$ to represent the heterophilic connections. In particular, there are two types of user labels in social networks, namely human accounts and social bot accounts. Therefore, for social bot detection, homophilic connections include the connections between two humans and the connections between two social bots while heterophilic connections include the connections between a human and a bot.

4 METHODOLOGY

4.1 Overview

Figure 2 shows an overview of our method, which comprises the user node information encoder, combination graph constructor, connection classifier, and neighbor aggregator module. Firstly, we use the multi-modal user node information encoder to obtain the user node representation, then we construct the combination graph which consists of the original graph and node similarity graph considering some wily social bots. After that, the homophilic and heterophilic connection classifier is leveraged to determine whether each connection in the graph is homophilic or heterophilic. And then we propose a message propagating strategy to update node features with the neighbor aggregator. Finally, the nodes are classified as either social bots or humans.

4.2 Multi-modal User Representation

We jointly use the multi-modal information for user representation, including the semantic information from user text content, the property information from user metadata, and the user topological structural information from social networks. The user semantic and property information are used to obtain the user node representation through the node information encoder. Then the user topological structural information and user node representation are used together to construct the graph.

4.2.1 User Node Information Encoder.

User semantic information encoder. The user semantic information can be obtained from the user tweets and descriptions on the profile. We use the pre-trained language model to encode the text content, which can be represented as follows:

$$\{\hat{t}_0, \cdots, \hat{t}_K\} = AvgPool(LM(\{w_1^i, \cdots, w_Q^i\}_{i=0}^{K})), \tag{1}$$

where LM represents the pre-trained language model, $\{\hat{t}_0, \cdots, \hat{t}_K\}$ represents the embedding of text contents.

In order to represent the user semantic information, we process all the text content of a user with average pooling, which can be defined as:

$$\hat{T}_i = AvgPool(\{\hat{t}_0, \cdots, \hat{t}_K\}), \tag{2}$$

where $\hat{T}_i$ denotes the user semantic information.

Considering the feature camouflage of social bots[43], we adopt a parameterized approach to reduce the impact of camouflage:

$$\tilde{T}_i = \sigma(W_s \hat{T}_i), \tag{3}$$

where W_s is the semantic information parameter matrix, σ is nonlinear activation function.

User property information encoder. The user property information is mainly extracted from the metadata of user profiles. The user's metadata can be divided into value type data **V** and Boolean type data **B**. In order to reduce the impact of feature camouflage, similar to semantic information encoder, we process user property information as follows:

$$\tilde{V}_i = \sigma(W_v V_i), \tag{4}$$

$$\tilde{B}_i = \sigma(W_b B_i), \tag{5}$$

where W_v and W_b are the value type and Boolean type property information parameter matrix, respectively. σ is a nonlinear activation function.

