Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.056930

ARTICLE

Unmasking Social Robots' Camouflage: A GNN-Random Forest Framework for Enhanced Detection

Weijian Fan[1,*], Chunhua Wang[2], Xiao Han[3] and Chichen Lin[4]

[1]School of Data Science and Intelligent Media, Communication University of China, Beijing, 100024, China

[2]School of Computer and Cyber Sciences, Communication University of China, Beijing, 100024, China

[3]Institute of Communication Studies, Communication University of China, Beijing, 100024, China

[4]State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, 100024, China

*Corresponding Author: Weijian Fan. Email: wjfan@cuc.edu.cn

Received: 02 August 2024 Accepted: 12 October 2024 Published: 03 January 2025

ABSTRACT

The proliferation of robot accounts on social media platforms has posed a significant negative impact, necessitating robust measures to counter network anomalies and safeguard content integrity. Social robot detection has emerged as a pivotal yet intricate task, aimed at mitigating the dissemination of misleading information. While graph-based approaches have attained remarkable performance in this realm, they grapple with a fundamental limitation: the homogeneity assumption in graph convolution allows social robots to stealthily evade detection by mingling with genuine human profiles. To unravel this challenge and thwart the camouflage tactics, this work proposed an innovative social robot detection framework based on enhanced HOmogeneity and Random Forest (HORFBot). At the core of HORFBot lies a homogeneous graph enhancement strategy, intricately woven with edge-removal techniques, to meticulously dissect the graph into multiple revealing subgraphs. Subsequently, leveraging the power of contrastive learning, the proposed methodology meticulously trains multiple graph convolutional networks, each honed to discern nuances within these tailored subgraphs. The culminating stage involves the fusion of these feature-rich base classifiers, harmoniously aggregating their insights to produce a comprehensive detection outcome. Extensive experiments on three social robot detection datasets have shown that this method effectively improves the accuracy of social robot detection and outperforms comparative methods.

KEYWORDS

Social robot detection; graph neural networks; random forest; homophily; heterophily

1 Introduction

Social media platforms, such as Twitter (now known as X), Facebook, and Sina Weibo, have become an indispensable part of people's daily lives for sharing information and communicating with each other. However, these platforms are being undermined by automated programs called social bots. The presence of social bots poses numerous problems for social media. They are often used to disseminate false information [1], manipulate elections [2], promote conspiracy theories [3], and cause significant negative social impacts [4–6]. Detecting bots on social media is crucial for ensuring the

stability of the platform and safeguarding user interest. Therefore, a critical demand exists for precise and effective techniques to detect social bots and counteract their detrimental impacts.

Researchers have proposed a large number of social bot detection methods, which can be broadly categorized into three categories: feature-based, text-based, and graph-based. The feature-based methods involve extracting features from user metadata [7], temporal features [8], and tweets [9,10]. These features are then input into various classifiers, such as Random Forest (RF) or deep neural networks, to detect social bots. However, social bots can evade detection by simulating real users through sophisticated strategies [5]. Moreover, such methods rarely consider the diverse relationships between social accounts (such as follows, comments, etc.), making it challenging to guarantee detection accuracy [11]. Text-based methods often utilize NLP techniques such as word embeddings [12], recurrent neural networks [13], and pretrained language models [14,15] to encode tweet content and identify malicious intent. However, text-based methods cannot effectively identify social bots that mix malicious tweets with regular tweets [16].

Graph neural networks, through their efficient capture of semantic relationship features by leveraging the relationships between nodes in the graph structure, have been pivotal in the advancement of social bot detection. These methods, treating social bot detection as a node classification problem, consider social media platform accounts as nodes, the relationships between accounts as edges, and then employ graph neural networks (such as GCN [17,18], RGCN [9], RGT [10]) to learn user node representations for bot detection. This practical application of graph neural networks has led to the achievement of state-of-the-art performance in social bot detection, effectively detecting new social bots with better generalization [9,10].

Although graph-based social bot detection methods have made significant progress, they ignore the negative impact of heterogeneity (i.e., connections between different types of accounts) due to their homogeneity assumption. Existing graph-based social bot detection methods use a low-pass filter to smooth the user features within neighborhoods, i.e., using the sum of neighboring representations as the representation of the central node. This allows social bots to evade detection by engaging in inter-action behaviors with real accounts [19]. Fig. 1 illustrates the camouflage behavior of a social bot, which establishes connections with many real users. Its suspicious features are smoothed and weakened by its genuine account neighbors, thus evading detection by homogenous graph-based detection algorithms.

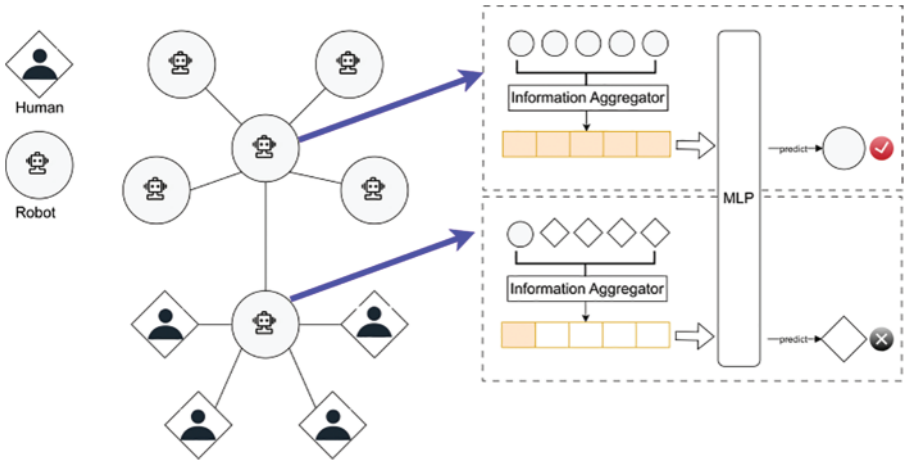

Figure 1: Social bot evades detection by camouflage behavior

This article proposes a social robot detection framework based on enhanced homogeneity and random forest (HORFBot). Specifically, this article first designs a graph homogeneity enhancement method that includes node enhancement and topo-logical structure enhancement. By combining with existing edge perturbation enhancement methods, different subgraphs are obtained. Then, using contrastive learning in a cross-view manner, we obtain node representations that are class-consistent between different subgraphs. Finally, to effectively harness the benefits of ensemble learning and GNN for encoding node relationships, we employ GNN as the foundational classifier within the RF, while utilizing the selected features as input for the fully connected neural network. The outputs from the GNN base classifier are then aligned through a straightforward fusion mechanism. Our experimental results demonstrate that HORFBot attains exceptional performance on three benchmark datasets for social robot detection, surpassing existing methods significantly. Furthermore, numerous experiments validate the efficacy of the graph enhancement technique proposed in this article and the enhancement in the performance of GNN base classifiers when utilizing RF as a traditional ensemble learning algorithm. The main contributions of this work are as follows:

