Rumor Detection Based on Conflict and Bot Features

Chenbo Fu[1,2,3](✉), Yuchen Xiong[1,2,3], Jianquan Qiu[1,3,6], Ye Wu[4,5], Yong Ming[3,4,5], Qi Xuan[1,2,3], and Shanqing Yu[1,2,3](✉)

[1] Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China

[2] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

[3] Binjiang Institute of Artificial Intelligence, ZJUT, Hangzhou 310056, China
{cbfu,yushanqing}@zjut.edu.cn

[4] Computational Communication Research Center, Beijing Normal University, Zhuhai 519087, China

[5] School of Journalism and Communication, Beijing Normal University, Beijing 100875, China

[6] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract. The complexity of rumors makes them spread fast and challenging to detect. Therefore, multi-modal rumor detection become one of the popular solutions. In this work, we integrate conflict features and bot features into the framework of rumor detection, and by utilizing graph convolutional networks to discern the underlying interplay between these factors, which enhances the model's understanding of the rumor spread process. Specifically, we employ a pre-trained BERT model to capture stance and construct conflict relations, and employ Bot detectors (i.e., Botometer X and Weibo Bot Finder) to obtain bot score confidences, and finally combine these two features to detect rumors. Experiments on two public datasets, i.e., PHEME and Weibo, validate the effectiveness of our proposed method. Our work supplements current rumor detection methods and highlights the important role of conflict and bot features.

Keywords: Rumor Dectection · Conflicts Features · Bot Features

1 Introduction

With the widespread use of social media, users can easily access information, express their opinions, and communicate with one another. The widespread use of social media has greatly facilitated users' access to information and channels of communication. However, social media has also accelerated the rapid spread of rumors [1], which has had a significant negative impact on society. Therefore, detecting rumors on social media becomes more and more pressing [2–4].

X. Meng et al. (Eds.): BDSC 2024, CCIS 2161, pp. 279–297, 2024.
https://doi.org/10.1007/978-981-97-5803-6_17

The negative effects of rumors have deeply influenced society [5–8]. For example, Allcott et al. [9] analyzed the 2016 US presidential election and found that rumors were widely spread three months before the election. Meanwhile, recent studies show that up to 71% of accounts discussing financial information about U.S. stocks on Twitter were found to be bots [10,11]. Bots spread rumors extensively and actively promote low-value stocks to lure uninformed investors to invest, engaging in cryptocurrency financial fraud [12]. Therefore, it is necessary to develop effective rumor detection methods [5,13,14]. Fortunately, with the continuous development of technology [15–21], more and more rumor detection methods have been proposed and are being used to detect rumors, this provides us with new tools for identifying rumors in an environment full of conflicts. However, there are a large number of bots in the existing media environment that confuse public opinion and influence collective positions through comments and speeches, thereby widely spreading and amplifying the influence of rumors [22]. These bots bring new challenges to existing detection methods because the environment where information conflict and uncertainty coexist seriously interferes with the learning and extraction of useful features, and it becomes more difficult to distinguish the text patterns behind rumors and non-rumors [23].

To address this challenge, the key factor is understanding and utilizing conflicts' role in spreading information. Rumor is more likely to provoke conflicts because it can trigger contradictory emotional reactions and resonate with people's feelings, making them more susceptible to believing in rumors [24,25]. However, existing conflict-related research focuses only on conflicts between news posts, ignoring conflicts between users [26,27]. To the best of our knowledge, most existing algorithms utilize the emotion features of rumor content delivered by publishers, paying little attention to the emotions in rumor comments [28]. Moreover, the viral spread of rumor among the population often leads to high arousal or activated emotions, thus triggering conflicts [29]. Therefore, it is necessary to explore user-to-user conflicts. Meanwhile, social bots distort public debate by spreading rumors, which can also be utilized as features to identify rumors.

In our work, we first define two types of conflict based on the stances of news posts and users: 1) **Local Conflict**, which is the stance conflict in the responsive relationships within the propagation network, and 2) **Global Conflict**, which is the stance conflict between users and news publishers. Specifically, local conflict is manifested in the opposition of stances when users comment or reply, such as the clash of support and opposition emotions when replying. Global conflict, on the other hand, is more macroscopic, focusing on the differences in stances between the user community and news publishers. In global conflicts, we mainly analyze the overall emotional trend and the emotional resonance or dissonance caused by news content. Our analyses indicate that these two types of conflict are closely related to the distinction between rumor and non-rumor. For example, if a large number of user comments are opposed to the stance of the news publishers, it suggests that the news content is more likely to be false. Furthermore, we extract the bot features of users based on their personal information and posting history. To effectively capture the impact of conflicts and bots on rumor detection, we combine the conflict features with bot features. By encoding text content, user stances in the spread network, and bot characteristics, we integrate user conflict features with bot features into the representation, proposing a new rumor detection model. Our results demonstrate that incorporating stance features and bot features significantly improves the accuracy of rumor detection.

Our main contributions are summarized as follows:

- We leverage a stance detection algorithm along with bot detection methodologies to furnish each post in the dataset with associated stance values and bot confidence scores, providing a richer context for rumor evaluation.
- We combine conflict features with bot features, harnessing social interactions and automated behaviors in datasets to enhance the effectiveness of rumor detection.
- We verify the performance of our proposed method on the two public datasets, i.e., the PHEME and Weibo datasets. Furthermore, we also conduct ablation experiments to discover the importance of conflict and bot features. Meanwhile, the bot feature experiments proved the effectiveness of attention and normalization processing.

The rest of this paper is organized as follows: Sect. 2 introduces related work, Sect. 3 presents the details of our method, Sect. 4 gives detailed experimental steps and results. Finally, Sect. 5 concludes our work.

