Social Bot-Aware Graph Neural Network for Early Rumor Detection

Zhen Huang[1,*,✉], **Zhilong Lv**[1,*], **Xiaoyun Han**[1,*], **Binyang Li**[2,✉], **Menglong Lu**[1], **Dongsheng Li**[1]
[1]National University of Defense Technology, China
[2]University of International Relations, China
[1]{huangzhen, lvzhilong, hxy_1021, lumenglong, dsli}@nudt.edu.cn
[2]byli@uir.edu.cn

Abstract

Early rumor detection is a key challenging task to prevent rumors from spreading widely. Sociological research shows that social bots' behavior in the early stage has become the main reason for rumors' wide spread. However, current models do not explicitly distinguish genuine users from social bots, and their failure in identifying rumors timely. Therefore, this paper aims at early rumor detection by accounting for social bots' behavior, and presents a Social Bot-Aware Graph Neural Network, named SBAG. SBAG firstly pre-trains a multi-layer perception network to capture social bot features, and then constructs multiple graph neural networks by embedding the features to model the early propagation of posts, which is further used to detect rumors. Extensive experiments on three benchmark datasets show that SBAG achieves significant improvements against the baselines and also identifies rumors within 3 hours while maintaining more than 90% accuracy.

1 Introduction

Rumor is defined as unverified information at the time of posting (Qazvinian et al., 2011; Zubiaga et al., 2018; Lu et al., 2022). Malicious rumors that are spread massively on social media have become a threat to mislead the public and cause social panic. There is a need to debunk rumors in the early stage so as to prevent rumors from the wide spread.

Sociological research (Shao et al., 2018a) shows that there exist social bots during the rumor spread, and these bots are particularly active in the early stage of rumors, which will affect the real users through replies and mentions and accelerate the spread (Shao et al., 2018b; Beskow and Carley, 2018; Feng et al., 2022).

Current models (Ma et al., 2016; Chen et al., 2018; Song et al., 2019; Zhou et al., 2019; Xia et al., 2020; Han et al., 2021) mainly focus on post content or propagation sequence. These methods model posts as a chronological sequence, and extract the textual feature through GRU, LSTM, and CNN for rumor detection. Other methods (Liu and Wu, 2018; Yuan et al., 2020) account for early rumor detection by modeling user characteristics or credibility with user propagation structures. However, they do not explicitly distinguish genuine users from social bots, so the participation of social bots will lead to the failure of the features captured from both contents and propagation structures.

To this end, this paper presents a model named Social Bot-Aware Graph Neural Network (SBAG) for early rumor detection. This model consists of two parts: Social bot Detection (SD) and Bot-Aware Graph Rumor Detection (BAG). The former one is pre-trained based on a large sample of bot users and genuine users, to extract the features to compute the bot possibility for each user. The latter one transfers the SD to the Bot-Aware Graph Neural Network, which consists of GNN-based User Publishing (GUP), GAT-based User Interaction (GUI), and textual encoder components. For GUP, the bot possibility computed in SD is involved in the aggregation process. For GUI, the bot possibility is also integrated into the calculation of the attention weight of user-user. The textual encoder utilizes a convolutional neural network (CNN) to capture textual features. In this way, we take user publishing features, user interaction features, and textual features into consideration for early rumor detection. The codes will be open sourced[1]. Our main contributions are summarized as follows:

- According to the observation of sociological research, we consider social bots' behaviors, and train a social bot detection model based on twelve datasets. The results prove the consistency with the sociological research that the

[*]These authors contributed equally to this work.
✉Corresponding authors

[1]https://github.com/sky-star-moon/SBAG

bots are very active in the early stage.

- We propose a method named SBAG for early rumor detection, which implements early rumor detection by incorporating social bot detection. The results demonstrate that SBAG can achieve more than 93% accuracy, and detect 90% rumors within 3 hours.

2 Problem Definition

Assume a set of posts $\mathbf{R} = \{r_1, r_2, ..., r_{|\mathbf{R}|}\}$ and a set of users $\mathbf{U} = \{u_1, u_2, ..., u_{|\mathbf{U}|}\}$. Each post r corresponds to one publisher and multiple users to repost it. A user publishing graph $G_p =< V_p, E_p >$ is constructed to denote publisher-post relations, where V_p is the set of all publishers and source posts, E_p is the set of edges and a edge (u_i, r_j) indicates that user u_i publishes post r_j. A user interaction graph $G_u =< V_u, E_u >$ is constructed to denote user-user relations, where V_u is the set of all users, E_u is the set of edges and a edge (u_i, u_j) indicates that user u_i replies user u_j.

Since our motivation is to debunk rumors by incorporating the influence of social bots, our goal is to learn two classifiers for social bot detection and rumor detection, respectively. For the social bot detection task, a classifier $g : u \rightarrow Y_u$ is learned to identify whether a user u is a bot user or a genuine user. For the rumor detection task, a classifier is learned $f : r \rightarrow Y_r$ to predict the class of each source post r.

3 SBAG Model

The framework of the SBAG model is shown in Fig. 1. The model consists of two main parts, Social Bot Detection (SD) mentioned in §3.1 and Bot-Aware Graph Rumor Detection (BAG) mentioned in §3.2. We will introduce each module in detail.