a) Introducing a graph augmentation method that enhances information homogeneity and aggregates both low-frequency and high-frequency information under the supervision of contrastive learning, ultimately facilitating improved learning of node representations.
b) Effectively integrating the random forest algorithm with GNN, harnessing GNN's ability in encoding relationships and capitalizing on RF's strengths to enhance model performance and robustness.
c) Presenting a straightforward fusion mechanism that leverages the remaining features post-selection, effectively utilizing features not utilized by the GNN-based classifier.
d) Performing experiments on three benchmark datasets for social bot detection. The outcomes demonstrate that HORFBot consistently surpasses previous state-of-the-art methodologies.

2 Related Work

2.1 Feature-Based Social Bot Detection

Discriminative features are designed from user metadata [7], temporal features [8], tweets [9,10], and follower relationships [20] through the process of feature engineering. Subsequently, these features are fed into traditional classifiers, including support vector machines, naive Bayes, and random forests [6], to identify social bot accounts. Mazza developed a system named BotOrNot, which utilizes RF for evaluating and detecting social bots [8]. Fernquist et al. conducted a study examining the impact of political bots on the 2018 Swedish election [21]. The study assessed several algorithms, such as AdaBoost, support vector machines, and RF, and discovered that RF outperformed the other algorithms, achieving an accuracy of 0.957. Cresci evaluated multiple classifiers, such as RF, AdaBoost, support vector machines, and K-Nearest Neighbors, for the purpose of detecting bots [16]. Upon analyzing the performance of these classifiers, it was found that the use of radio frequency mixing provided the most optimal results. Although the RF demonstrates effectiveness in bot detection, new camouflage and adversarial techniques continue to evolve, posing a persistent threat and evading detection [22].

However, the detectability of these features can be easily mimicked and evaded by social bots. Over time, social bot manipulators can deliberately disguise themselves by creating bot features [16], rendering feature-based detection methods ineffective.

2.2 Graph-Based Social Bot Detection

Alhosseini et al. utilize account and relationship graphs as graph structural features and apply graph convolutional networks (GCN) to social robot detection for the first time [17]. Satar model utilizes GCNs in a feature engineering manner and utilizes self-supervision for social robot detection [20]. BotRGCN [9] utilizes relational graph convolutional networks within an information aggregation mechanism, considering different relationships. Building upon this work, Feng et al. add relationship categories and applies graph transformers to better aggregate information from neighbors [10]. Yang et al. propose the RoSGAS framework for social robot detection, which utilizes heterogeneous information networks to effectively model multiple nodes and relationships in social networks and introduces reinforcement learning to acquire subgraph embeddings [23]. Most existing methods are based on the homogeneity assumption, where nodes in the graph tend to interact more with similar nodes. For example, RSGCN considers the similarity of features among similar nodes, applies weighted feature propagation, and ensures effective information aggregation by assigning higher weights to similar neighboring nodes. It can aggregate information from distant neighboring node features without excessive smoothing and gradient vanishing [24]. However, heterogeneity is widespread in social media, and social robots interact with real users. Existing graph-based methods that assume homogeneity significantly degrade their detection performance.

To address the challenge of heterogeneity, researchers have proposed various methods. These can be roughly divided into two categories: non-local neighbor information aggregation and adaptive message passing. GeMo-GNN [25] addresses heterogeneity by precomputing unsupervised node embedding and defining a two-level aggregation process. LINKX [26] embeds both the adjacent matrix and node features separately and combines them using a simple MLP. H2GCN [27] addresses heterogeneity with three key design choices: self-neighbor feature separation, high-order neighbor aggregation, and a combination of intermediate representations. CPGNN [28] annotates relevance through a compatibility matrix and propagates prior confidence estimation through this matrix to learn discriminative representations of heterogeneous graphs. However, in the social robot detection task, social robots exhibit higher heterogeneity, while real users exhibit higher homogeneity. Therefore, directly applying heterogeneous GNNs to social robot detection may not produce optimal results.

2.3 Graph Contrastive Learning

Supervised contrastive learning is a widely used strategy in deep learning that involves learning by comparing the differences in labels of various instances. This approach enables the model to effectively handle noisy, imbalanced, or complex data. Graph contrastive learning (GCL) extends this technique to the realm of graph data, with the primary objective of learning the inherent structure and patterns by comparing different representations or views of graph data. GRACE [29] generates views by applying edge removal and feature masking and considers the same nodes in different views as positive pairs. DGI [30], on the other hand, learns node representations by maximizing the mutual information between local and global embeddings. However, self-supervised graph contrastive learning methods face the challenge of class inconsistency, where representations of samples from the same class may be distant, while representations of different classes may be close [31]. Originally applied in computer vision, supervised contrastive learning treats intra-class images as positive pairs to achieve representations that are closer to the same class than those from different classes [32]. This paper leverages this property to train an encoder capable of adapting to both homogeneous and heterogeneous edges using a supervised contrastive loss to aggregate effective information.