2 Related Work

2.1 Single-Modal Rumor Detection

Existing Single-Modal rumor detection algorithms can be differentiated into methods based on content, social contexts, propagation information, and combined approaches [30,31].

Rumor Detection Based on Content. In traditional rumor detection, researchers primarily distinguish rumor from true news based on the content, using the news text as input for detection. For example, Ma et al. fed each news sentence into an RNN, LSTM, or GRU using the hidden vectors of the recurrent neural networks to represent news information and entering this hidden layer information into a classifier to obtain the classification result [32]. Yu et al. used a Convolutional Neural Network (CNN) to model content, mapping each post of a news event into vector space, then using CNN to extract textual features, and inputting the embedded vector into a classifier for the final classification result [33]. However, this approach ignores valuable information, such as the publisher's and spreader's features.

Rumor Detection Based on Social Context. Existing rumor authors often mimic the writing style of real news [30], and studies have shown that the credibility of news authors can aid in detecting rumors. News published by highly credible users are more likely to be real, while those by users with low credibility are more likely to be fake. Detection methods based on user credibility assess a user's credibility through their profile and history of posts to help detect rumors. Lu et al. constructed a user graph, using users' profiles as initial information for the nodes in the graph, obtaining user embedding through GCN for rumor detection [34]. Dou et al. identified user credibility based on their posting history for rumor detection [35]. However, rumor detection based on social context requires a substantial amount of data for assistance during detection. For newly emerging topics, there might be a lack of information or an inability to quickly obtain sufficient information for judgment.

Rumor Detection Based on Propagation. Besides content and user characteristics, propagation features also play an important role in rumor detection. Methods based on the news propagation tree model the spread of rumors as a tree structure, with the original news as the root node and shares and comments as branch nodes and leaf nodes. Ma et al. modeled the rumor spread process as a tree structure, constructing both a bottom-up and a top-down propagation tree and using recursive neural networks to model the nodes in the tree for rumor classification [36]. With the development of Graph Neural Networks (GNNs), many researchers have attempted to model the rumor spread process using graph structures. In the Bi-GCN study [37], the news propagation process was modeled as a graph structure, using top-down graphs to represent news propagation information and bottom-up graphs for news spread information, employing GCN to integrate node information in the graph to obtain node representations. Bi-GCN also introduced a root node reinforcement mechanism, suggesting that the original news can semantically enhance other information, thus incorporating original news information into other nodes. Song et al. used temporal features to create dynamic graphs, further enhancing rumor detection and achieved good results through graph neural networks based on propagation [38].

2.2 Multi-modal Rumor Detection

With the widespread adoption of social media and digital communication tools, the pathways for the spread of rumors have become more diverse and complex, especially in a modern trend of significant increases in multi-modal information (including text, images, and videos). Traditional single-modal-based rumor detection methods, such as those relying solely on textual content, are struggling to meet this challenge. This is because current technologies for editing images and videos can easily alter the semantic content of information, significantly diminishing the effectiveness of single-modal-based methods. Therefore, existing methods for detecting rumor in multi-modal contexts are categorized into three main approaches: the integration of multi-modal information, the exploitation of contrasts between modalities for identification purposes, and the enhancement of multi-modal information [6,39,40]. Singhal et al. employed VGG19 to extract visual cues and BERT for textual analysis, subsequently fusing these data types to feed into a classifier aimed at distinguishing rumors [41]. Another innovative approach, as outlined in Zhou et al.'s work [42], transformed visual information into textual format via an image-to-text model. Moreover, the synergy of BERT and ResNet in Qian et al.'s work [43] enabled the modeling of text and visual information, respectively. This approach used co-attention mechanisms to mutually enhance the text and visual data, enriching the detection process.

Furthermore, the multi-modal combination of text, social context, structure, and other information could also improve the performance of rumor detection. For example, GCAN [34] fused text embedding with propagation representations to detect rumors, showcasing the potential of comprehensive multi-modal analysis in tackling the complexities of rumor propagation. However, most multimodal methods only consider image, context, and structure information, which ignore the role of conflict and bot features. Conflicts involve differences in stance or sentiment between information sources or user comments, helping to uncover the controversial phenomena behind rumors [26,27,29]. And bot features may help us reduce the bias caused by bots. Therefore, this work proposes a rumor detection framework that integrates multi-modal information from text, conflict, and bot features.

2.3 Conflict Detection

Conflicts can arise between posts regarding a statement about an event, among comments from different users, or between users and the news. Thus, the conflict features could be used to improve the performance of rumor detection. For example, Ref. [26] detects and analyzes conflicts by stances across multiple news articles concerning a statement, accurately assessing the reliability of news articles and statements. Jin et al. [27] discover conflicting viewpoints in news tweets, then build a credibility propagation network of tweets linked with supporting or opposing relations to generate the final evaluation result for news. Lao et al. [44] develop a stance attention aggregator to measure response nodes' importance according to the stance conflict between the root node and child nodes, thereby improving detection performance.

2.4 Bot Detection

While exploring multi-modal rumor detection methods, the integration of social context features provides a new dimension to enhance the accuracy of detection models, such as features of social bots. Social bots are social media accounts driven by complex algorithms, capable of autonomously generating content and interacting with human users or other bots [45]. Some bots are designed as malicious entities to manipulate public opinion and spread rumors on social media platforms. Alessandro et al. [4] collected Twitter data during the four weeks leading up to the final vote of the 2016 US presidential election to estimate the impact of social bot spread rumors on social media. Their findings showed that 15% of the accounts were social bots, and about 19% of tweets were published by social bots [22]. Highly active malicious social bot accounts on social media are not only major sources of rumor but also significantly facilitate its spread [46–48]. By posting rumors and possibly commenting to support certain rumor stances, bots complicate the process of discerning the truthfulness of the information. In such an environment, contradictory and uncertain information coexist, severely hindering the identification and learning of effective features. Additionally, social bots may post provocative statements online, greatly affecting people's emotions and the context of online discussions, triggering negative emotions such as anger and fear, leading to doubt, distrust, and irrational behavior, which makes rumors easy to spread [25,46]. Therefore, the detection of social bots is important in detecting rumors, aiming to reduce the noise caused by those bots specifically designed to spread rumors.

