JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, XXX-XXX (2024)
DOI: 10.6688/JISE.20240X_38(X).00XX

A Rumor Detection Model Based on Weighted Graph Convolutional Neural Network

XUEHAN+,LIN-LIN GU AND JIAN-YONG YU
School of Computer Science and Engineering
Hunan University of Science and Technology
Xiangtan, China
E-mail: xhan@hnust.edu.cn

More and more social media provide convenience for people to quickly publish and spread information including all kinds of rumors, which bring increased risk and harm. How to quickly and efficiently detect rumors on social networks has become an urgent task to be realized. The existing rumor detection methods ignore the variability of nodes in feature extraction. Based on the above problems, we propose a rumor detection model based on a weighted graph convolutional neural network called CR-GCN. First, we construct three heterogeneous graphs based on the rumor propagation structure: the heterogeneous graph of rumor propagation, the heterogeneous graph of rumor diffusion, and user social networks. The graph convolutional neural network can update the node representation by aggregating neighboring nodes based on the structural information, but GCN treats all neighboring nodes equally when aggregating. Therefore, we consider the heterogeneity of nodes based on GCN, use the similarity between user behavior and the sentiment difference between posted content as the weights of heterogeneous graphs, and use the weighted graph convolutional neural network to learn the dependencies on different nodes. To enhance the learning of important features, an attention mechanism is introduced into the model. Finally, experimental results of the public microblog dataset show that our proposed CR-GCN model can reach 91% accuracy in detecting rumors, which has higher rumor detection performance than other baseline methods.

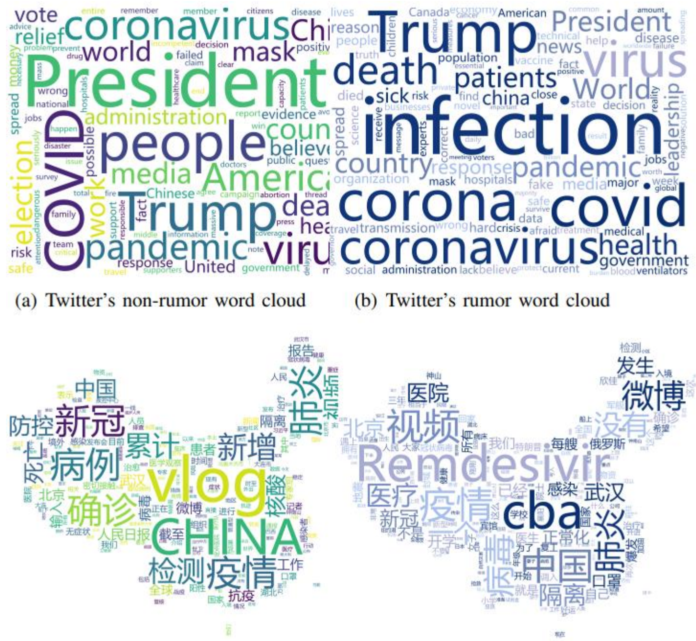
Keywords:rumor detection, weighted graph convolutional neural network, attention mechanism

1. INTRODUCTION

Nowadays, social networks play an important role in people's lives. The ease of sharing and high data transfer speeds make it easy for information to spread through social media, but the lack of effective monitoring and identification of rumor has caused great social unrest as rumors spread rapidly through social media. Science magazine even named the spread of rumors and misinformation related to the epidemic as the "Crash of the Year". The proliferation of rumors has become a more serious social problem than ever before, and timely detection of rumors has become a top priority. We used the Word-Cloud module to visualize the frequency of rumors and non-rumor texts related to COVID-19 on tweets and microblogs, as shown in Fig. 1.

Fig. 1 shows that the keyword gap between rumor and non-rumor on different social media platforms is very small, and rumor is packaged in such a way that it is difficult for users to distinguish whether the information is a rumor directly from the content based on

ReceivedApril10, 2024; revised; accepted.
Communicated by Editor.
+ Corresponding author.



(a) Twitter's non-rumor word cloud (b) Twitter's rumor word cloud

(c) Weibo's non-rumor word cloud (d) Weibo's rumor word cloud

Fig.1. The wordcloud of rumors and non-rumors

their own knowledge. In order to identify rumors on social media as possible, an increasing number of researchers are dedicated to exploringrumor recognition and detection. Traditional rumor detection methods extract various features of the textual content of rumor [1][2] and the user's social environment [3][4] and feed them into traditional machine learning classifiers such as random forests [5] and support vector machines [6]. Yet, they suffer from a fatal flaw in that these models rely too heavily on manually defined feature engineering and the resulting feature vectors aren't robust enough to handle complex and variable scenarios. Some recent studies have used deep learning models to extract higher-order features, such as recurrent neural networks [7], convolutional neural networks [4][8] and deep neural networks [9][10] to identify rumor. Although these methods can achieve good accuracy when dealing with textual data on defined features, they don't perform well in data on non-Euclidean structures.