3 Methodology

3.1 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, A)$ denotes a social network, where $\mathcal{V} = \{v_i|1, 2, \ldots, N\}$ is the set of all users; $\mathcal{E} = \{\mathcal{E}_e = e \in 1, 2, \ldots, E\}$ denotes the collection of edges without self-loops formed by different relations $\mathcal{R}$; $A \in \mathbb{R}^{N*N}$ is the symmetric adjacency matrix. The features of user v_i are represented as $X \in \mathbb{R}^{N*NF}$, where NF is the dimension of user node features. The social bot detection task is to use the network graph $\mathcal{G}$ and the users' labels Y_{train} to predict the user label $\hat{Y}_{test}$ by model f. The detection process can be formalized as Eq. (1).

$$f\left(\mathcal{G}, Y_{train}\right) \rightarrow \hat{Y}_{test}. \tag{1}$$

To gain a deeper understanding of homophily and heterophily in the task of detecting social bots, this paper employs the class-insensitive homophily metric [26] for calculation. The homophily and heterophily values are determined using Eqs. (2) and (3), respectively.

$$ho = \frac{1}{\mathcal{R}-1} \sum_{r=1}^{\mathcal{R}} \max\left(0, ho_r - \frac{|\mathcal{R}_r|}{|\mathcal{V}|}\right), \tag{2}$$

$$he = 1 - ho, \tag{3}$$

where $\mathcal{R}$ is the number of relations; $\mathcal{R}_r$ is the number of users of class r; ho_r represents the relation homophily of users of class r.

3.2 Graph Augmentor

To deal with the camouflage behavior of social robots, this paper proposes a simple method of graph enhancement to improve homogeneity in heterogeneous graphs. This method mainly consists of three steps. Firstly, this paper trains a two-layer multilayer perceptron (MLP) classification network using the train dataset and takes the output of the last layer as user representation.

$$\hat{y}_c = soft\max\left(w_2 * \mathrm{Re}LU\left(w_1 \mathrm{X} + \mathrm{b}_1\right) + \mathrm{b}_2\right), \tag{4}$$

$$mlp_v = w_1 \mathrm{X} + \mathrm{b}_1, \tag{5}$$

where $w_1, w_2, \mathrm{b}_1, \mathrm{b}_2$ are learnable parameters. Since MLP does not use the original graph structure, the user representation based on MLP is still not affected by the heterogeneity of the original graph.

Then, the cosine similarity is used to calculate the k most similar users $simK_i$ for a certain user v_i.

$$simK_i = TopK\left(\cos\left(mlp_{v_i}, mlp_{v_j}\right)\right), v_i, v_j \in \mathcal{V}_{train}; i \neq j, \tag{6}$$

where $\cos(\cdot, \cdot)$ is the cosine similarity between two different users' representation.

Finally, the paper performs a two-step enhancement on the original graph: first, for each user node, applied randomly replace with one of the same-category users in the similar nodes $simK_i$; second, merged the subgraph composed of the k most similar users $simK_i$ without the replaced node with the original graph.

This graph augmenter cannot only effectively improve the homogeneity of graphs in social networks and reduce the impact of social robot interaction behaviors but also enable GNN to learn neighborhood invariant features through the exchange of Rene similarity features. However, it should be noted that this method does not eliminate the camouflage behavior of social robots and may introduce redundant edges during the introduction of similar nodes. To mitigate these negative effects, this paper introduces a contrastive learning framework into the GNN training process.

3.3 Node Representation

Following the traditional graph contrastive learning framework, based on the graph augmentor in Section 3.2, this paper uses edge removing method to construct two different subgraph views. The specific process is described as follows:

Unlike using a two-layer MLP classification network for user node similarity, this section uses the Transformer encoder to represent node features. As shown in Eq. (7), different types of feature vectors are treated as different tokens.

$$tr_i^0 = Transformer\left(\left[x_i^1, x_i^2, \ldots, x_i^t\right]\right), \tag{7}$$

where x_i^t references the feature of user node v_i with type t; tr_i^0 is Transformer encoder output.

Then, as shown in Eq. (8), applying an MLP to tr_i^0 as the GNN input.

$$h_i^{(0)} = \mathrm{Re}LU\left(w_3 * tr_i^0 + b_3\right). \tag{8}$$

Next, inspired by the previous work [33] on graphs with heterophily, the paper uses high-pass filters to capture differentiating neighbor features in the spatial domain and uses low-pass filtering to aggregate information from neighboring nodes. This can be formulated as Eqs. (9) and (10).

$$\left(h_i^{(l)}\right)_{Low} = W^{\{l-1\}} \sum \left(h_i^{\{l-1\}} + h_j^{\{l-1\}}\right), \tag{9}$$

$$\left(h_i^{(l)}\right)_{High} = W^{\{l-1\}} \sum \left(h_i^{\{l-1\}} - h_j^{\{l-1\}}\right), \tag{10}$$

where h_j is one of the neighbors of user v_i representation.

To use the contrastive learning framework, this paper uses a projection head consisting of another MLP layer to obtain z_i as Eq. (11).

$$z_i = w_5 * \mathrm{Re}LU\left(w_4 * \left[h_i^{(L-1)}; \left(h_i^{(L)}\right)_{Low} \oplus \left(h_i^{(L)}\right)_{High}\right] + \mathrm{b}_4\right) + \mathrm{b}_5, \tag{11}$$

where $w_4, w_5, \mathrm{b}_4, \mathrm{b}_5$ are learnable parameters; L is the last layer of GNN; $[\cdot;\cdot]$ represents concat operation.

Thus, following the above process, user v_i in two subgraph views can be calculated as $z_i^{\mathcal{G}_1}$ and $z_i^{\mathcal{G}_2}$.

Finally, to avoid overfitting, this paper uses cross-subgraph-view contrastive learning. The same-class nodes in two subgraphs are positive pairs. The different class nodes are negative pairs. The final loss for all user nodes can be calculated as Eq. (12).

$$\mathcal{L} = \frac{1}{2\mathcal{V}} * \sum_{i=1}^{\mathcal{V}} \left(\left(-\frac{1}{\mathcal{V}_{y_i}} \sum_{j=1}^{\mathcal{V}} 1_{y_i=y_j} \cdot \log \frac{e^{\cos\left(z_i^{\mathcal{G}_1}, z_j^{\mathcal{G}_2}\right)/\tau}}{\sum_{k=1}^{\mathcal{V}} e^{\cos\left(z_i^{\mathcal{G}_1}, z_k^{\mathcal{G}_2}\right)/\tau}} \right) + \left(-\frac{1}{\mathcal{V}_{y_i}} \sum_{j=1}^{\mathcal{V}} 1_{y_i=y_j} \cdot \log \frac{e^{\cos\left(z_i^{\mathcal{G}_1}, z_j^{\mathcal{G}_2}\right)/\tau}}{\sum_{k=1}^{\mathcal{V}} e^{\cos\left(z_i^{\mathcal{G}_1}, z_k^{\mathcal{G}_2}\right)/\tau}} \right) \right), \tag{12}$$

where $\mathcal{V}_{y_i}$ references the number of user v_i with same class; τ is the temperature coefficient.