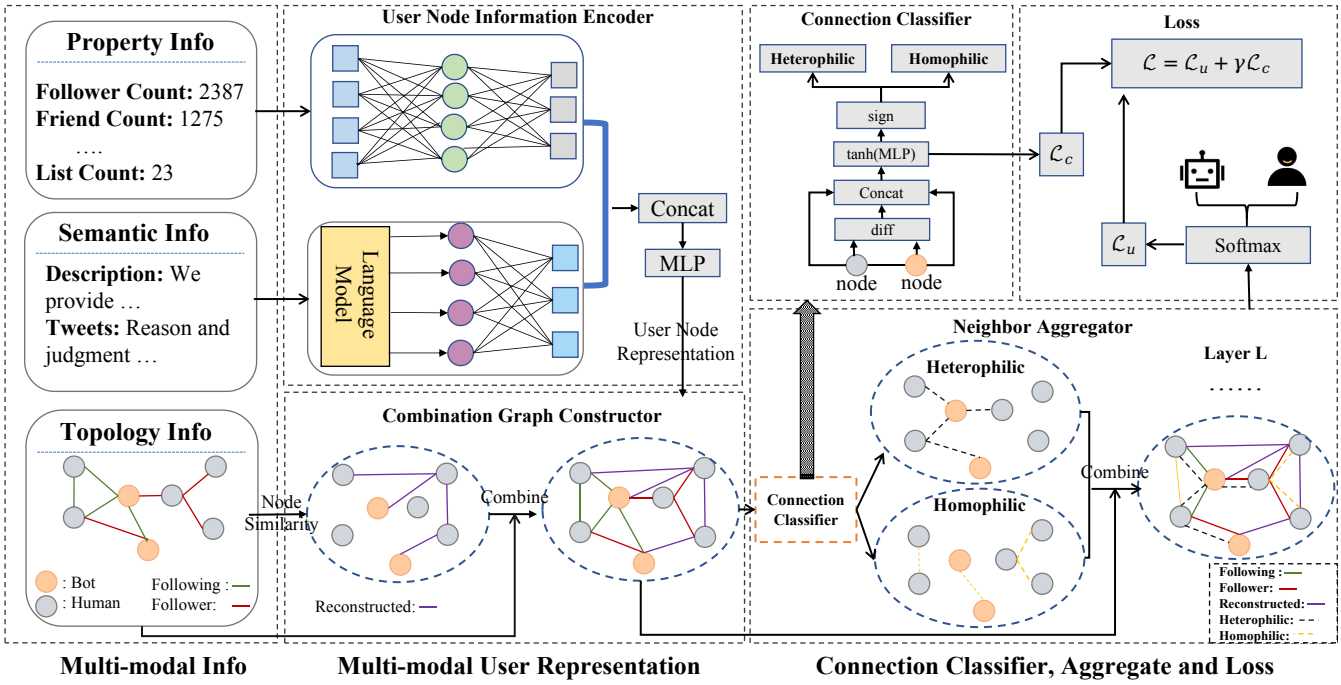

Figure 2: The overview of our proposed framework.

User Node Information Representation. We combine the property information, semantic information to obtain the user node information representation, which can be denoted as follows:

$$u_i = Concat([\tilde{V}_i; \tilde{B}_i; \tilde{T}_i]), \tag{6}$$

where u_i is the representation of user node i.

4.2.2 *Combination Graph Constructor.*

Original graph construction. We combine the user node information and topological structural information to construct the social graph $G = \{V, X, E_r|_{r=0}^{R}, Y\}$, in which the user is treated as the node and the topological structural information is treated as the edge.

Node similarity graph construction. Considering that some wily social bots are trapped in over many humans neighborhoods, these nodes can only obtain dissimilar information from neighboring nodes. To overcome this problem, we construct a node similarity graph based on the feature similarity among each node pair.

For a given node v_i, we obtain the similarity between v_i with other nodes. Their cosine similarity can be calculated as:

$$sim_{ij} = \frac{u_i^\top u_j}{||u_i|| ||u_j||}, \tag{7}$$

where sim_{ij} represents the similarity between node i and node j. Then, we select k most similar neighbors $v_{i1}, v_{i2}, \ldots, v_{ik}$ based on the similarity for each node pair. Therefore, we obtain the similarity connections E_s based on the similar node pairs. Combined the user node information, we construct the node similarity graph $G_s = \{V, X, E_s, Y\}$ by utilizing the node information U and the constructed connections E_s. We consider the edge type in the node similarity graph G_s as a new special relation. Due to our assumption that homophilic connections have similar node characteristics, all edges in the node similarity graph are considered as homophilic connections. The node similarity graph can effectively capture the similarities between different nodes, allowing for more accurate modeling of complex relationships within the graph. By constructing the node similarity graph, it not only addresses the issue of wily social bots whose H_E is 0 but also allows the discovery of potential neighbor nodes, thereby improving the expressive ability of GNN.