3 Method

The architecture of our model is shown in Fig. 1, which contains four parts: (A) Graph Construction: Each post is embedded by using BERT to obtain text embedding y_m^t. Furthermore, each post is assigned stance values y_m^s, and bot values y_m^b through a stance detection algorithm and a bot detection algorithm, respectively. The weight of each edge is defined by the conflict value, which arises from the differences in stance values between connected nodes. The initial node embedding is calculated by multiplying the text embedding y_m^t with the attention-processed bot score coefficient y_m^b. (B) Dynamic graph construction module: build propagation conflict graphs and global conflict graphs; (C) Dual-GCN: A dual dynamic GCN module composed of dual static GCN units and a temporal fusion unit, which captures the structural features of events and integrates propagation information and conflict information at each time stage; (D) Classifier: Aggregates final propagation and text information to produce classification labels.

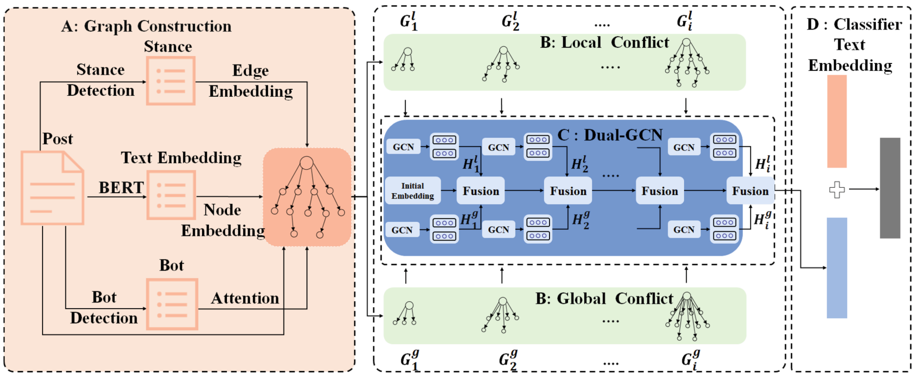


Fig. 1. The architecture of the proposed rumor detection model.

3.1 Problem Statement

Assuming the content set of each news article and its corresponding reply posts (comments) as $T^i = \{s^i, t_1^i, t_2^i..., t_m^i\}$, where s^i is the news post, i corresponds to the i^{th} news article, and t_m^i represents the m^{th} reply post of the i^{th} news article. The conflict situation for news article i and its corresponding reply posts are represented by $y_{jk}^c = y_j^s - y_k^s$, y_j^s and y_k^s are the stance value of two posts t_j^i and t_k^i, y_{jk}^c is the conflict score of these two posts. Thus, our model outputs the y_i $= \{0, 1\}$ to detect the event label, where $y_i = 1$ and $y_i = 0$ denote the event as rumor and non-rumor, respectively.

3.2 Graph Construction

Firstly, we conduct stance detection for each post in the dataset based on its text content. Utilizing the algorithm by He et al. [49] with a pre-trained BERT, we calculate the stance values y_m^s for each post. Then, we use the stance values y_m^s to construct two conflict relationships C_i^1 and C_i^2, respectively. Subsequently, we carry out bot detection for each post according to its author, i.e., bot score confidence y_m^b for the users. For the English dataset PHEME, we employ Botometer X [50], and for the Chinese dataset, we use Weibo Bot Finder [51].

After getting the conflict relationships and bot score confidence, each event is divided into three time windows. The first time window ranges from the news post to 1/3 of the life cycle, and the second time window ranges from the news post to 2/3 of the life cycle; similarly, the third time window includes the entire life cycle. Then, for each time window, corresponding propagation conflict graphs and global conflict graphs are constructed.

- Based on the conflict relationship C_i^1 generated between the source post and comment posts during propagation, the local conflict graph G_i^l is constructed, with the news source post s^i as the source node, comment posts t_m^i as child nodes, and the comment relationship as edges. In each news event, edges between two nodes with stance conflicts are defined as conflict edges, and the weight of the edge is determined by the conflict value between the two nodes. After constructing the local conflict graph G_i^l related to each news s^i, this is used to analyze the impact of conflicts between users on rumor detection during the news propagation process.
- The global conflict graph G_i^g is directly constructed based on the conflict relationship C_i^2 between the source post and various comment posts, with the news source post s^i as the source node, and various user comment posts t_m^i as child nodes. If there is a conflict between the news source post node and a user comment post node, the two nodes are connected. After constructing the global conflict graph G_i^g related to each news, this is used to analyze the overall conflict level of the news and the impact of conflicts between news publishers and users on rumor detection.

3.3 Bot Feature

For all users in the dataset, social bot detection is conducted to calculate the bot score for each user, and all posts and corresponding replies are assigned social bot feature values based on the bot score of their publishing user, enhancing the node features in the propagation graph with bot features. Using the algorithm to detect bot score confidence where a higher value indicates a higher likelihood of being a bot. After normalizing, the bot scores range from 0 to 1, with a smaller value indicating a higher likelihood of the node being a bot. The bot confidence coefficients of all nodes are further refined using an attention mechanism. Each node is then multiplied by the embedding corresponding to its user, yielding new node feature values adjusted for bot features.