3.4 HORFBot

Based on the proposed graph augmentor and node representation based on contrastive learning, this paper combines the random forest algorithm to propose the social robot detection framework HORFBot. Fig. 2 shows the overall structure of HORFBot, including the base classifier construction module, feature alignment module, and base classifier integration module.

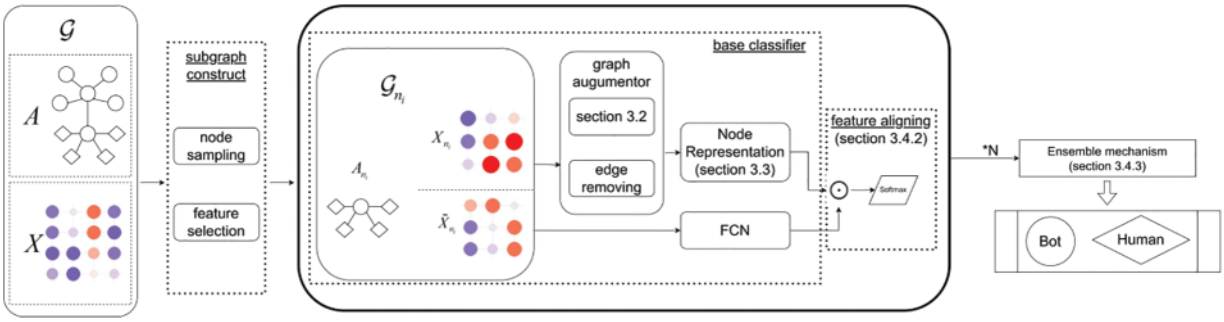

Figure 2: Overall structure of HORFBot

3.4.1 Base Classifier

Constructing subgraphs is to obtain subgraphs as the training data for base classifiers. This paper uses two methods, node sampling and feature selection, to ensure the diversity of the sustaining sets.

Due to the characteristics of social graph data, this paper does not use the RF algorithm itself to randomly select nodes for replacement in the node sampling process. Instead, while preserving the nodes, they also select the edges connected to them. The probability of preserving nodes α follows an *i.i.d.* uniform distribution. In feature selection, this work randomly selects a β proportion of the feature vector from the subgraph $\mathcal{G}_{n_i}$, named X_{n_i}, which is used as the input feature matrix of the i-th GNN base classifier. The GNN basic classifier has been described in detail in Section 3.3.

3.4.2 Aligning Mechanism

As show in Fig. 2, the i-th branch utilizes only a subset of the feature dimensions X_{n_i} for training the base classifier $\mathcal{G}_{n_i}$. The remaining features $\widetilde{X}_{n_i}$, which are selected through feature selection, are employed to train a fully connected neural network (FCN) F_{n_i}. The outputs of the GNN-based classifier and the FCN are aligned using the Hadamard product, as described in Eq. (13).

$$output_{n_i} = \mathcal{G}_{n_i}\left(X_{n_i}\right) \odot F_{n_i}\left(\widetilde{X}_{n_i}\right), \tag{13}$$

where $\odot$ is the Hadamard product function.

Then, by passing the output embedding $output_{n_i}$ to a classification layer can be achieved the i-th brunch prediction result $\hat{\mathrm{Y}}_{n_i}$.

$$\hat{\mathrm{Y}}_{n_i} = soft\max\left(w_6 * output_{n_i} + b_6\right), \tag{14}$$

where w_6, b_6 are learnable parameters.

3.4.3 Ensemble Mechanism

The base classifiers can be trained simultaneously. Once the base GNN classifiers are finished in the training stage. Their outputs can be combined to produce the ultimate classification result, as shown in Eq. (15).

$$\hat{Y} = \sum_{i=1}^{N} \hat{Y}_{n_i}. \tag{15}$$

4 Experiment and Parameter Setup

4.1 Dataset

In this study, we evaluate social bot detection models on three graph-structured datasets: Cresci-15 [4], TwitBot-20 [34] and TwitBot-22 [35]. Here is a detailed description of these datasets:

a) The Cresci-15 dataset is a collection of real and fake followers on Twitter from 01 May to 31 July 2015. It contains 5301 users and provides metadata for each account, including the number of followers, friends, and the account's registration time, as well as textual information for the accounts.
b) The TwitBot-20 dataset is a large-scale benchmark for Twitter bot detection that includes 229,573 users (11,826 accounts are labeled), 33,488,192 tweets, 8,723,736 user attributes, and 455,958 follower relations. This dataset covers a diverse range of bots and genuine users to better represent the real-world Twitter landscape.
c) The TwitBot-22 dataset collects four types of entities: users, tweets, lists, and hashtags. It also provides relationships between these entities, such as follows, posts, top tweets, likes, mentions, retweets, quotes, replies, ownerships, memberships, and contains. Furthermore, TwitBot-22 provides detailed information for each entity, such as user metadata and tweet content. This dataset provides 50,538 annotated users and involves different user types, such as genuine users, spammers, and commercial accounts. Compared to TwitBot-20, TwitBot-22 has a larger graph size with more nodes and edges.

Table 1 summarizes the statistical data of these datasets. Consistent with the Shi et al. method [11], this work uses a 10%, 10%, and 80% random split as the train, validation, and test sets for the experimental stage and report the average performance of five runs to ensure a fair comparison between HORFBot and baseline models.