Graph Combination. In order to obtain both the structural information on the original graph and the information from possible distant but similar nodes, we combine the connections on the two graphs. The combined connections can be represented as:

$$E_c|_{r=0}^{|R+1|} = E_r|_{r=0}^{|R|} \cup E_s. \tag{8}$$

The combination graph can be represented as $G_c = \{V, X, E_c|_{r=0}^{|R|+1}, Y\}$. In the original graph, there are various different relations, each relation with both homophilic and heterophilic connections. While in the node similarity graph, there is only one new special relation, and all connections of this relation are treated as homophilic connections. It is worth noting that during the combination process, there may be some duplicated connections between the two graphs. For these connections, the connection types in these two graphs are different. Therefore, we remove these connections from the node similarity graph because these connection has been taken into consideration on the original graph. In other words, we do not

break the original connections, but add some new connections to the original graph.

4.3 Homophilic and Heterophilic Connection Classifier

The homophilic and heterophilic connection classifier aims to identify the type of each connection based on the difference between embeddings of two connected nodes. Inspired by [6, 35], we assume that homophilic connections correspond to similar features, whereas heterophilic connections correspond to dissimilar features. Therefore, We devise an end-to-end classification module to determine the connection type. As mentioned in section 4.3, we only differentiate the connections of the edge $E_r|_{r=0}^{|R|}$ in the original graph. Besides, the process of predicting connection types may lead to certain errors. We analyze the possible errors caused by the classifier and find that its impact is within an acceptable range.

In order to use more comprehensive information, we input two connected node features and the difference between them into a feedforward neural network to identify each connection. The process can be formulated as follows:

$$\begin{aligned}\mathbf{s}_{ij}^{(l)} = \mathbf{f}_s^{(l)}(\mathbf{W}_t^{(l)}\mathbf{u}_i^{(l)} \oplus \mathbf{W}_t^{(l)}\mathbf{u}_j^{(l)} \oplus \\ (\mathbf{W}_t^{(l)}\mathbf{u}_i^{(l)} - \mathbf{W}_t^{(l)}\mathbf{u}_j^{(l)})),\end{aligned} \tag{9}$$

where $\mathbf{f}_s^{(l)}$ represents a feedforward neural network composed of a linear layer and an activation function *tanh*, and $\mathbf{s}_{ij}^{(l)}$ represents the connection score. $\mathbf{u}_i^{(l)}$ and $\mathbf{u}_j^{(l)}$ are the embedding of node i and node j, respectively. $\mathbf{W}_t^{(l)}$ is a linear layer. Then we use $\mathbf{s}_{ij}^{(l)}$ to determine the type of each connection:

$$\mathbf{e}_{ij}^{(l)} = \begin{cases} homophilic\ \ connection, & \text{if } \mathbf{s}_{ij}^{(l)} > 0 \\ heterophilic\ \ connection, & \text{otherwise} \end{cases}$$

4.4 Neighbor Aggregator

Distinguish between homophilic and heterophilic connections. It is worth noting that there are connections of multiple relations in social network graph, such as following and following. This is why the social network graph is addressed as heterogeneous graph. The existing methods for heterogeneous graphs mainly rely on distinguishing the explicit different heterogeneous relations, while ignoring the implicit differences between homophilic and heterophilic connections. To account for this, we consider homophilic connections and heterophilic connections as separate types of connections under each relation. In order to represent different connection types, we use the connection embedding to learn their semantic information adaptively. Specifically, we apply the homophilic and heterophilic connection classifier to distinguish between homophilic connections and heterophilic connections under each relation in the original graph. Taking a certain relation $r_i \in R$ as an example, by using the connection classifier, we can obtain the homophilic connection r_i^+ and the heterophilic connection r_i^-. After each connection is distinguished, the set of the connections in the original graph is expanded from R to R^* which includes homophilic connection R^+ and the heterophilic connection R^- .

Attention weight. We refer to the attention mechanism in SimHGN [31] to aggregate the features from neighborhoods. We consider the attention scores to denote the influence of the neighbor nodes by the attention mechanism. Firstly, we calculate the attention scores through connection embedding and node embedding:

$$\begin{aligned} n_{ij} &= W_n[Wu_i \| Wu_j \| W_r r^*_{e(\langle i,j\rangle)}], \\ \hat{\alpha}_{ij} &= \frac{\exp\left(\text{LeakyReLU}\left(n_{ij}\right)\right)}{\sum_{k\in N_i}\exp\left(\text{LeakyReLU}\left(n_{ik}\right)\right)}, \end{aligned} \tag{10}$$

where $r^*_{e(\langle i,j\rangle)} \in R^*$ represents the expanded connection relation between node i and node j. u_i and u_j represents the embedding of node i and node j. N_i represents the neighbor node of node i. W_r and W_n are the learnable parameters. $\|$ represents the operation of concat. For the sake of conciseness, we omit the superscript (l) out of this equation.