3.1 Social Bot Detection

To incorporate bot behavior information into the model, we first pre-train the SD module on twelve datasets to learn the features of genuine users and bot users. Then we transfer this module to BAG as a bot possibility scorer, which assists in capturing the propagation pattern of bot behavior.

During the pre-training stage, we use a Multi-layer Perceptron (MLP) as the backbone network. Formally, let $c \in \mathbb{R}^v$ denote the user characteristics, such as length of username, number of followers, etc. Then c is normalized and fed into the module. The process is as follows:

$$\tilde{c} = \tanh(W_c^T c + b_c) \tag{1}$$

$$\hat{Y_u} = \text{softmax}(W_u^T \tilde{c} + b_u) \tag{2}$$

where $W_c \in \mathbb{R}^{v \times v}$, $W_u \in \mathbb{R}^{v \times 2}$, $b_c \in \mathbb{R}^v$ and $b_u \in \mathbb{R}^2$ are the parameters of the MLP, $\hat{Y_u} \in \{bot, human\}$ is the predicted probability distribution of the user class.

SD module will compute the users' bot possibility within [0,1] to indicate the degree to the user shows social bot behavior.

3.2 Bot-Aware Graph Neural Network

3.2.1 GCN-based User Publishing

Since user publishing graph G_p is a bipartite graph with only one hop at most, it is well locality. We design a GCN-based user publishing component. Formally, let $P \in \mathbb{R}^{m \times d}$ denote the initial embedding of the user nodes and $C \in \mathbb{R}^{n \times d}$ denote the initial embedding of the source post nodes, where m and n are the number of publisher nodes and source post nodes respectively, and d is the embedding dimension. We construct the adjacency matrix $A \in \mathbb{R}^{m \times n}$ base on G_p, where the element A_{ij} denotes user u_i publishes post r_j, then we normalize A to the matrix $\hat{A} = D_m^{-\frac{1}{2}} A D_n^{-\frac{1}{2}}$, where $(D_m)_{ii} = \sum_j A_{ij}$ and $(D_n)_{jj} = \sum_i A_{ij}$ are the diagonal matrices.

In order to incorporate the user's early bot possibility into the component, we constitute a bot possibility matrix $\hat{s} \in \mathbb{R}^{m \times d}$, where each element of row i of $\hat{s}$ is the bot possibility of publisher u_i and treat it as a bias. Finally, the aggregated features are summed with the initial features to obtain the publishing feature. The formulas are as follows:

$$\hat{C} = \text{ReLU}(\hat{A} C W_c + \hat{s}) \tag{3}$$

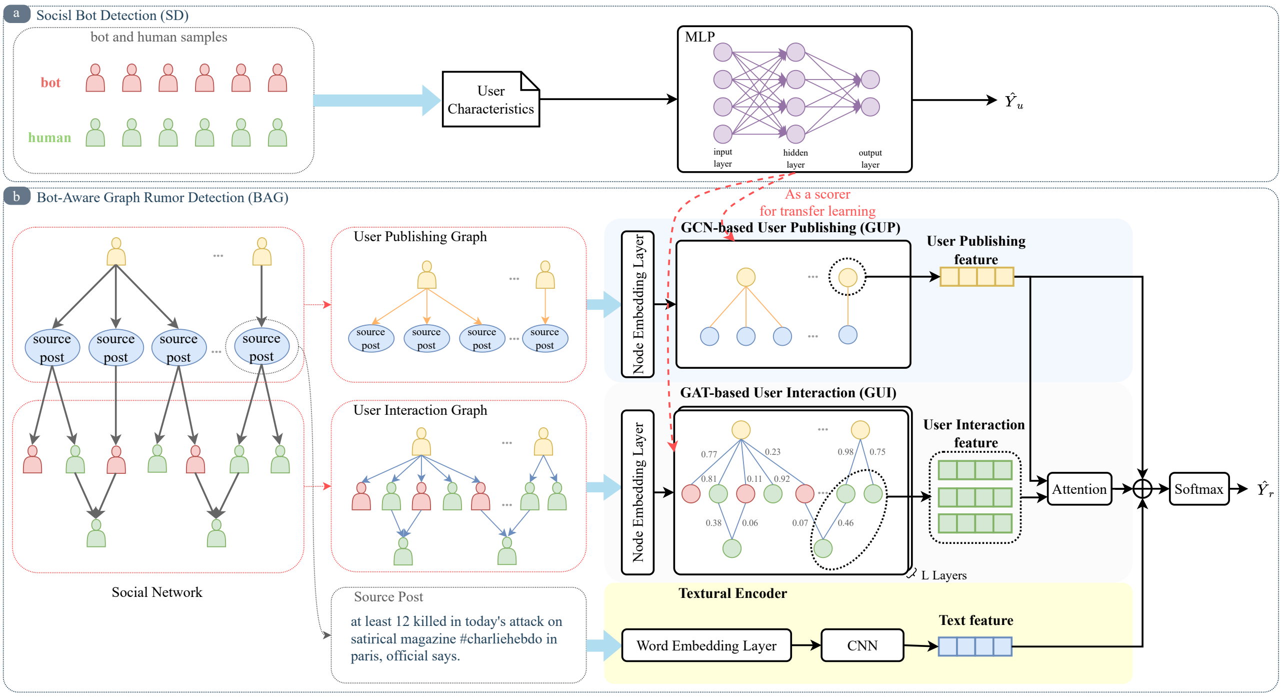
$$\hat{P} = \hat{C} + P \tag{4}$$

where $W_c \in \mathbb{R}^{d \times d}$ is the learnable matrix, $\hat{C} \in \mathbb{R}^{m \times d}$ is the aggregated feature, and $\hat{P} \in \mathbb{R}^{m \times d}$ is the publishing feature.