3.4 Dual GCN Module

The constructed graphs from Sect. 3.2 are analyzed using a Graph Convolutional Network (GCN), where each news time window is represented by two subgraphs: the local conflict graph G_i^l and the global conflict graph G_i^g. These graphs serve as inputs to the GCN, where they are encoded sequentially to obtain propagation conflict features H_i^L and global conflict features H_i^G. Leveraging a dual-static GCN unit, an initial embedding $H_{(0)}$ is assigned at the start. Node embedding H_i^L and H_i^G are then derived from the respective dual-static GCN units of the propagation conflict graph and the global conflict graph. The three embedding H_i^L, H_i^G, and $H_{(0)}$ are combined, and this fused embedding becomes the initial node embedding $H_{(1)}$ for the next stage unit. This fusion process is adopted in subsequent GCN units to obtain the features of all news $H_{(i)}$, thereby linking the conflict features of all news windows throughout the analysis.

Single Dual GCN. We utilize two GCNs to concurrently encode the global propagation graph and the local propagation graph, thereby obtaining global conflict features and local conflict features.

First, we introduce how to encode the dynamic propagation graph, and the process for dynamic knowledge graphs is similar. Let $H \in \mathbb{R}^{m_i \times F}$ be the matrix containing all m_i node features in the propagation graph G_p, where F is the dimension of the feature vector. Let $h \in \mathbb{R}^F$ represent the feature vector of the node u in the graph. Its adjacency matrix is $A \in \mathbb{R}^{m_i \times m_i}$. The dual-layer GCN is defined as

$$H_{ir}^{P(1)} = g\left(\tilde{A}_{ir}^{P} H_{ir}^{(0)} W_r^{P(0)} + b_r^{P(0)}\right) \tag{1}$$

$$H_{ir}^{P(2)} = g\left(\tilde{A}_{ir}^{P} H_{ir}^{P(1)} W_r^{P(1)} + b_r^{P(1)}\right) \tag{2}$$

where $H_{ir}^{P(0)}$ and $H_{ir}^{P(1)}$ are node feature matrices, $W_r^{P(0)} \in \mathbb{R}^{F \times F_1}$, $W_r^{P(1)} \in \mathbb{R}^{F \times F_2}$, $b_r^{P(0)} \in \mathbb{R}^{F_1}$ and $b_r^{P(1)} \in \mathbb{R}^{F_2}$ are trainable parameters. F_1 and F_2 are the output node feature dimension sizes for the first and second layers respectively, and g is a non-linear activation function. $\tilde{A}^P_{ir}$ is the normalized symmetric adjacency matrix.

For the l-th layer GCN, we enhance the feature matrix by concatenating each node's hidden feature vector with the source post's node vector learned from the previous layer. This process yields the following equations:

$$\tilde{H}^{P(1)}_{ir} = \text{Concat}\left(H^{P(1)}_{ir}, (H^{P(0)}_{ir})_{\text{Source}}\right) \tag{3}$$

$$H^{P(2)}_{ir} = \sigma\left(\tilde{A}^p_{ir}\tilde{H}^{P(1)}_{ir}W^{P(1)}_r + b^{P(1)}_r\right) \tag{4}$$

$$\tilde{H}^{P(2)}_{ir} = \text{Concat}\left(H^{P(2)}_{ir}, (H^{P(1)}_{ir})_{\text{Source}}\right) \tag{5}$$

$$\hat{H}^{P(2)}_{ir} = \text{ReLU}\left(W^P_r\tilde{H}^{P(2)}_{ir} + b^P_r\right) \tag{6}$$

where $W^{P(1)}_r \in \mathbb{R}^{2F_1\times F_2}$, $b^{P(1)}_r \in \mathbb{R}^{F_2}$, $\hat{H}^{P(2)}_{ir} \in \mathbb{R}^{m_i\times F}$, $W^P_r \in \mathbb{R}^{2F_2\times F}$, $b^P_r \in \mathbb{R}^F$. For simplicity, we set $F = F_1 = F_2$.

Time Fusion Module. In the previous step, based on a single dual-static GCN unit, we extracted node embedding from each time window. Next, we designed a temporal fusion unit to combine these node embeddings, and the fused embedding will serve as the initial node embedding for the dual-static GCN unit in the next time window. We considered three types of node embedding: two obtained by applying the GCN on the local conflict graph and the global conflict graph, and the initial node embedding from the previous time window. To achieve this, we applied three different linear transformations to these node embedding, using three trainable weight matrices W^0_{tf}, W^L_{tf} and $W^G_{tf} \in \mathbb{R}^{F\times F}$. $\tilde{H}^{L(2)}_{ir}$ and $\tilde{H}^{G(2)}_{ir}$ are the corresponding $\hat{H}^{P(2)}_{ir}$ in the global conflict graph G^g_i and local conflict graph G^l_i respectively.

$$\begin{aligned} *H^{P(0)}_{ir} &= W^0_{tf}H^{P(0)}_{ir} \\ *H^{L(2)}_{ir} &= W^L_{tf}\tilde{H}^{L(2)}_{ir} \\ *H^{G(2)}_{ir} &= W^G_{tf}\tilde{H}^{G(2)}_{ir} \end{aligned} \tag{7}$$

The $*H^{P(0)}_{ir}$, $*H^{L(2)}_{ir}$, $*H^{G(2)}_{ir}$ are the value after linear transformation. We then concatenate the three transformed features, combining source post embedding, responsive post embedding, and initial node embedding. The initial node embedding of all posts for the first time window is the pre-trained text embedding of the event. The concatenation matrix goes through a linear function and a tanh activation function to produce $\hat{H}_{ir}$, which serves as the initial input for the next time window.

$$\text{Concat}_H = \text{Concat}(*H^{P(0)}_{ir}, *H^{L(2)}_{ir}, *H^{G(2)}_{ir}) \tag{8}$$