Table 1: The statistic of Cresci-15, TwiBot-20 and TwiBot-22

Dataset	Number of nodes	Number of edges	Class	Each class number
Cresci-15 [4]	5301	14,220	Human	1950
			Bot	3351
TwiBot-20 [34]	229,580	227,979	Human	5237
			Bot	6589
TwiBot-22 [35]	11,000,000	3,743,634	Human	860,057
			Bot	139,943

4.2 Compare Models

To validate the effectiveness of HORFBot, this paper compares it with multiple social bot detection models and heterogeneous graph learning methods. Here are the detailed descriptions of these compare models:

a) SGBot [7] extracts feature from user metadata and utilizes a random forest classifier for scalable and generalizable bot detection.
b) RoBERTa [36] encodes user descriptions and tweets using a pretrained RoBERTa model, and feeds user characteristics into an MLP for bot identification.
c) LOBO [37] extracts feature from user metadata and tweets and uses a random forest to identify different bots.
d) GCN [18] represents spectral graph convolution methods. It simplifies Chebyshev polynomials to first-order neighborhoods, gathers features from neighbors to obtain node embedding vectors, and passes user representations to an MLP for classification.
e) SGC [38] is a streamlined version of the GCN, designed to simplify its intricate nature by systematically eliminating non-linearities between GCN layers and condensing the resultant function into a solitary linear transformation. This method guarantees comparable performance to GCNs while significantly decreasing the parameter size.
f) GAT [39] is a semisupervised graph model that incorporates attention mechanisms to determine the significance of adjacent users during aggregation. By assigning weights to various neighborhoods in an adaptive manner, it enhances the performance of graph neural networks. The representations obtained through learning are then passed to an MLP for classification.
g) BoostingGNN [40] is a graph ensemble learning technique that integrates GNNs with AdaBoost to enhance GNN performance in situations with imbalanced classes.
h) BotBuster [41] enhances cross-platform bot detection by processing user metadata and text information through a hybrid expert architecture.
i) RGT [10] models inherent heterogeneity in the Twittersphere using a relational graph transformer, enhancing social bot detection.
j) BIC [42] proposes a text-graph interaction module and models semantic consistency to improve bot detection performance and defend against evolving bots.
k) BotRGCN [9] establishes a heterogeneous social network and employs relational graph convolutional networks to learn user representations and detect social bots.
l) LINKX [26] is a simple and scalable method that separately embeds node features and adjacency matrices and feeds them into an MLP for heterogeneous graph representation learning.
m) H2GCN [27] embeds self- and higher-order neighborhood features, performing well in heterogeneous benchmark tests.

4.3 Variant Models

For a comprehensive understanding of the operational mode of HORFBot and a thorough evaluation of each module's contribution to performance enhancement, this work developed several variant models based on the three improved components of HORFBot. These components include graph augmentation methods, node representations based on contrastive learning, and alignment mechanisms. For ablation studies, we selectively enable or disable these components. The following are detailed descriptions of these variant models:

a) HORFBot+w/o HO: This variant does not use the graph augmentation method proposed in this paper. To ensure the integrity of the entire framework, we adopt the node sampling enhancement method as a replacement.
b) HORFBot+w/o CL: This variant does not use contrastive learning to complete node vectors and replaces it with RGAT.
c) HORFBot+w/o AM: This variant does not use the alignment mechanism, and each base classifier only retains the selected features.
d) HORFBot+w/o RF: This variant does not use the random forest ensemble algorithm and only uses a single base classifier.

4.4 Implementation Details

This paper implements all baseline models using PyTorch, PyTorch geometry, Scikit-learn, and PyGCL. Only edge deletion is used to enhance the original graph. The number of attention heads in GAT is set to 4. The hyperparameter configuration of HORFBot proposed in this paper is shown in Table 2. Different learning rates and batch sizes are set according to the size of the dataset. In addition, dropout mechanisms are adopted to prevent overfitting. For MLPs, the dropout rate is set to 0.5; for generated edge embeddings, the dropout rate is set to 0.3. Since a smaller temperature coefficient in contrastive learning means more attention to difficult samples, the temperature coefficient is set to 0.07 in this paper.

Table 2: Hyperparameter setting on Cresci-15, TwiBot-20 and TwiBot-22

Parameter	Cresci-15	TwiBot-20	TwiBot-22
Optimizer	Adam	Adam	Adam
Learning rate	0.01	0.001	0.0005
Batch	128	128	512
Dropout	0.5	0.5	0.5
Temperature	0.07	0.07	0.07
Epochs	100	200	60

4.5 Evaluation Metrics

Due to the class imbalance problem in social bot detection tasks, both in real-world scenarios and in benchmark datasets, we evaluate the performance of the model's using accuracy and F1-score. The specific calculation formulas are as Eqs. (16) and (17).

$$\text{Acc} = \frac{TP + TN}{TP + FP + FN + TN}, \tag{16}$$

$$\text{F1} = \frac{2 * \frac{TP}{TP + FP} * \frac{TP}{TP + FN}}{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}}, \tag{17}$$

where TP is True Positive; TN is True Negative; FP is False Positive; and FN is False Negative.

5 Results and Discussion

In this section, we conduct experiments on the three datasets to evaluate the effectiveness of HORFBot in social robot detection. Specifically, this paper proposes the following four research questions (RQs) to guide the experiment:

a) RQ1: Does heterogeneity affect the performance of social robot detection models (Section 5.1)?
b) RQ2: Is HORFBot superior to state-of-the-art methods in the detection task of social robots (Section 5.2)?
c) RQ3: Does each module in HORFBot effectively improve the overall performance (Section 5.3)?
d) RQ4: Is the performance of HORFBot impacted by the quantity of base classifiers used (Section 5.4)?

5.1 Heterogeneity Tendency (RQ1)

Before conducting the overall experiments, this paper analyzes the levels of homogeneity and heterogeneity of the three datasets. As shown in Table 3, compared with the TwiBot-20 and TwiBot-22 datasets, the homogeneity score of the Cresci-15 dataset is 0.99, significantly higher than the other two datasets. This indicates that in the early stage of social robot development, there is still a clear social circle division between robot accounts and real user accounts, and there is no interaction between them.

Table 3: Homogeneity score on Cresci-15, TwiBot-20 and TwiBot-22

Dataset	Node class	Relation type	Homogeneity
Cresci-15	All	All	0.99
TwiBot-20	Human	Follower	0.82
		Following	0.34
	Bot	Follower	0.29
		Following	0.75
TwiBot-22	Human	Follower	0.88
		Following	0.96
	Bot	Follower	0.17
		Following	0.06

In the TwiBot-20 dataset, compared with real users, social robots show obvious heterogeneity tendencies in follower relationships, while in TwiBot-20's following relationships, they show stronger homogeneity tendencies. In the TwiBot-22 dataset, social robots show extremely high heterogeneity in both follower and following relationships. This is because the number of social robot accounts in this dataset is far less than that of real human users. At the same time, compared with Cresci-15, social robot accounts involved in the relationship account type are more real human users, rather than similar robot accounts. This further indicates that with the development of technology, social robots are increasingly interacting with humans.