3.2.2 GAT-based User Interaction

In the user interaction graph G_u, considering the different importance of neighbor nodes to the target node, we design a GAT-based user interaction component.

Let $U^{(l)}$ denote the node features at layer l. Every user nodes' embedding in the graph is initialized to $U^{(0)} = \{u_0^{(0)}, u_1^{(0)}, ..., u_{|V_u|-1}^{(0)}\} \in \mathbb{R}^{|V_u| \times d}$



at least 12 killed in today's attack on satirical magazine #charliehebdo in paris, official says.

Figure 1: Overview of SBAG. **SD** scores the user's bot possibility according to user characteristics, then is transferred to the BAG module as a scorer. **BAG** module consists of three key components: GUP learns the publishing features of the publishers, GUI learns the interaction features of the repliers and textual encoder learns the textual features of the source post. Finally, the three types of features are fused to predict the class of the source post.

by a embedding layer according to normal distribution, and d is the dimension of the node embedding. Referring to the multi-head attention mechanism, the node feature at layer $l+1$ is updated as follows:

$$u_i^{(l+1)} = \Big\Vert_{k=1}^{K} \text{ReLU}\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l,k)} W_v^{(l,k)} u_j^{(l)}\Big) \quad (5)$$

where $\|$ denotes the concatenating operation. $\mathcal{N}(i)$ is the set of node i and its direct neighbors. $\alpha_{ij}^{(l,k)}$ is the attention weight of neighbor node j to target node i at the l-th layer in the k-th head, and $W_v^{(l,k)}$ is the learnable transformation matrix. Particularly, the output embedding in the last layer (denoted as the L-th layer) is the average of the features from the K heads instead of the concatenation. The formula is as follows:

$$u_i^{(L)} = \text{ReLU}\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(L-1,k)} W_v^{(L-1,k)} u_j^{(L-1)}\Big) \quad (6)$$

To capture the propagation pattern of bot behavior, we introduce bot possiblity into the attention weight $\alpha_{ij}^{(l,k)}$. Specifically, we utilize the SD mentioned in §3.1 to generate bot possibility s_i and s_j for two nodes of edge (u_i, u_j), and take their mean value as the edge weight e_{ij}, then $\alpha_{ij}^{(l,k)}$ is calculation as follows:

$$z_{ij}^{(l,k)} = \text{LeakyReLU}([W_q^{(l,k)} u_i \| W_k^{(l,k)} u_j] W_\alpha^{l}) \quad (7)$$

$$e_{ij} = \frac{s_i + s_j}{2} \quad (8)$$

$$\hat{z}_{ij}^{(l,k)} = e_{ij} \times z_{ij}^{(l,k)} \quad (9)$$

$$\alpha_{ij}^{(l,k)} = \frac{\exp(\hat{z}_{ij}^{(l,k)})}{\sum_{t\in\mathcal{N}(i)} \exp(\hat{z}_{ij}^{(l,k)})} \quad (10)$$

where $W_\alpha^{(l)}$, $W_q^{(l,k)}$ and $W_k^{(l,k)}$ are learnable parameters. Through the L graph attention layers, the interaction features $U^{(L)} = \{u_0^{(L)}, u_1^{(L)}, ..., u_{|V_u|-1}^{(L)}\}$ of all user nodes are obtained.

Next, for one source post r with one publisher and a repliers, the features of the publisher are $\tilde{P}_r \in \mathbb{R}^{1\times d}$, the features of the repliers are obtained from $U^{(L)}$, denoted as $\tilde{U} \in \mathbb{R}^{a\times d}$. To distinguish the importance of the repliers to the publisher, we calculate the attention weights and then aggregate the features from $\tilde{U}$ into an interaction feature $\tilde{U}_r$:

$$\beta = \text{softmax}(\tilde{U} W_\beta \tilde{P}_r^T) \quad (11)$$

$$\tilde{U}_r = \beta^T \tilde{U} \quad (12)$$

where β is the vector of the attention weight, $W_\beta \in \mathbb{R}^{d\times d}$ is the trainable matrix.