Residual connection. In order to solve the problem of degradation of deep GNN performance due to over-smooth [28, 41], we apply residual connection [21, 22] to the attention score:

$$\alpha_{ij}^{(l)} = (1-\beta)\hat{\alpha}_{ij}^{(l)} + \beta\alpha_{ij}^{(l-1)}, \tag{11}$$

where the hyperparameter $\beta \in [0, 1]$ is a learnable parameter.

Neighbor nodes aggregation. Next, we need to aggregate neighbor nodes to obtain the updated node embedding:

$$u_i^{(l)} = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(l)}W^{(l)}u_j^{(l-1)} + W_{\text{res}}^{(l)}u_i^{(l-1)}\right), \tag{12}$$

where $\alpha_{ij}^{(l)}$ is the attention score of the edge $\langle i, j\rangle$ and $W_{\text{res}}^{(l)}$ is an learnable parameter.

4.5 Learning and Optimization

We adopt the user's feature representation in the last layer as the final embedding of nodes for social bot detection. The social bot detector aims to determine whether a user is a bot or not. Given y^i as the actual label and $\hat{y}^i$ as the predicted label. We use cross entropy Loss for classification:

$$\hat{y} = softmax(u^{(l)}), \tag{13}$$

$$\mathcal{L}_u = -\sum_{i\in G}(y_i \log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)). \tag{14}$$

To learn the loss of the homophilic and heterophilic connection classifier, we add an auxiliary loss with a supervision signal from the training nodes:

$$\mathcal{L}_c = \frac{1}{||E||}\sum_{e_{ij}\in E} max(0, 1 - s_{ij}^{(l)}y_{ij}), \tag{15}$$

where $s_{ij}^{(l)}$ is the result of inputting the edge between node i and node j into the connection classifier. y_{ij} is the true type of connection between node i and node j. For the edge connected by node i and node j, y_{ij} is as follows:

$$y_{ij} = \begin{cases} +1, & \text{if } label_i = label_j \\ -1, & \text{otherwise} \end{cases}$$

To sum up, the overall loss of our model consists of two parts, the node prediction loss $\mathcal{L}_u$ and the connection classifiers loss $\mathcal{L}_c$:

$$\mathcal{L} = \mathcal{L}_u + \gamma \mathcal{L}_c, \tag{16}$$

where γ is the trade-off coefficient.

5 EXPERIMENTAL SETUP

5.1 Dataset

Following the previous work[36, 37], we evaluate our approach on three real-world social bot detection datasets: Cresci-15 [8], MGTAB [36] and Twibot-20 [14]. These datasets are standard social bot detection benchmark that provide multiple relations to support graph-based models. In order to facilitate comparison with previous work, we construct the graph based on following and follower relations in all three datasets. The statistic of the datasets is shown in Table 1.

5.2 Baseline Methods

In order to make a comprehensive comparison, we choose the mainstream social bot detection methods as baselines. These methods are as follows:

- **Kugugunta *et al.*** [26] jointly encode the user metadata features and tweet content features to conduct social bot detection.
- **SGBOT** [42] mainly obtains the user features from metadata on the profiles and use the random forest classifiers for social bot identification.
- **Wei *et al.*** [39] utilize a recursive neural model with tweet contents to extract user semantic information and differentiate between bots and human accounts.
- **Liu *et al.*** [30] encodes user text information with pre-trained language model and classify with MLP layer.
- **BotRGCN** [15] constructs a relational graph network using the following and followed relations.
- **HGT** [24] considers the different type of connections by the heterogeneous mutual attention.
- **RGT** [12] detects social bot by distinguishing the differences of the relations and influence between user nodes.