In Table 4, the work of various researchers on social robot detection at different time stages is presented. It is evident that as the heterogeneity of the dataset increases, the performance of the detection model decreases to varying degrees. This also indicates the importance of studying heterogeneous camouflage behavior.

Table 4: Accuracy on Cresci-15, TwiBot-20 and TwiBot-22

	Cresci-15	TwiBot-20	TwiBot-20
SGBot [7]	96.30	79.93	78.44
BIC [42]	96.13	75.78	73.78
BotRGCN [9]	96.56	85.75	79.58

5.2 Performance Comparison (RQ2)

Table 5 summarizes the detection results of 13 baseline methods and the proposed HORFBot on Cresci-15, TwiBot-20, and TwiBot-22. The symbol "–" indicates out-of-memory on the experimental machine. HORFBot outperforms the other 13 baseline models on all three datasets.

Table 5: Performance comparison on Cresci-15, TwiBot-20 and TwiBot-22

Model	Cresci-15		TwiBot-20		TwiBot-22	
	Acc	F1	Acc	F1	Acc	F1
SGBot [7]	96.30	96.18	79.93	83.89	78.44	54.97
RoBERTa [36]	96.41	92.22	72.46	76.72	78.30	50.57
LOBO [37]	96.01	96.32	76.18	80.45	79.05	45.27
GCN [18]	95.19	94.78	77.53	79.96	78.29	54.86
SGC [38]	95.69	94.69	68.01	67.73	71.02	44.19
GAT [39]	96.10	95.29	83.05	85.12	79.58	54.96
BoostingGNN [40]	95.69	94.90	68.80	68.36	79.92	45.17
BotBuster [41]	96.68	96.52	79.34	82.47	79.82	46.07
RGT [10]	96.63	98.24	86.57	87.81	76.04	42.54
BIC [42]	96.13	96.23	75.78	79.24	73.78	32,06
BotRGCN [9]	96.56	96.97	85.75	86.65	79.58	56.50
LINKX [26]	91.65	93.27	76.22	80.35	78.03	54.37
H2GCN [27]	95.15	96.03	78.17	80.17	–	–
HORFBot	97.32	98.04	87.06	88.49	81.99	60.53

The Cresci-15 dataset does not interact much between social bots and real users. Therefore, most detection methods achieve an accuracy rate of over 95%, and there is little room for improvement on the Cresci-15 dataset. However, the HORFBot based on the RF framework still has room for improvement, indicating that the overall algorithm framework with GNN as the base classifier is effective for social bot detection tasks. With the increase of graph heterogeneity, the advantage of HORFBot over the baseline will also increase. On the TwiBot-22 and TwiBot-20 datasets, compared with the current best results, the proposed method can improve accuracy by 2.07 and 0.49, respectively. These results indicate that HORFBot has higher effectiveness in addressing social bot camouflage behavior. This excellent performance also demonstrates the importance of considering both homogeneity and heterogeneity modeling for detecting social bots with interactive behavior, and it also verifies the effectiveness of the random forest algorithm framework with GNNs as the base classifier.

Among all the comparison models, SGBot and LOBO models, like this article, both used the random forest algorithm. But they only consider user metadata and tweets as features. In contrast, the graph-based classifier construction method proposed in this article performs better in social robot detection tasks, with improvements of 4.5% and 3.7% on the TwiBot-22 dataset, respectively. BotBuster, LINKX, and H2GCN have been optimized from the perspectives of cross platform and heterogeneity, and their performance on TwiBot-22 is superior to TwiBot-20, indicating their effectiveness in addressing the heterogeneous camouflage behavior of current social robots. Compared to these three models, HORFBot performs better in detecting heterogeneous camouflage behavior, achieving an accuracy of 81.99% and a recall rate of 60.53% on the TwiBot-22 dataset.

5.3 Ablation Study (RQ3)

To answer RQ3, this paper conducted an ablation study to investigate the impact of different module designs on the performance of HORFBot. To achieve this goal, four ablation models were constructed as described in Section 4.3: w/o HO, w/o CL, w/o AM, and w/o RF. The experimental results of these ablation models on Cresci-15, TwiBot-20, and TwiBot-22 are shown in Table 6.

Table 6: Ablation study on Cresci-15, TwiBot-20 and TwiBot-22

Variant	Cresci-15		TwiBot-20		TwiBot-22	
	Acc	F1	Acc	F1	Acc	F1
w/o HO	96.42	96.23	78.02	80.91	74.07	73.16
w/o CL	96.45	96.18	81.69	81.34	80.13	47.39
w/o AM	96.40	96.11	82.21	81.87	81.99	60.44
w/o RF	95.23	94.92	71.41	71.19	72.82	72.04
HORFBot	97.32	98.04	87.06	88.49	81.99	60.53

On the Cresci-15 dataset, each ablation model achieves over 95% accuracy rate since there is no camouflage behavior in the social robot accounts in this dataset. This result also indicates that this dataset fails to truly represent the behavior characteristics of social robots in real environments. On the TwiBot-20 and TwiBot-22 datasets, compared with HORFBot, w/o HO and w/o RF show significant performance degradation in model performance. This suggests that the proposed graph enhancement method can effectively aggregate information of the same or different categories during message passing. The w/o RF model demonstrates that training base classifiers on different feature spaces can increase model diversity. Combining the predictions of multiple base classifiers can significantly improve the overall model performance. Additionally, w/o AM only uses a subset of features, sometimes leading to performance degradation. Although the w/o CL model has less performance degradation, it is still evident that the contrastive learning strategy is beneficial for improving model performance.

Through comparisons between these ablation models and the complete model, this paper demonstrates the proposed HORFBot's effectiveness in terms of each submodule's design choices.

5.4 Study on Base Classifiers Quantity (RQ4)

The importance of the number of base classifiers in a random forest cannot be ignored, as it directly affects the model's performance, generalization ability, and computational complexity. In

this work, the core concept of constructing HORFBot is derived from the random forest algorithm. Therefore, the number of base classifiers S is also a key factor affecting the performance of HORFBot. This section focuses on the impact of the number of base classifiers on the performance of HORFBot and conducts experiments on three datasets to further determine the optimal setting of the number of base classifiers.