3.2.3 Textual Encoder

The semantic features of the post text are also important for rumor detection. For In this component, we utilize a CNN, which is consistent with the baseline models like SMAN and GLAN for fairness, to encode the source post. Each source post can be represented as a sequence of word embeddings $\mathbf{X} = [x_1, x_2, ..., x_{|\mathbf{X}|}] \in \mathbb{R}^{|\mathbf{X}| \times d}$. In CNN, one element of a feature map obtained from $\mathbf{X}$ through the convolutional operation is as follows:

$$h_i = \text{ReLU}(< W_h, x_{i:i+\omega-1} >_{\mathcal{F}}) \tag{13}$$

where $W_h \in \mathbb{R}^{\omega \times d}$ is a convolution kernel of size ω, and $\mathcal{F}$ is the Frobenius inner product. The feature map can be represented as $h = [h_1, h_2, ..., h_{|\mathbf{X}|-\omega+1}] \in \mathbb{R}^{|\mathbf{X}|-\omega+1}$. We then extract the maximum value from the feature map h to obtain $\hat{h} = \max(h)$.

We utilizes d filters of different kernel sizes ω, where $\omega \in \{3, 4, 5\}$, to obtain various features. Finally, the output of each filters are concatenated to obtain the textual feature $\tilde{X} \in \mathbb{R}^{1 \times 3d}$.

3.2.4 Output Layer

Assume that the textual feature of the source post r is $\tilde{X}_r \in \mathbb{R}^{1 \times 3d}$, the publishing feature of the publisher is $\tilde{P}_r \in \mathbb{R}^{1 \times d}$, and the aggregated interaction feature is $\tilde{U}_r \in \mathbb{R}^{1 \times d}$. We concatenate the features from different types of the source post, i.e., $\tilde{P}_r$, $\tilde{U}_r$, and $\tilde{X}_r$, to obtain the final feature of the source post. Lastly, the final feature is fed into a fully connected layer to predict the class:

$$\hat{Y}_r = \text{softmax}(W_r^T [\tilde{X}_r \| \tilde{P}_r \| \tilde{U}_r]^T + b_r) \tag{14}$$

where $W_r \in \mathbb{R}^{5d \times c}$ and $b_r \in \mathbb{R}^c$ are the weight and bias of the fully connected layer, and $\hat{Y}_r \in \{rumor, non\text{-}rumor\}$ or $\hat{Y}_r \in \{non\text{-}rumor, false\ rumor, true\ rumor, unverified\ rumor\}$ is the predicted class distribution.

3.3 Training

We apply the cross-entropy loss to optimize the social bot detection task and rumor detection task. The loss functions are as follows:

$$L_u = -\sum_{i=1}^{T} Y_{u_i} log \hat{Y}_{u_i} \tag{15}$$

$$L_r = -\sum_{j=1}^{|\mathbf{R}|} Y_{r_j} log \hat{Y}_{r_j} \tag{16}$$

where L_u is the cross-entropy loss of the social bot detection task, Y_{u_i} and $\hat{Y}_{u_i}$ are the ground truth and predicted label of the i-th user respectively, and T is the size of the social bot detection dataset. L_r is the cross-entropy loss of the rumor detection task, Y_{r_j} and $\hat{Y}_{r_j}$ is the ground truth and predicted label of the j-th source post respectively, and $|\mathbf{R}|$ is the size of the rumor detection dataset.

4 Experiments

4.1 Datasets

For rumor detection task, we conduct experiments on three benchmark datasets: Twitter15 (Ma et al., 2017), Twitter16 (Ma et al., 2017), and Weibo16 (Ma et al., 2016). The statistics of the three datasets are shown in Tab. 1.

For fair comparison, we choose the same way of splitting datasets as in the baseline work(Yuan et al., 2020), 10% of samples are selected as the validation set, and the rest of samples are split into the training set and testing set with a ratio of 3:1.

Dataset	Twitter15	Twitter16	Weibo16
# Source posts	1,490	818	4,664
# Non-rumors (NR)	374	205	2,351
# False rumors (FR)	370	205	2,313
# True rumors (TR)	372	205	0
# Unverified rumors (UR)	374	203	0
# Users	276,663	173,487	2,746,818
# Posts	331,612	204,820	3,805,656

Table 1: Statistics of the rumor detection datasets.

For the social bot detection task, we select 12 datasets provided by Bot Repository[2] and the statistics of the datasets are shown in Tab. 2. The datasets are split into the training set, testing set, and validation set with the ratio of 8:1:1.

4.2 Experimental Settings

For the SD module, since Twitter15 and Twitter16 do not involve the user characteristics, we utilize Twitter API to crawl user characteristics based on user ID. The user characteristics selection of the three datasets is not exactly the same. The details are shown in Tab. 3.

For the BAG module, the dimension of the node embedding d is 100, the number of heads of the Multi-Head Attention K is 8, the number of graph attention network layers is 2, and the convolutional kernel sizes are {3,4,5}. The model utilizes the

[2] botometer.osome.iu.edu/bot-repository

Dataset	**# bots**	**# humans**
caverlee(Lee et al., 2011)	0	14,895
cresci-17(Cresci et al., 2017)	9,894	3,474
pronbots(Yang et al., 2019)	17,882	0
celebrity(Yang et al., 2019)	0	5,918
vendor-purchased(Yang et al., 2019)	1,088	0
gilani-17(Gilani et al., 2017)	0	1,413
cresci-rtbust(Mazza et al., 2019)	0	340
cresci-stock(Cresci et al., 2018)	0	6,174
botowiki(Yang et al., 2020)	698	0
midterm-2018(Yang et al., 2020)	17,968	8,092
verified(Yang et al., 2020)	0	1,987
TwiBot-20(Feng et al., 2021)	0	5,237
Total	47,530	47,530

Table 2: Statistics of the social bot detection datasets.