At each time window, the initial node embedding $*H_{ir}^{P(0)}$ is concatenated with other features to emphasize the importance of initial semantic information, further leveraging the annotated semantic embedding. To capture and integrate the structural information of the graph at each time stage in the form of node embedding, this embedding will serve as the initial node embedding propagated to the next stage's dual-static GCN. We have designed the following cross-stage graph convolutional layers.

$$\hat{H}_{ir} = \tanh(W_{tf}\mathrm{Concat}_H + b_{tf}) \tag{9}$$

$$H_{i(r+1)}^{L(1)} = g\left(\tilde{A}_{ir+1}\hat{H}_{ir}W_{r+1}^{L(0)} + b_{r+1}^{L(0)}\right) \tag{10}$$

$$H_{i(r+1)}^{G(1)} = g\left(\tilde{A}_{ir+1}\hat{H}_{ir}W_{r+1}^{G(0)} + b_{r+1}^{G(0)}\right) \tag{11}$$

3.5 Rumor Classification Module

The fused embedding node output from the final GCN unit module represents the common conflict features and bot features of all news post nodes and comment nodes, denoted as H_i. Then, this representation H_i is concatenated with the semantic representation of the source post node, merging the information into a final representation.

$$\tilde{S} = \mathrm{Concat}(H_i, \mathrm{BERT}(s_i)) \tag{12}$$

Finally, the label for the event $\hat{y}$ is computed through several fully connected layers followed by a sigmoid layer:

$$\hat{y}_i = \sigma(w_f\tilde{S} + b_f) \tag{13}$$

where w_f and b_f are the weight and bias parameters, respectively. We then use cross-entropy loss as the loss function for rumor classification:

$$\mathcal{L}_c = -\sum_i y_i \log(\hat{y}_i) \tag{14}$$

4 Experiment

4.1 Datasets

We conducted experiments on two real-world datasets, one in English (PHEME) [52] and the other in Chinese (Weibo) [32]. The statistical results of these datasets are shown in Table 1. The PHEME dataset includes 5,748 events, while the Weibo dataset has 4,664 events. Both datasets include labels and posts' publication time.

Table 1. Statistics of the PHEME and Weibo datasets.

	PHEME	Weibo
Posts	102,440	3,752,428
Events	5,748	4,664
Non-Rumors	3,654	2,351
Rumors	2,094	2,312

4.2 Baseline

We used the following baseline model:

- BERT [16] is a pre-trained language model based on deep bidirectional transformers.
- RVNNBU [36] learns discriminative features from content by following its non-sequential propagation structure, representing the bottom-up tree-structured network of RVNN.
- RVNNTD [36] also learns discriminative features from content by adhering to its non-sequential propagation structure, but it represents the top-down tree-structured network of RVNN.
- BiGCN [37] is a model based on GCN that can embed both propagation and dispersion structures and enhance node representations with root node features.
- GACL [53] is a model based on adversarial graph contrastive learning, which uses BERT for word embedding.
- CBT-HGCN [54] is a hypergraph convolutional network that leverages a hypergraph to model the internal structure and content of tweet conversations, enhancing misinformation detection by capturing the complex relationships between tweets and their replies.
- DDGCN [55] is a GCN-based model that utilizes propagation graphs and knowledge graphs for the detection.

4.3 Experimental Settings

In all comparison methods, we adopt the default optimization settings as reported in the corresponding papers. We implement our method using the Pytorch framework [56]. Parameters were optimized using the Adam algorithm [57]. We divide the Pheme dataset and Weibo dataset into training, validation, and test sets with no overlap between them. The best parameter settings are selected based on the performance of the validation set. In our model, we use the default threshold parameters of two bot detectors, i.e., the threshold for Botometer is set at 2.5, and the threshold for Weibo Bot Finder is set at 0.5. We utilize the pre-trained BERT model, with bert-base-uncased for English and bert-base-chinese for Chinese. We employ four metrics to evaluate model performance: Accuracy, Precision, Recall, and F1 Score. To ensure the stability and reliability of the model, we conduct a 5-fold cross-validation by randomly splitting the datasets into five parts.

4.4 Results

In this section, we compare our proposed method with baseline models. Table 2 presents a comparison of the performances of various models. Across both datasets, our model outperformed all other methods on all metrics, confirming the benefits of considering conflict and bot features for the task of rumor detection. While BERT shows prominence among text-based models, it still lags behind the remaining strategies, demonstrating the importance of combining multiple features. Compared to models focusing solely on text, BiGCN's performance enhancement is significant; it exceeds BERT by 5.2% on Weibo and by 1.8% on PHEME, indicating the importance of propagation structure in rumor detection. GACL achieves comparable performance to BiGCN. The comparison with CBT-HGCN shows that our method is better than using only the hypergraph and the relationship between tweets. Moreover, it is noteworthy that DDGCN surpasses most text-based and temporal-based models on the PHEME dataset. This suggests that integrating knowledge features provides additional layers of information that can significantly enhance the accuracy of detection. Compared to baseline models, our approach captures a broader range of conflict information and bot information from source posts and reply posts, thus enabling us to achieve the best performance.

Table 2. Experimental Results on the PHEME and Weibo datasets.