5.3 Implementation Details

For user tweets, we use Roberta base model [30] to obtain the features of text content. For the construction of the original graph, we use follower and follower relations to obtain connections. For the activation function of the user information encoder, we use *Relu* for activation [20]. For the node similarity graph, cosine similarity is employed as the measure of node similarity. For the dataset Cresci-15 and Twibot-20, we set the number of similar neighboring nodes k to 2 which makes the number of node similarity connections similar to that of the original graph, and for the dataset MGTAB, we set k to 10 due to the graph scale and experimental efficiency. During the training phase, we adopt Adam as the optimizer, with batch size set to 256, and learning rate set to 1e-4. Besides, we set the parameter γ set to 0.7 after multiple experiments due to performance considerations.

Our model is implemented in PyTorch [32]. It is worth noting that when we train the homophilic and heterophilic connection classifier, we obtain the label of each connection type label y_{ij} based on the node label of the training set. Our experiments are conducted on Tesla V100. We train the models with up to 250 epochs and obtain the best model on the validation set. In order to directly compare our work with earlier research, we follow the same splits provided in the benchmark.

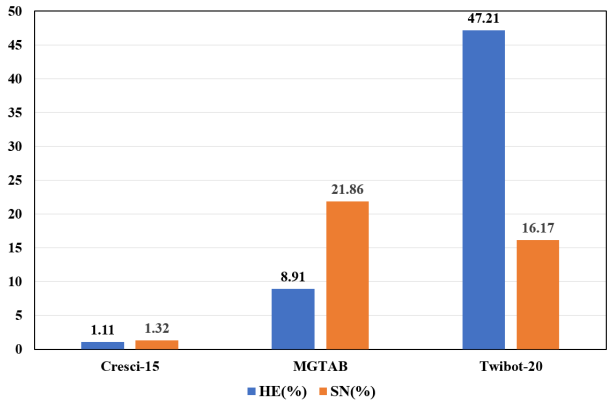

Figure 3: Heterophilic Evidence. The y-coordinate represents the percentage of *HE* and *SN* in the graph. *HE* represents the heterophilic connections. *SN* represents the special nodes whose all neighboring node labels are different from the central nodes.

6 EXPERIMENT AND ANALYSIS

In this section, we evaluate the effectiveness of our model under different settings. We want to answer the following questions:

- **RQ1:** Are heterophilic connections and special nodes whose all neighboring node labels are different from the central node widely present in social bot detection?
- **RQ2:** Does our model outperform the state-of-the-art methods for socail bot detection?
- **RQ3:** How do the different modules in our method benefit social bot detection?
- **RQ4:** What are the effects of homophilic and heterophilic connections under different relations?
- **RQ5:** What is the performance of user information in different modalities?

6.1 Measure Of Heterophily (RQ1)

To answer the RQ1, firstly we count the number of homophilic and heterophilic connections in the social network graph. The results are displayed in Table 3. We use the metric to measure the proportion of heterophilic connections as follows:

$$\mathcal{H}_E = \frac{|\{(v,u) \in E : y_v = y_u\}|}{|E|}. \tag{17}$$

where E is the set of the connections. Then, we calculate the percentage of special nodes whose all neighbor node labels are different from the central nodes. The results of the three datasets are shown

Table 1: The statistic of datasets.

Dataset	Training	Validation	Test	User	Human	Bot	Edge
Cresci-15	3,708	958	535	5,301	1,950	3,351	14,220
MGTAB	7,139	2,040	1,020	10,199	7,451	2,748	720,695
Twibot-20	8,278	2,365	1,183	11,826	5,237	6,589	15,434

Table 2: Bot detection performance on TwiBot-20 benchmark. Bold marks the highest score.