Adam optimizer with 1e-3 learning rate and 1e-6 weight decay coefficient. Besides, the batch size is set to 16 and the epoch is set to 20.

Similar to the existing work(Liu and Wu, 2018; Yuan et al., 2019, 2020), we also adopt Accuracy, Precision, Recall, and F1 score as the evaluation metrics.

user characteristic	Twitter15	Twitter16	Weibo16
Length of username	✓	✓	✓
Length of screenname	✓	✓	✓
Length of description	✓	✓	✓
Followers count	✓	✓	✓
Friends count	✓	✓	✓
Listed count	✓	✓	
Favorites count	✓	✓	✓
Statuses count	✓	✓	✓
Days of Registration	✓	✓	✓
URL	✓	✓	
Protected	✓	✓	
Geo enabled	✓	✓	✓
Verified	✓	✓	✓
Profile use background image	✓	✓	
Default profile	✓	✓	

Table 3: User characteristics selection.

4.3 Baselines

To evaluate the performance of SBAG, we compare SBAG with the following methods:

(1) **DTR** (Zhao et al., 2015) is a decision tree-based ranking approach that searches for inquiry phrases, clusters controversial posts, and then ranks the clustered results.

(2) **DTC** (Castillo et al., 2011) is a decision tree model that uses hand-crafted features of posts to detect rumors.

(3) **RFC** (Kwon et al., 2017) is a random forest classifier that learns user, linguistic, and structural features of posts for rumor detection.

(4) **SVM-RBF** (Yang et al., 2012) is an SVM model with an RBF kernel, which classifies rumors based on statistical features of posts.

(5) **SVM-TS** (Ma et al., 2015) is a linear SVM model, which uses dynamic series-time structure to capture social context features over time.

(6) **cPTK** (Ma et al., 2017) is an SVM model, which uses the tree-based kernel to evaluate the similarity of propagation tree structures.

(7) **GRU** (Ma et al., 2016) utilizes RNN to learn the textual feature of the chronological post sequences to detect rumors.

(8) **RvNN** (Ma et al., 2018) models the source post and its reposts as a conversation tree, and adopts a recursive neural network to learn its propagation pattern.

(9) **PPC** (Liu and Wu, 2018) employs RNN and CNN to model the sequence based on user features for early rumor detection.

(10) **GLAN** (Yuan et al., 2019) models posts and users as a heterogeneous graph, and identifies rumors by local semantic features and global structural features extracted from the graph neural network.

(11) **SMAN** (Yuan et al., 2020) jointly optimizes rumor detection task and users' credibility prediction task via a structure-aware multi-head attention network for early rumor detection.

4.4 Experimental Results

4.4.1 Analysis of Rumor Detection

Tab. 4 and Tab. 5 show the rumor detection results on Twitter15, Twitter16, and Weibo16. SBAG achieves 93.8%, 94.6%, and 95.7% in terms of accuracy on three datasets, respectively, and outperforms the best run of the baseline models.

More detailedly, compared with traditional machine learning models, such as SVM-RBF, SVM-TS, and cPTK, SBAG can capture a higher-level representation of posts. Moreover, SBAG outperforms textual feature-based methods such as GRU and RvNN, which proves that the social bot-aware user features are effective in rumor detection. In addition, compared with PPC, GLAN, and SMAN which capture user propagation features or user credibility, SBAG achieves a better performance. It is because that SBAG is beneficial for exploring the features of social bot behaviors.

4.4.2 Analysis of Early Detection

To evaluate the timeliness of SBAG, we set different detecting deadlines, where we only utilize the interaction of users before the deadline. Fig.

Method	Twitter15					Twitter16				
	Acc.	NR-F1	FR-F1	TR-F1	UR-F1	Acc.	NR-F1	FR-F1	TR-F1	UR-F1
DTR	0.409	0.501	0.311	0.364	0.473	0.414	0.394	0.273	0.630	0.344
DTC	0.454	0.733	0.355	0.317	0.415	0.465	0.643	0.393	0.419	0.403
RFC	0.565	0.810	0.422	0.401	0.543	0.585	0.752	0.415	0.547	0.563
SVM-RBF	0.318	0.455	0.037	0.218	0.225	0.321	0.423	0.085	0.419	0.037
SVM-TS	0.544	0.796	0.472	0.404	0.483	0.574	0.755	0.420	0.571	0.526
cPTK	0.750	0.804	0.698	0.765	0.733	0.732	0.740	0.709	0.836	0.686
GRU	0.646	0.792	0.574	0.608	0.592	0.633	0.772	0.489	0.686	0.593
RvNN	0.723	0.682	0.758	0.821	0.654	0.737	0.662	0.743	0.835	0.708
PPC	0.842	0.811	0.875	0.818	0.790	0.863	0.820	0.898	0.843	0.837
GLAN	0.905	0.924	0.917	0.852	0.927	0.902	0.921	0.869	0.847	0.968
SMAN	0.914	0.915	0.926	**0.933**	0.881	0.935	0.946	0.920	0.905	0.968
SBAG	**0.938**	**0.965**	**0.953**	0.897	**0.933**	**0.946**	**0.947**	**0.930**	**0.926**	**0.978**

Table 4: Results of rumor detection on Twitter15 and Twitter16.