Method	PHEME				Weibo			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
BERT	0.821	0.818	0.788	0.803	0.866	0.867	0.866	0.866
RVNNTD	0.804	0.803	0.803	0.803	0.847	0.843	0.847	0.845
RVNNBU	0.789	0.788	0.788	0.789	0.903	0.901	0.897	0.900
BiGCN	0.840	0.840	0.834	0.835	0.919	0.918	0.916	0.913
GACL	0.831	0.804	0.799	0.782	0.937	0.931	0.931	0.937
CBT-HGCN	0.854	0.852	0.857	0.855	0.893	0.806	0.806	0.892
DDGCN	0.849	0.835	0.838	0.833	0.940	0.937	0.940	0.939
Ours	**0.867**	**0.858**	**0.856**	**0.860**	**0.944**	**0.940**	**0.941**	**0.940**

4.5 Ablation Study

In our study, we assessed the impact of the proposed components by introducing the following variants: (1) w/o global: wherein the global conflict features were excluded; (2) w/o local: wherein the local conflict features were omitted; (3) w/o conflict: wherein no conflict features were utilized; and (4) w/o bot: wherein the processing of bot-related features was eliminated.

As depicted in Table 3, all ablation variants exhibit performance decrements compared to our complete Ours model across both datasets. Notably, the removal of all conflict features results in a 2.7% decrease for the Pheme dataset and a 1.4% reduction in accuracy for the Weibo dataset, highlighting the significance of conflict features. Moreover, the absence of bot feature processing accounts for a 1.2% decrease in accuracy on both the PHEME and Weibo datasets.

Table 3. Ablation study results on the PHEME and Weibo datasets.

Method	PHEME				Weibo			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Ours	**0.867**	**0.858**	**0.856**	**0.860**	**0.944**	**0.940**	**0.941**	**0.940**
-w/o global	0.850	0.836	0.843	0.828	0.938	0.934	0.935	0.934
-w/o local	0.846	0.831	0.845	0.819	0.935	0.930	0.932	0.931
-w/o conflict	0.840	0.824	0.841	0.809	0.930	0.926	0.927	0.926
-w/o bot	0.855	0.848	0.840	0.836	0.932	0.928	0.929	0.928

4.6 Bot Feature Selection

In our approach, four strategies are employed to incorporate bot features into our node features as coefficients: (1) Binary: nodes with bot confidence exceeding a specific threshold are assigned a coefficient of 0, while legitimate users were assigned a coefficient of 1, effectively removing the bot nodes from the propagation graph to mitigate the influence of bots on rumor detection; (2) Double: nodes with high bot confidence were assigned a coefficient of 2 and normal users 1, thus amplifying the bot interference in rumor detection; (3) Confidence Normalization: bot confidence for all nodes was normalized across the confidence range and directly multiplied with the node features as coefficients; (4) Confidence Normalization with Attention: after normalization, the bot confidence coefficients of all nodes were further refined using an attention mechanism.

As shown in Table 4, the experimental results indicated that incorporating bot features through attention and confidence normalization yielded the most significant improvements.

Figure 2 is an illustration of the propagation graph, showing the importance of different nodes, where the size of the nodes represents their attention score. Blue nodes indicate bot and red nodes indicate human. The size of the nodes is proportional to their attention scores, i.e., the larger the attention score, the larger the node size, indicating the higher importance of that node in the model

Table 4. Bot Feature Selection on the PHEME and Weibo datasets.

Method	PHEME				Weibo			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Binary	0.856	0.845	0.845	0.844	0.934	0.934	0.935	0.933
Double	0.855	0.842	0.848	0.836	0.933	0.933	0.932	0.933
Normalized	0.858	0.848	0.850	0.846	0.936	0.935	0.934	0.934
Normalized + Attention	**0.867**	**0.858**	**0.856**	**0.860**	**0.944**	**0.940**	**0.941**	**0.940**

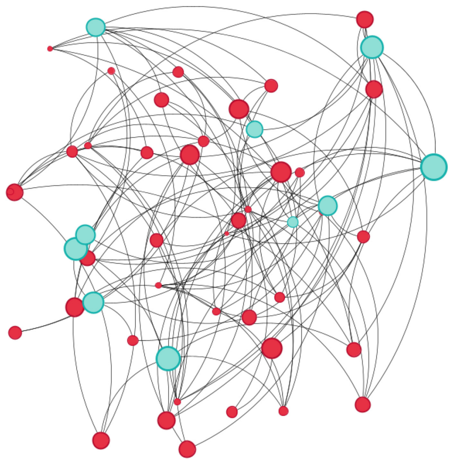
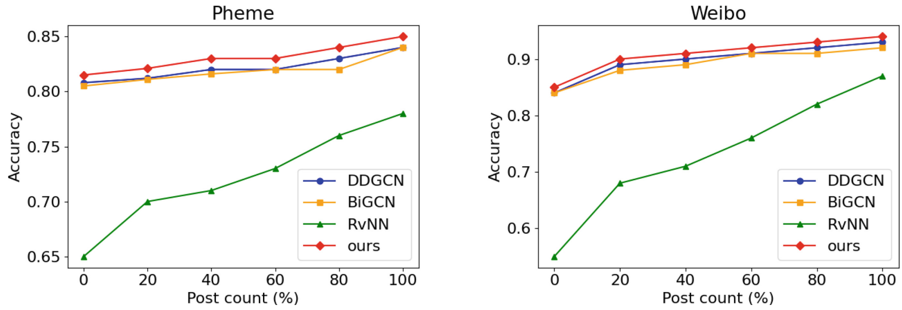


Fig. 2. The illustration of the propagation graph, where blue nodes indicate bot and red nodes indicate human, the size of the nodes is proportional to their attention score. (Color figure online)

prediction. The illustration demonstrates that, compared to human accounts, the node size of bot accounts is generally larger, which means they have higher attention scores than human accounts and play a more significant role in rumor detection. The result of this illustration is consistent with the ablation experiments, indicating the indispensability of bot features in detection.

4.7 Early Detection

In this work, we design our experimental setup based on the number of comments received. For comparative purposes, we selected RvNNTD, BiGCN, and DDGCN as our early detection baselines, as they all account for temporal information. Figure 3 presents the performance results on two datasets. As shown in Fig. 3, our model achieves high accuracy at each early stage after the posting of the source post.