Method	Cresci-15		MGTAB		Twibot-20	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
Kudugunta *et al.*	75.32	75.54	69.22	68.42	59.59	47.26
SGBOT	77.10	77.91	81.60	84.90	74.76	78.69
Wei *et al.*	96.10	82.65	84.71	80.58	70.23	53.61
Liu *et al.*	97.01	95.86	86.27	82.57	75.55	73.09
BotRGCN	96.52	97.30	87.65	84.76	84.67	87.07
HGT	96.10	96.93	88.43	85.96	85.62	87.21
RGT	97.15	97.78	88.53	85.89	86.14	87.78
BothH	**98.80**	**98.87**	**90.29**	**87.54**	**90.33**	**91.27**

Table 3: Heterophilic connections Evidence.

Dataset	Homophilic	Heterophilic	Special nodes
Cresci-15	14062	158	70
MGTAB	656495	64200	2230
Twibot-20	8148	7286	1913

in Figure 3. Therefore, in order to evade detection, interactions between bots and humans have become increasingly frequent, which has also led to the widespread existence of heterophilic connections. Besides, the existence of special nodes further increases the difficulty of detection.

6.2 Performance Comparison (RQ2)

To answer the RQ2, we conduct experiments on the three datasets and compare it with recent social bot detection methods. There are slight differences in the experimental results compared to the original paper, and we argue that the reason for this may be due to different experimental environments. Table 2 shows our experimental results.

Our method significantly outperforms the baseline models on the three datasets. It is worth mentioning that compared to the state-of-the-art method RGT [12], our method achieves accuracy with 1.76% increase and F1-score with 1.65% increase on MGTAB, and achieves accuracy with 4.19% increase and F1-score with 3.49% increase on Twibot-20. The existing GNN-based social bot detection methods treat all connections as homophilic connections retaining the commonality of node features among the neighbors, which inevitably neglect the difference between homophilic connections and heterophilic connections. In contrast, our method can learn the effects of homophilic and heterophilic connection adaptively, which can effectively reduce the impact of relation camouflage of social bots.

These three datasets, from Cresci-15 to MGTAB, and then to Twibot-20, show an increasing proportion of heterophilic connections. Therefore, for the dataset Twibot-20 with the highest proportion of heterophilic connections, our model has the most significant improvement effect on this dataset, which verifies the effectiveness of distinguishing heterophilic connections.

Meanwhile, we can observe that the performance of the graph based method is significantly better than other methods, which proves the importance of modeling social network topology for social bot detection.

6.3 Ablation Study (RQ3)

To answer the RQ3, we conduct further analysis to obtain the impact of different modules. The results are shown in Table 4:

- To study the impact of the homophilic and heterophilic connection classifiers, we removed the connection classifiers from the model, which makes all connections treated as homophilic connections. The results demonstrate the effectiveness of capturing social bots by perceiving homophilic and heterophilic connections.
- In order to evaluate the effectiveness of the node similarity graph, we only process the original graph with our method, which results in a slight decrease of the performance. This indicates the effectiveness of providing similar information for nodes with over many neighbor nodes with different types.

Table 4: F1-score of the ablation experiment.

Model	Cresci-15	MGTAB	Twibot-20
w/o connection classifier	97.10	86.69	88.83
w/o similarity graph	98.29	87.02	90.42
BothH-deletion	97.36	85.48	88.51
BothH-weight	97.52	86.90	89.59
BothH	**98.87**	**87.54**	**91.27**

- In order to investigate the effectiveness of our proposed method with learning homophilic and heterophilic connections adaptively, we adopt two other typical strategies to our framework: deleting heterophilic connections directly [6] and conveying opposite information with heterophilic connections [5]. [19] mentions the harmful information spread by heterophilic connections in some domains, but the results compared to the strategy of deleting heterophilic connections directly and assigning negative weights to the heterophilic connections indicate that in the domain of social bot detection, the rational use of heterophilic connections can help detect social bots.