Method	Acc.	NR			FR		
		Precision	Recall	F1	Precision	Recall	F1
DTR	0.732	0.726	0.749	0.737	0.738	0.715	0.726
DTC	0.831	0.815	0.847	0.830	0.847	0.815	0.831
RFC	0.849	0.947	0.739	0.830	0.786	0.959	0.864
SVM-RBF	0.818	0.815	0.824	0.819	0.822	0.812	0.817
SVM-TS	0.857	0.878	0.830	0.857	0.839	0.885	0.861
GRU	0.910	0.952	0.864	0.906	0.876	0.956	0.914
PPC	0.921	0.949	0.889	0.918	0.896	0.962	0.923
GLAN	0.946	0.949	0.943	0.946	0.943	0.948	0.945
SMAN	0.951	0.937	**0.967**	0.952	**0.967**	0.936	0.951
SBAG	**0.957**	**0.967**	0.947	**0.957**	0.947	**0.967**	**0.957**

Table 5: Results of rumor detection on Weibo16.

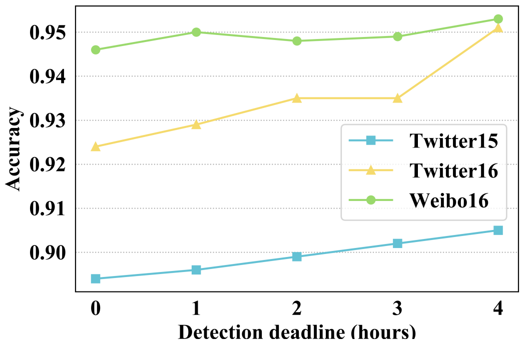


Figure 2: Results of timeliness.

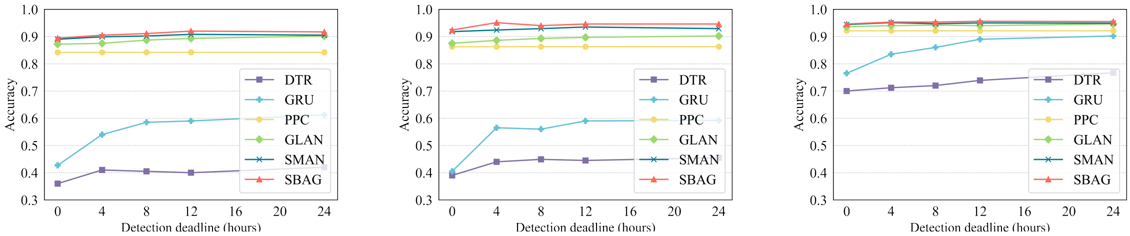
2 shows the results on Twitter15, Twitter16, and Weibo16. We can observe that within 0 to 3 hours, SBAG achieves the accuracy of over 90% on three datasets, and the results in the early stage are close to the results by accounting for all users, which indicates that SBAG has a strong capability for early detection.

Fig. 3 shows the comparison of early detection with several baselines on Twitter15, Twitter16, and Weibo16. We can see that SBAG can debunk rumors earlier, and maintain a high accuracy, which even outperforms the state-of-the-art baselines such as SMAN and GLAN. Furthermore, within 24 hours, SBAG can achieve similar performance to those with learning the features of all users.

4.5 Ablation Study

To demonstrate the effectiveness of different features, we also conduct ablation study, and the experiments are as follows:

(1) **-p**: removing the GUP, the model predicts



(a) Twitter15.

(b) Twitter16.

(c) Weibo16.

Figure 3: Results of early rumor detection on Twitter15, Twitter16 and Weibo16.

rumors by textual feature and interaction feature, without publisher feature.

(2) **-i**: removing the GUI, the model predicts rumors by textual feature and publishing feature, without interaction feature.

(3) **-p-i**: it means that we discard two components mentioned (1) and (2) and detect rumors by textual feature only.

(4) **-t**: removing the textual encoder, the model predicts rumors by publisher feature and interaction feature, without textual feature.

(5) **-s**: without the pre-trained scorer, we score the bot possibility randomly.

As shown in Tab. 6, we can observe that each component of SBAG is essential. Specifically, **-p** and **-i** perform worse than the original model on three datasets, which shows that the publishing feature and interaction feature are significant for rumor detection. There is a sharp decrease in **-p-i**, which indicates that it is suboptimal to detect rumors only by the textual feature. Besides, the performance of **-t** also decreases significantly. It is because the textual feature of the source post is crucial for detecting rumors. The results of **-p-i** and **-t** show that user features and textual features have a complementary relationship. The performance of **-s** demonstrates that social bot detection is beneficial for rumor detection.