(a) PHEME (b) Weibo

Fig. 3. Early rumor detection accuracy of different methods on (a) PHEME and (b) Weibo datasets.

5 Discussions and Conclusion

In this work, we introduce a dual-dynamic graph convolutional rumor detection model that leverages features of conflicts and bots. Our model contains two dynamic GCNs, operating on local and global conflict graphs respectively to capture conflict information across various dimensions of the spread process. These dynamic GCNs not only extract information on direct interactions between users (local conflicts), as well as stance differences between users and news authors (global conflicts) but are also adjusted by bot features to effectively detect rumors in the propagation of information. Experiments on two public datasets, PHEME and Weibo, show our method outperforms baseline models. Additionally, our ablation study shows the effectiveness of conflict features and bot features in rumor detection, and the attention-processed bot score further amplifies the model's performance. In conclusion, our work supplements current approaches to rumor detection and uncovers the important role of conflict and bot features in rumor propagation.

In our current study, we only analyze rumors using Chinese and English datasets. Although this provides valuable insights into the effectiveness of our proposed method, there remains significant potential to enhance and expand our research by incorporating multi-language datasets, which we plan to include multi-language datasets in future work. Additionally, we only use Botometer and Weibo Bot Finder for bot detection, which may introduce bias and represent a limitation of our work. Thus, in future work, we will consider the impact of different threshold values.

Acknowledgements. This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant (ZCLY24F0302), the Zhejiang Fundamental Public Welfare Research Project under Grant (LGF21G010003), the National Natural Science Foundation of China under Grant (62103374 and U21B2001), the Key R&D Program of Zhejiang under Grant (2022C01018), the Social Science Fund of Beijing under Grant (21DTR040), and the Key Research and Development Program of Zhejiang Province under Grant (2024C01025).

References

1. Peng, H., et al.: Streaming social event detection and evolution discovery in heterogeneous information networks. ACM Trans. Knowl. Discov. Data (TKDD) **15**(5), 1–33 (2021)
2. Pang, G., Shen, C., Cao, L., Van Den Hengel, A.: Deep learning for anomaly detection: a review. ACM Comput. Surv. (CSUR) **54**(2), 1–38 (2021)
3. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R.: Detection and resolution of Rumours in social media: a survey. ACM Comput. Surv. (CSUR) **51**(2), 1–36 (2018)
4. Bessi, A., Ferrara, E.: Social bots distort the 2016 us presidential election online discussion. First Monday **21**(11-7) (2016)
5. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359**(6380), 1146–1151 (2018)
6. Chen, J., Jia, C., Zheng, H., Chen, R., Fu, C.: Is multi-modal necessarily better? robustness evaluation of multi-modal fake news detection. IEEE Trans. Network Sci. Eng. (2023)
7. Fu, C., Chen, K., Pan, X., Yu, S., Ni, J., Min, Y.: Rumor detection based on the temporal sentiment. In: Meng, X., Xuan, Q., Yang, Y., Yue, Y., Zhang, Z.K. (eds.) BDSC 2022, LNCS, vol. 1640, pp. 275–290. Springer, Cham (2022). https://doi.org/10.1007/978-981-19-7532-5_18
8. Chenbo, F., Pan, X., Liang, X., Shanqing, Y., Xiaoke, X., Min, Y.: Feature drift in fake news detection: an interpretable analysis. Appl. Sci. **13**(1), 592 (2023)
9. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. J. Econ. Perspect. **31**(2), 211–236 (2017)
10. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: Emergent properties, models, and laws of behavioral similarities within groups of twitter users. Comput. Commun. **150**, 47–61 (2020)
11. Cresci, S., Lillo, F., Regoli, D., Tardelli, S., Tesconi, M.: Fake: evidence of spam and bot activity in stock microblogs on twitter. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 12 (2018)
12. Nizzoli, L., Tardelli, S., Avvenuti, M., Cresci, S., Tesconi, M., Ferrara, E.: Charting the landscape of online cryptocurrency manipulation. IEEE Access **8**, 113230–113245 (2020)
13. Asudeh, A., Jagadish, H.V., Wu, Y., Yu, C.: On detecting cherry-picked trendlines. Proc. VLDB Endowment **13**(6), 939–952 (2020)
14. Chen,Y., Conroy, N.J., Rubin, V.L.: Misleading online content: recognizing clickbait as "false news". In: Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection, pp. 15–19 (2015)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

16. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
17. Chenbo, F., et al.: Link weight prediction using supervised learning methods and its application to yelp layered network. IEEE Trans. Knowl. Data Eng. **30**(8), 1507–1518 (2018)
18. Xuan, Q., et al.: Subgraph networks with application to structural feature space expansion. IEEE Trans. Knowl. Data Eng. **33**(6), 2776–2789 (2019)
19. Chenbo, F., Zheng, Y., Liu, Y., Xuan, Q., Chen, G.: NES-TL: network embedding similarity-based transfer learning. IEEE Trans. Network Sci. Eng. **7**(3), 1607–1618 (2019)
20. Chenbo, F.: A novel spatiotemporal behavior-enabled random walk strategy on online social platforms. IEEE Trans. Comput. Soc. Syst. **9**(3), 807–817 (2021)
21. Fu, C., Luo, H., Liang, X., Min, Y., Xuan,Q., Chen, G.: Mutual influence in citation and cooperation patterns. IEEE Trans. Comput. Soc. Syst. (2023)
22. Willmore, A.: This analysis shows how viral fake election news stories outperformed real news on facebook (2016)
23. Ma, J., Li, J., Gao, W., Yang, Y., Wong, K-F.: Improving rumor detection by promoting information campaigns with transformer-based generative adversarial learning. IEEE Trans. Knowl. Data Eng. (2021)
24. Li, C., et al.: Joint stance and rumor detection in hierarchical heterogeneous graph. IEEE Trans. Neural Netw. Learn. Syst. **33**(6), 2530–2542 (2021)
25. Stella, M., Ferrara, E., De Domenico, M.: Bots increase exposure to negative and inflammatory content in online social systems. Proc. Natl. Acad. Sci. **115**(49), 12435–12440 (2018)
26. Zhang, D., Jiawei, X., Zadorozhny, V., Grant, J.: Fake news detection based on statement conflict. J. Intell. Inf. Syst. **59**(1), 173–192 (2022)
27. Jin, Z., Cao, J., Zhang, Y., Luo, J.: News verification by exploiting conflicting social viewpoints in microblogs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
28. Luvembe, A.M., Li, W., Li, S., Liu, F., Xu, G.: Dual emotion based fake news detection: a deep attention-weight update approach. Inf. Process. Manag. **60**(4), 103354 (2023)
29. Levy, S., Kraut, R.E., Yu, J.A., Altenburger, K.M., Wang, Y.-C.: Understanding conflicts in online conversations. In: Proceedings of the ACM Web Conference 2022, pp. 2592–2602 (2022)
30. Zhou, X., Zafarani, R.: A survey of fake news: fundamental theories, detection methods, and opportunities. ACM Comput. Surv. (CSUR) **53**(5), 1–40 (2020)
31. Lin, H., Zhang, X., Fu, X.: A graph convolutional encoder and decoder model for rumor detection. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 300–306. IEEE (2020)
32. Ma, J., et al.: Detecting rumors from microblogs with recurrent neural networks (2016)
33. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T., et al.: A convolutional approach for misinformation identification. In: IJCAI, pp. 3901–3907 (2017)
34. Lu, Y.-J. Li, C.-T.: GCAN: graph-aware co-attention networks for explainable fake news detection on social media. arXiv preprint arXiv:2004.11648 (2020)
35. Dou, Y., Shu, K., Xia, C., Yu, P.S., Sun, L.: User preference-aware fake news detection. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2051–2055 (2021)

36. Ma, J., Gao, W., Wong, K.-F.: Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics (2018)
37. Bian, T., et al.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 549–556 (2020)
38. Song, C., Shu, K., Bin, W.: Temporally evolving graph neural network for fake news detection. Inf. Process. Manag. **58**(6), 102712 (2021)
39. Abdali, S.: Multi-modal misinformation detection: approaches, challenges and opportunities. arXiv preprint arXiv:2203.13883 (2022)
40. Alam, F., et al.: A survey on multimodal disinformation detection. arXiv preprint arXiv:2103.12541 (2021)
41. Singhal, S., Shah, R.R., Chakraborty, T., Kumaraguru, P., Satoh, S.: Spotfake: a multi-modal framework for fake news detection. In: 2019 IEEE fifth International Conference on Multimedia Big Data (BigMM), pp. 39–47. IEEE (2019)
42. Zhou, X., Wu, J., Zafarani, R.: SAFE: similarity-aware multi-modal fake news detection. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12085, pp. 354–367. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47436-2_27
43. Qian, S., Wang, J., Hu, J., Fang, Q., Xu, C.: Hierarchical multi-modal contextual attention network for fake news detection. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 153–162 (2021)
44. Lao, A., Shi, C., Yang, Y.: Rumor detection with field of linear and non-linear propagation. In: Proceedings of the Web Conference 2021, pp. 3178–3187 (2021)
45. Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Flammini, A., Menczer, F.: Detecting and tracking political abuse in social media. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 5, pp. 297–304 (2011)
46. Shao, C., Ciampaglia, G.L., Varol, O., Yang, K.-C., Flammini, A., Menczer, F.: The spread of low-credibility content by social bots. Nat. Commun. **9**(1), 1–9 (2018)
47. Subrahmanian, V.S., et al.: The Darpa twitter bot challenge. Computer **49**(6), 38–46 (2016)
48. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. Commun. ACM **59**(7), 96–104 (2016)
49. He, Z., Mokhberian, N., K. Lerman, N.: Infusing knowledge from Wikipedia to enhance stance detection. In: Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, pp. 71–77 (2022)
50. Yang, K.-C., Varol, O., Hui, P.-M., Menczer, F.: Scalable and generalizable social bot detection through data selection. In: Proceedings of the AAAI Conference on Artificial Intelligence vol. 34, pp. 1096–1103 (2020)
51. mengxiao2000. weibo_bot_finder. https://github.com/mengxiao2000/weibo_bot_finder (2023)
52. Zubiaga, A., Liakata, M., Procter, R.: Exploiting context for rumour detection in social media. In: Ciampaglia, G.L., Mashhadi, A., Yasseri, T. (eds.) SocInfo 2017. Exploiting context for rumour detection in social media, vol. 10539, pp. 109–123. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67217-5_8
53. Sun, T., Qian, Z., Dong, S., Li, P., Zhu, Q.: Rumor detection on social media with graph adversarial contrastive learning. In: Proceedings of the ACM Web Conference 2022, pp. 2789–2797 (2022)

54. Li, F., Liu, Z., Duan, J., Mao, X., Shi, H., Zhang, S.: Exploiting conversation-branch-tweet hypergraph structure to detect misinformation on social media. ACM Trans. Knowl. Discov. Data **18**(2), 1–20 (2023)
55. Sun, M., Zhang, X., Zheng, J., Ma, G.: DDGCN: dual dynamic graph convolutional networks for rumor detection on social media. In: Proceedings of the AAAI Conference on Artificial Intelligence, 36, pp. 4611–4619 (2022)
56. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
57. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)