6.4 Relations Analysis(RQ4)

To answer RQ5, we conduct experiments under different relations. We use two types of relations in all three datasets, namely the follower relation and following relation. In the experiment, we only distinguish between homophilic and heterophilic connections for a certain relation, and do not differentiate the other relation. Figure 4 shows the experimental results under different relations. The experimental results indicate that processing all relations has the best performance. At the same time, we observe that BothH-following outperforms BothH-follower. We argue that it is related to the proportion of heterophilic connections under different relations. For following relation, social bots can actively follow humans, especially some celebrities, to avoid being detected. But for follower relation, few humans would actively become fans of social bots. Therefore, the proportion of heterophilic connection under the following relation may be greater than that in the following relation, which results in the performance under different relations.

6.5 Modalities Analysis(RQ5)

In order to evaluate the effectiveness of different modal information, we delete user property information, user semantic information and user topology structure information respectively, and compare the experimental results. Figure 5 shows the results. The results indicate that the impact of different modalities of information varies in different datasets. In Cresci-15 topological modal information contributes most to its performance, in MGTAB semantic modal information contributes most to its performance, and in Twibot-20 property modal information contributes most to its performance. This also indicates that different social bots may focus on different features for camouflage. Besides, we can find that deleting any modal information will lead to a decrease in the performance. Therefore, all the different modalities of user information in social bots have an impact on the detection.

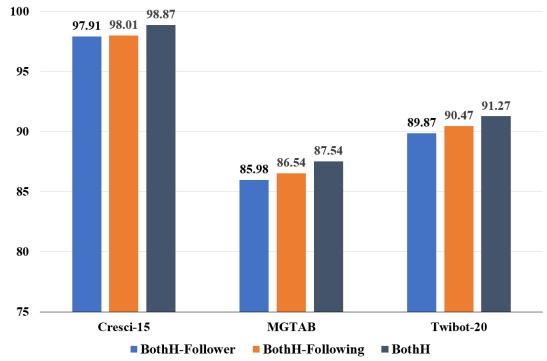

Figure 4: F1-score of the model under different relations.

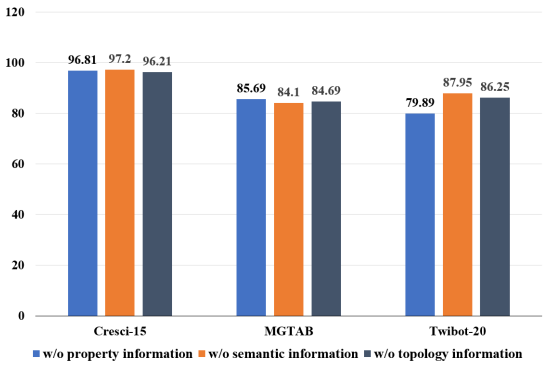

Figure 5: F1-score of the model with removing different modal information.

7 CONCLUSION

In this paper, we propose a GNN-based social bot detection method with learning homophilic and heterophilic connections adaptively. Specifically, firstly we determine whether each connection is a homophilic or heterophilic with the homophilic and heterophilic connection classifier, and then we design a novel GNN model that can learn the homophilic and heterophilic connections adaptively. To demonstrate the effectiveness of our proposed method, we conduct experiments on the mainstream datasets. The experimental results indicate that our method outperforms the state-of-the-art methods. In the future, we plan to further study the connection preferences of different types of social bots and expand our social bot detection methods.

ACKNOWLEDGMENTS

This research is supported in part by the National Key Research and Development Program of China (2022YFC3302102).

REFERENCES

[1] Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. 2019. Detect me if you can: Spam bot detection using inductive representation learning. In *Companion Proceedings of The 2019 World Wide Web Conference*. 148–153.

[2] Jonathon M Berger and Jonathon Morgan. 2015. The ISIS Twitter Census: Defining and describing the population of ISIS supporters on Twitter. (2015).

[3] David M Beskow and Kathleen M Carley. 2018. Bot-hunter: a tiered approach to detecting & characterizing automated activity on twitter. In *Conference paper. SBP-BRiMS: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*, Vol. 3. 3.

[4] David M Beskow and Kathleen M Carley. 2019. Its all in a name: detecting and labeling bots by their name. *Computational and mathematical organization theory* 25, 1 (2019), 24–35.

[5] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3950–3957.