As shown in Fig. 3, for all three datasets, when the number of base classifiers is small, the classification accuracy of HORFBot is very low, and the performance is poor. With the increase in base classifiers, its accuracy has significantly improved. On the TwiBot-22 dataset, when N increases from 2 to 10, the accuracy significantly improves. On the TwiBot-20 and Cresci-15 datasets, when it is less than 6, the accuracy rate shows an upward trend. On all datasets, when N increases to 10, the classification accuracy of HORFBot no longer increases and starts to fluctuate. This may be because when the number of base classifiers is too large, the complexity of the model increases, and the risk of overfitting increases. Increasing the number of base classifiers also increases computational complexity and memory requirements, leading to decreased model efficiency. Therefore, in this paper, the number of base classifiers on Cresci-15, TwiBot-20, and TwiBot-22 is set to 11, 8, and 10, respectively.

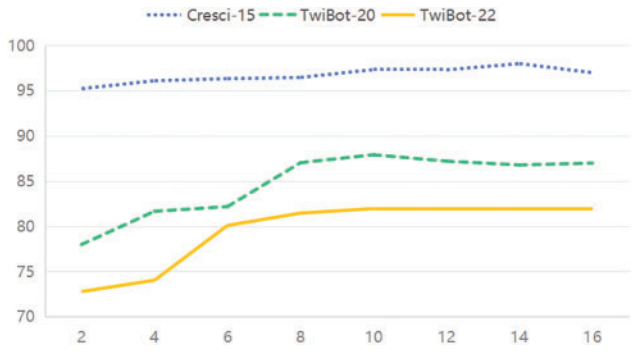

Figure 3: HORFBot with different numbers of base classifiers on three datasets

6 Conclusion

Detecting social bots is a vital and complex task that is crucial for protecting user interests and ensuring the stable functioning of social media platforms. This paper finds that compared with earlier social bots, modern bots tend to engage in interactive behavior with real users to evade detection. To address this issue, this paper proposes a new social bot detection framework called HORFBot, which includes a graph enhancement module targeting camouflage behavior and uses GNN as the base classifier for RF. The framework constructs subgraphs, selects features, and aligns different subgraphs to train base classifiers. Then, it integrates the whole branch's results. Experiments show that HORFBot achieves state-of-the-art performance on three social bot detection benchmarks. Other studies further demonstrate the effectiveness of the proposed graph enhancement module and the combination of RF and showcase HORFBot's ability to identify camouflage interactive behavior in social bot detection. HORFBot has the potential to serve as a pivotal tool for bolstering social media security and enhancing user protection in the future. Its capabilities can aid in mitigating the dissemination of misinformation and fostering a more credible online environment.

However, there are some limitations of the HORFBot model: firstly, in the training stage, this paper adopts a three-stage training mode, which may lead to suboptimal performance of the model. Secondly, due to limited data sources, this paper can only detect social robot accounts on Twitter.

Future research directions include extending to other social platforms and building an end-to-end training framework.

Acknowledgement: All authors sincerely thank all organizations and institutions that have provided data and resources.

Funding Statement: This work has been supported by the Fundamental Research Funds for the Central Universities (grant number CUC24SG018).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Weijian Fan; data collection: Weijian Fan, Chunhua Wang and Chichen Lin; analysis and interpretation of results: Weijian Fan and Xiao Han; draft manuscript preparation: Weijian Fan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in ResearchGate and GitHub at https://www.researchgate.net/figure/Details-of-the-Cresci-2015-dataset_tbl2_370918138 (accessed on 11 October 2024), https://github.com/BunsenFeng/TwiBot-20 and https://github.com/LuoUndergradXJTU/TwiBot-22 (accessed on 11 October 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] K. Starbird, "Disinformation's spread: Bots, trolls and all of us," *Nature*, vol. 571, no. 7766, pp. 449–450, Jul. 2019. doi: 10.1038/d41586-019-02235-x.

[2] S. Rossi, M. Rossi, B. R. Upreti, and Y. Liu, "Detecting political bots on Twitter during the 2019 Finnish parliamentary election," in *Annu. Hawaii Int. Conf. Syst. Sci.*, 2020, pp. 2430–2439.

[3] H. R. Greve, H. Rao, P. Vicinanza, and E. Y. Zhou, "Online conspiracy groups: Micro-bloggers, bots, and coronavirus conspiracy talk on Twitter," *Am. Sociol. Rev.*, vol. 87, no. 6, pp. 919–949, 2022. doi: 10.1177/00031224221125937.

[4] C. Stefano, Roberto, P. Marinella, S. Angelo, and T. Maurizio, "Fame for sale: Efficient detection of fake Twitter followers," *Decis. Support Syst.*, vol. 80, no. 8, pp. 56–71, 2015. doi: 10.1016/j.dss.2015.09.003.

[5] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *WWW'17 Companion: Proc. 26th Int. Conf. World Wide Web Companion*, Perth, WA, Australia, 2017, pp. 963–972.

[6] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96–104, 2016. doi: 10.1145/2818717.

[7] K. C. Yang, O. Varol, P. M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," *Proc. AAAI*, vol. 34, no. 1, pp. 1096–1103, 2020. doi: 10.1609/aaai.v34i01.5460.

[8] M. Mazza, S. Cresci, M. Avvenuti, W. Quattrociocchi, and M. Tesconi, "RTbust: Exploiting temporal patterns for botnet detection on Twitter," in *Proc. 10th ACM Conf. Web Sci.*, Boston, MA, USA, 2019, pp. 183–192.

[9] S. Feng, H. Wan, N. Wang, and M. Luo, "BotRGCN: Twitter bot detection with relational graph convolutional networks," in *Proc. 2021 IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min.*, Netherlands, 2022, pp. 236–239.

[10] S. Feng, Z. Tan, R. Li, and M. Luo, "Heterogeneity-aware twitter bot detection with relational graph transformers," *Proc. AAAI*, vol. 36, no. 4, pp. 3977–3985, 28 Jun. 2022. doi: 10.1609/aaai.v36i4.20314.