Method	Twitter15	Twitter16	Weibo16
SBAG	**0.938**	**0.946**	**0.957**
-p	0.913	0.920	0.947
-i	0.894	0.924	0.946
-p-i	0.848	0.885	0.915
-t	0.658	0.723	0.919
-s	0.931	0.927	0.948

Table 6: Ablation Study (Acc.).

4.6 Analysis of Social Bot Detection

Then, we will analyze the performance of the pre-trained social bot detection module. As mentioned in Tab. 3, we select 15 and 10 user characteristics to represent users on Twitter and Weibo datasets, respectively. Therefore, we pre-train two social bot detection modules with different dimensions, i.e., MLP-15d and MLP-10d. For comparison, we choose the baseline models as follows:

(1) **Botometer-v4** (Sayyadiharikandeh et al., 2020) is a public program that can be used to evaluate the bot score of any user on Twitter.

(2) **AdaBoost** (Kudugunta and Ferrara, 2018) extracts 10 user characteristics to represent a user, and employs AdaBoost classifier for bot detection.

(3) **RF** (Yang et al., 2020) is a random forest model, which extracts 8 original features and 12 derived features from the user information, and utilizes the random forest classifier to identify users.

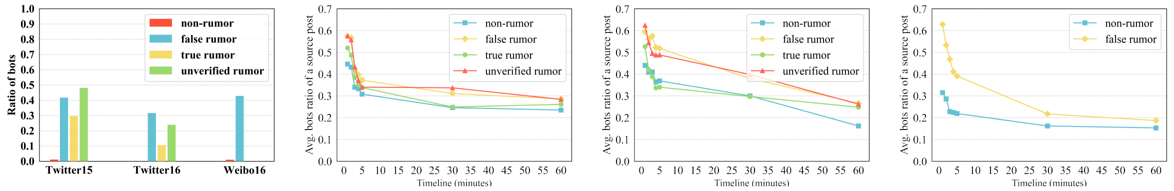
The results are shown in Tab. 7. The accuracy of MLP-15d and MLP-10d are better than the baselines, which indicates that they have a great ability to identify users. MLP-15d outperforms MLP-10d because the user information input to MLP-15d is richer. The MLP models perform better than the machine learning models like Botometer-V4, AdaBoost, and RF, which demonstrates that MLP can learn high-quality user representation with fewer features on this task.

4.7 Analysis of Social Bot Behavior

On the test sets of Twitter15, Twitter16, and Weibo16, we list the relation of rumors and publishers, i.e., the ratio of bot-behavior publishers under each source post class. As shown in Fig. 4a, we can observe that the bot possibility scorer identifies very few users who post non-rumors as bots. On the contrary, the bot possibility scorer identifies the majority of users who post rumors as bots. Moreover, in false rumors and unverified rumors, the

Metric	Botometer-V4	AdaBoost	RF	MLP-10d	MLP-15d
Acc.	0.722	0.917	0.930	0.935	**0.944**

Table 7: Result of social bot detection.



(a) Relationship between rumors and publishers.

(b) Twitter15: Avg. bots ratio per source post.

(c) Twitter16: Avg. bots ratio per source post.

(d) Weibo16: Avg. bots ratio per source post.

Figure 4: Relationship between rumors and users.

ratio of bots is higher than that in true rumors.

We also make statistics on the average ratio of bot-behavior users among all participants under a source post for each source post class over time. As shown in Fig. 4b-4d, in the first five minutes after the source post is published, the bot behaviors are more active than that in the successive period. The bot-behavior users' ratio of false rumors and unverified rumors is higher than that of non-rumors and true rumors. The results of SBAG are consistent with sociological research, which also prove that our model has strong interpretability.

4.8 Case Study

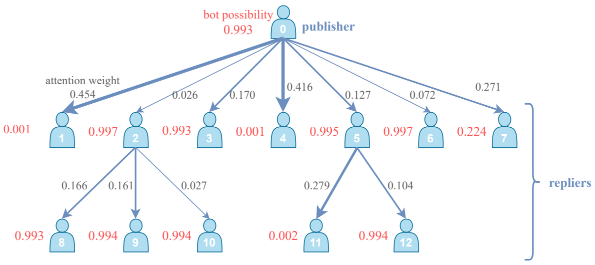


Figure 5: Case study.

To demonstrate the relation of bot possibility and attention weight, we choose a classic sample for visualization. Fig. 5 shows the attention weights of one publisher and the corresponding repliers, where the attention weights are computed by GUI. Fig. 5 illustrates that the edges connecting to users with lower bot possibility have higher attention weights. This way of aggregation helps learn more effective patterns of early propagation.

5 Related Work

Conventional rumor detection methods adopted machine learning to classify rumors based on the features of content, user, and propagation pattern, such as decision tree(Castillo et al., 2011), support vector machine(Yang et al., 2012; Liu et al., 2015; Ma et al., 2015; Wu et al., 2015), random forest(Kwon et al., 2013), etc. However, these methods involved feature engineering, which is hard to obtain high-order features.