[6] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. 2022. GCCAD: Graph Contrastive Learning for Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[7] Stefano Cresci. 2020. A decade of social bot detection. *Commun. ACM* 63, 10 (2020), 72–83.

[8] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems* 80 (2015), 56–71.

[9] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2016. DNA-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems* 31, 5 (2016), 58–64.

[10] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[11] Mohd Fazil, Amit Kumar Sah, and Muhammad Abulaish. 2021. Deepsbd: a deep neural network model with attention mechanism for socialbot detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4211–4223.

[12] Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022. Heterogeneity-aware twitter bot detection with relational graph transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3977–3985.

[13] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021. Satar: A self-supervised approach to twitter account representation learning and its application in bot detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3808–3817.

[14] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4485–4494.

[15] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021. 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.

[16] Emilio Ferrara. 2017. Disinformation and social bot operations in the run up to the 2017 French presidential election. (2017).

[17] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (2016), 96–104.

[18] Eric Ferreira Dos Santos, Danilo Carvalho, Livia Ruback, and Jonice Oliveira. 2019. Uncovering social media bots: a transparency-focused approach. In *Companion Proceedings of The 2019 World Wide Web Conference*. 545–552.

[19] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing Heterophily in Graph Anomaly Detection: A Perspective of Graph Spectrum. In *Proceedings of the ACM Web Conference 2023*. 1528–1538.

[20] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 315–323.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[22] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. 2021. RealFormer: Transformer Likes Residual Attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 929–943.

[23] W. Herzallah, H. Faris, and O. Adwan. 2018. Feature engineering for detecting spammers on Twitter: Modelling and analysis. *Journal of Information Science* 44, 2 (2018), 230–247.

[24] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[25] Mücahit Kantepe and Murat Can Ganiz. 2017. Preprocessing framework for Twitter bot detection. In *2017 International conference on computer science and engineering (ubmk)*. IEEE, 630–634.

[26] Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences* 467 (2018), 312–322.

[27] Zhenyu Lei, Herun Wan, Wenqian Zhang, Shangbin Feng, Zilong Chen, Qinghua Zheng, and Minnan Luo. 2022. BIC: Twitter Bot Detection with Text-Graph Interaction and Semantic Consistency. *arXiv preprint arXiv:2208.08320* (2022).

[28] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[29] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. 2022. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*. PMLR, 13242–13256.

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. [n. d.]. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ([n. d.]).

[31] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 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.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[33] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. [n. d.]. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*.

[34] P. Pham, Ltt Nguyen, B. Vo, and U. Yun. 2021. Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. *Information Systems* 5 (2021), 101771.

[35] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. 2022. H2-FDetector: a GNN-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM Web Conference 2022*. 1486–1494.

[36] Shuhao Shi, Kai Qiao, Jian Chen, Shuai Yang, Jie Yang, Baojie Song, Linyuan Wang, and Bin Yan. 2023. MGTAB: A Multi-Relational Graph-Based Twitter Account Detection Benchmark. (2023).

[37] Shuhao Shi, Kai Qiao, Jie Yang, Baojie Song, Jian Chen, and Bin Yan. 2023. Over-Sampling Strategy in Feature Space for Graphs based Class-imbalanced Bot Detection. *arXiv e-prints* (2023), arXiv–2302.

[38] Onur Varol, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. 2018. Feature engineering for social bot detection. *Feature engineering for machine learning and data analytics* 311 (2018).

[39] Feng Wei and Uyen Trang Nguyen. 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)*. IEEE, 101–109.

[40] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[41] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2022. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1287–1292.

[42] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1096–1103.

[43] Yingguang Yang, Renyu Yang, Yangyang Li, Kai Cui, Zhiqin Yang, Yue Wang, Jie Xu, and Haiyong Xie. 2022. RoSGAS: Adaptive Social Bot Detection with Reinforced Self-Supervised GNN Architecture Search. *ACM Transactions on the Web* (2022).

[44] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph neural networks for graphs with heterophily: A survey. (2022).

[45] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems* 33 (2020), 7793–7804.