[11] S. Shi *et al.*, "MGTAB: A multi-relational graph-based twitter account detection benchmark," 2023. doi: 10.48550/arXiv.2301.01123.
[12] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *2019 First IEEE Int. Conf. TPS-ISA*, 2019, pp. 101–109. doi: 10.1109/TPS-ISA48467.2019.00021.
[13] J. Diaz, F. Bravo-Marquez, and B. Poblete, "Language modeling on location-based social networks," *ISPRS Int J. Geo-Inf.*, vol. 11, no. 2, 18 Feb. 2022, Art. no. 147. doi: 10.3390/ijgi11020147.
[14] D. Dukić, D. Keča, and D. Stipić, "Are you human? Detecting bots on twitter using BERT," in *2020 IEEE 7th Int. Conf. DSAA*, Sydney, NSW, Australia, 2020, pp. 631–636.
[15] M. Heidari and J. H. Jones, "Using BERT to extract topic-independent sentiment features for social media bot detection," in *11th IEEE Annu. UEMCON*, New York, NY, USA, 2020, pp. 542–547.
[16] S. Cresci, "A decade of social bot detection," *Commun. ACM*, vol. 63, no. 10, pp. 72–83, 2020. doi: 10.1145/3409116.
[17] S. A. Alhosseini, R. B. Tareaf, P. Najafi, and C. Meinel, "Detect me if you can: Spam bot detection using inductive representation learning," presented at WWW'19: Companion Proc. 2019 World Wide Web Conf., San Francisco, CA, USA, May 13–17, 2019, pp. 148–153.
[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016. doi: 10.48550/arXiv.1609.02907.
[19] T. Le, L. Tran-Thanh, and D. Lee, "Socialbots on fire: Modeling adversarial behaviors of socialbots via multi-agent hierarchical reinforcement learning," presented at WWW'22: Proc. ACM Web Conf. 2022, New York, NY, USA, 25 Apr. 2022, pp. 545–554.
[20] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "SATAR: A self-supervised approach to twitter account representation learning and its application in bot detection," in *CIKM'21: Proc. 30th ACM Int. Conf. Inform. Knowl. Manag.*, New York, NY, USA, 30 Oct. 2021, pp. 3808–3817.
[21] J. Fernquist, L. Kaati, and R. Schroeder, "Political bots and the Swedish general election," in *2018 IEEE Int. Conf. Intell. Secur. Inform. (ISI)*, Miami, FL, USA, 2018, pp. 124–129.
[22] N. G. des Mesnards , D. S. Hunter, Z. el Hjouji , and T. Zaman, "Detecting bots and assessing their impact in social networks," *Oper. Res.*, vol. 70, no. 1, pp. 1–22, 2022. doi: 10.1287/opre.2021.2118.
[23] Y. Yang *et al.*, "RoSGAS: Adaptive social bot detection with reinforced self-supervised GNN architecture search," *ACM Trans. Web.*, vol. 17, no. 3, pp. 1–13, 2023. doi: 10.1145/3572403.
[24] F. Wu, D. Li, K. Lin, and H. Zhang, "Efficient nodes representation learning with residual feature propagation," in *Advances in Knowledge Discovery and Data Mining*, Cham: Springer International Publishing, 2021, vol. 12713, pp. 156–167. doi: 10.1007/978-3-030-75765-6_13.
[25] H. Pei, B. Wei, K. C. -C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," 2020. doi: 10.48550/arXiv.2002.05287.
[26] D. Lim *et al.*, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," presented at NeurIPS'21, Red Hook, NY, USA, 2021, pp. 20887–20902.
[27] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," presented at NeurIPS'20, Red Hook, NY, USA, 2020, pp. 7793–7804.
[28] J. Zhu *et al.*, "Graph neural networks with heterophily," presented at Adv. Neural Inform. Process. Syst. (NeurIPS 2020), Red Hook, NY, USA, 2021, pp. 11168–11176.
[29] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu and L. Wang, "Deep graph contrastive representation learning," 2020. doi: 10.48550/arXiv.2006.04131.
[30] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio and R. D. Hjelm, "Deep graph infomax," 2018. doi: 10.48550/arXiv.1809.10341.
[31] M. Zheng *et al.*, "Weakly supervised contrastive learning," presented at 2021 IEEE/CVF Int. Conf. Comput. Vision (ICCV), Montreal, QC, Canada, 2021, pp. 10022–10031.
[32] P. Khosla *et al.*, "Supervised contrastive learning," presented at Adv. Neural Inform. Process. Syst. (NeurIPS 2020), Red Hook, NY, USA, 2020, pp. 18661–18673.

[33] S. Luan *et al.*, "Is heterophily a real nightmare for graph neural networks to do node classification?" 2021. doi: 10.48550/arXiv.2109.05641.
[34] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "TwiBot-20: A comprehensive twitter bot detection benchmark," presented at CIKM '21: Proc. 30th ACM Int. Conf. Inform. Knowl. Manag., New York, NY, USA, 2021, pp. 4485–4494.
[35] S. Feng *et al.*, "TwiBot-22: Towards graph-based twitter bot detection," presented at Adv. Neural Inform. Process. Syst. (NeurIPS 2022), Red Hook, NY, USA, 2024, pp. 35254–35269.
[36] Y. Liu *et al.*, "RoBERTa: A robustly optimized bert pretraining approach," 2019. doi: 10.48550/arXiv.1907.11692.
[37] J. Echeverrï£¡a *et al.*, "LOBO: Evaluation of generalization deficiencies in twitter bot classifiers," presented at ACSAC '18: Proc. 34th Annu. Comput. Secur. Appl. Conf., New York, NY, USA, 2018, pp. 137–146.
[38] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu and K. Weinberger, "Simplifying graph convolutional networks," presented at PMLR, 2019, pp. 6861–6871.
[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, "Graph attention networks," 2017. doi: 10.48550/arXiv.1710.10903.
[40] S. Shi, K. Qiao, S. Yang, L. Wang, J. Chen and B. Yan, "Boosting-GNN: Boosting algorithm for graph networks on imbalanced node classification," *Front. Neurorobot.*, vol. 15, 2021, Art. no. 775688. doi: 10.3389/fnbot.2021.775688.
[41] L. H. X. Ng and K. M. Carley, "BotBuster: Multi-platform bot detection using a mixture of experts," *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 17, no. 1, pp. 686–697, 2023. doi: 10.1609/icwsm.v17i1.22179.
[42] Z. Lei *et al.*, "BIC: Twitter bot detection with text-graph interaction and semantic consistency," *Proc. 61st Annu. Meet. Assoc. Comput. Linguist.* Toronto, ON, Canada, 2013, vol. 1, pp. 10326–10340. doi: 10.18653/v1/2023.acl-long.575.