Recent studies exploited deep learning methods for rumor detection. Most of the existing rumor detection methods mainly modeled a source post and its reposts together as a sequence or a graph, using RNN(Ma et al., 2016; Song et al., 2019; Zhou et al., 2019), CNN(Yu et al., 2017) , Transformer(Khoo et al., 2020; Rao et al., 2021), GCN(Bian et al., 2020; Song et al., 2021; Wei et al., 2021; Sun et al., 2022) and GAT(Lin et al., 2021) to learn the textual content features. However, these models do not consider the participant of social bots on social media to publish fraudulent content, which may lead to training noise by these fraudulent contents. Several studies(Liu and Wu, 2018; Yuan et al., 2019; Lu and Li, 2020; Yuan et al., 2020) integrated features of user feature or user credibility to learn the propagation patterns of source posts. However, they do not explicitly explore the unique user propagation pattern in the early stage, which limits the early-detection ability of the model.

6 Conclusion

In this work, according to the observation of sociological research, we propose a Social Bot-Aware Graph Neural Network for early rumor detection.

First, we pre-train a bot possibility scorer called SD on a large dataset containing bot users and genuine users, then SD is transferred to the BAG module. The BAG module takes the user's bot possibility information into the calculation of the features from different views, which enables the module to have a priori knowledge of the user in early detection. The experimental results on three public datasets show that SBAG effectively captures the early propagation of rumors, and further improves performance of early rumor detection.

Acknowledgements

This project was partially supported by National Natural Science Foundation of China (Grant number: 61976066), Beijing Natural Science Foundation (Grant number: 4212031), the Fundamental Research Fund for the Central Universities (Grant numbers:3262021T23), and Research Funds for NSD Construction, University of International Relations (Grant numbers: 2021GA07).

References

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

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 549–556.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 40–52. Springer.

Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion*, pages 963–972.

Stefano Cresci, Fabrizio Lillo, Daniele Regoli, Serena Tardelli, and Maurizio Tesconi. 2018. $ fake: Evidence of spam and bot activity in stock microblogs on twitter. In *Twelfth international AAAI conference on web and social media*.

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*, volume 36, pages 3977–3985.

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*, pages 4485–4494.

Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. 2017. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 349–354.

Xiaoyun Han, Zhen Huang, Menglong Lu, Dongsheng Li, and Jinyan Qiu. 2021. Rumor verification on social media with stance-aware recursive tree. In *International Conference on Knowledge Science, Engineering and Management*, pages 149–161. Springer.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8783–8790.

Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences*, 467:312–322.

Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PloS one*, 12(1):e0168344.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE International Conference on Data Mining (ICDM)*, pages 1103–1108. IEEE.

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

Hongzhan Lin, Jing Ma, Mingfei Cheng, Zhiwei Yang, Liangliang Chen, and Guang Chen. 2021. Rumor detection on twitter with claim-guided hierarchical graph attention networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10035–10047.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1867–1870.

Yang Liu and Yi-Fang Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Menglong Lu, Zhen Huang, Binyang Li, Yunxiang Zhao, Zheng Qin, and Dongsheng Li. 2022. Sifter: A unified framework for robust rumor detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1751–1754.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Michele Mazza, Stefano Cresci, Marco Avvenuti, Walter Quattrociocchi, and Maurizio Tesconi. 2019. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In *Proceedings of the 10th ACM conference on web science*, pages 183–192.

Vahed Qazvinian, Emily Rosengren, Dragomir Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599.

Dongning Rao, Xin Miao, Zhihua Jiang, and Ran Li. 2021. Stanker: Stacking network based on level-grained attention-masked bert for rumor detection on social media. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3347–3363.

Mohsen Sayyadiharikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2020. Detection of novel social bots by ensembles of specialized classifiers. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2725–2732.

Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018a. The spread of low-credibility content by social bots. *Nature communications*, 9(1):1–9.

Chengcheng Shao, Pik-Mai Hui, Lei Wang, Xinwen Jiang, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2018b. Anatomy of an online misinformation network. *PloS one*, 13(4):e0196087.

Changhe Song, Cheng Yang, Huimin Chen, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2019. Ced: Credible early detection of social media rumors. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3035–3047.

Yun-Zhu Song, Yi-Syuan Chen, Yi-Ting Chang, Shao-Yu Weng, and Hong-Han Shuai. 2021. Adversary-aware rumor detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1371–1382.

Mengzhu Sun, Xi Zhang, Jiaqi Zheng, and Guixiang Ma. 2022. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media.

Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue, and Songlin Hu. 2021. Towards propagation uncertainty: Edge-enhanced bayesian graph convolutional networks for rumor detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3845–3854.

Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *2015 IEEE 31st international conference on data engineering*, pages 651–662. IEEE.

Rui Xia, Kaizhou Xuan, and Jianfei Yu. 2020. A state-independent and time-evolving network for early rumor detection in social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9042–9051.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*, pages 1–7.

Kai-Cheng Yang, Onur Varol, Clayton A Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019. Arming the public with artificial intelligence to counter social bots. *Human Behavior and Emerging Technologies*, 1(1):48–61.

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*, volume 34, pages 1096–1103.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3901–3907.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2019. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In *2019 IEEE international conference on data mining (ICDM)*, pages 796–805. IEEE.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2020. Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5444–5454.

Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th international conference on world wide web*, pages 1395–1405.

Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. 2019. Early rumour detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1614–1623.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.