To better extract the structure of rumor propagation and its textual information, several researchers have constructed propagation trees to identify rumors by means of response relationships between posts: using implicit links such as tags and web links [11]. Later improvements were made by using graph convolutional networks (GCN) to detect structural information about rumor propagation [12][13][14]. These GCN-based models do achieve better results in rumor detection, but we believe that the existing GCN-based models ignore the different contributions to nodes in the rumor propagation network. Clearly, different nodes have different degrees of interaction with each other, and counting these nodes indiscriminately reduces the accuracy of the models.

For this reason, we proposed a rumor detection model based on a weighted graph convolutional neural network (CR-GCN). The CR-GCN model leverages a weighted graph to represent rumor propagation, addressing node heterogeneity and complex dependencies. This approach captures varying levels of user credibility, engagement, and temporal dynamics, enhancing the model's accuracy. Compared to GCN or other rumor detection algorithms, CR-GCN integrates modeling of both rumor propagation and user social graphs, offering a more comprehensive view. Its attention mechanism strengthens global dependency learning, focusing on crucial aspects of the network. The weighted graph representation, validated through experiments on real datasets, proves superior in capturing essential features and nuances of rumor propagation. By integrating advanced techniques and utilizing weighted graphs, CR-GCN emerges as an effective and robust solution for rumor detection in complex social media landscapes. The main contributions to this paper can be summarized as follows.

• We take full account of node heterogeneity by using weighted graphs to describe the topology between nodes of a rumor propagation network.

• We use weighted graph convolutional neural networks to deal with the dependencies between parent and child nodes to further capture the complex features of the propagation structure.

• An attention mechanism is introduced in the model to further enhance the learning of global dependencies.

• We conducted a series of experiments on a real dataset, thus proving the validity of CR-GCN model.

This article is structured as follows. Section 2 reviews recent related work on rumor detection; Section 3 provides a detailed description of the rumor detection task. Section 4 describes our proposed model in detail. Section 5 presents experimental results and provides a detailed analysis. The conclusions of the paper are presented in Section 6.

2. THE RELATED WORK

Rumor detection is a classification task to determine the authenticity of information spread on social media [15]. Traditional rumor detection methods have focused on training classifiers based on various handcrafted features carefully extracted from the text content and the user's social environment, for example, extracting linguistic attributes, user features [3][4]. Pan et al. analyzed the generalization ability of rumors across domains [16]. Shi et al. [17] proposed a proxy-guided efficient re-sampling method (POGER) to improve black-box AI-generated text (AIGT) detection by estimating word generation probabilities as pseudo white-box features. In addition, Sicilia et al. [18] extracted more effective features, such as impact potential and network features. As previous models only modeled user information and textual content, ignoring the information about temporal changes, Ma et al. [19] proposed a DSTS model using timeseries modelling techniques to integrate various social contextual information in rumor detection. Wu and Ma et al. [6][20] detected rumors by comparing rumor trees with non-rumor trees. These methods rely heavily on manual feature design, which is time-consuming and laborious.

To better learn advanced features for rumor detection, some of them started to use deep learning methods such as recurrent neural networks [7], convolutional neural networks [4][8], and deep neural networks [10]. Sida et al. improved the accuracy of rumor detection by introducing an attention mechanism and Bi-GRU neural network [21]. Kaliya et al. [9] proposed a deep learning method based on the BERT, Fake-BERT, which improves long-range dependent feature detection. Jiang et al [22] conducted a preliminary study on augmenting news with auxiliary entity descriptions from knowledge as inputs for classifying fake news. In a related area of rumor detection, a recent study used just-in-time learning for zero-shift learning [23]. Yang et al. used a Crowd intelligence and ChatGPT-Assisted Network (CICAN) to improve rumor detection by exploiting large language models and heterogeneous attention mechanisms [24]. Li et al. explored the application of various deep learning models for rumor detection on social media platforms, highlighting both accuracy and efficiency improvements in terms of accuracy and efficiency [25].

However, these methods ignore the information-rich social graph topology, so Liu et al. proposed a method that combines user relevance and information propagation representation for rumor detection [26]. Ma et al [27] proposed a GAN-style approach to improve the performance of rumor classifier. Subsequently, Ma et al. [28] and Li et al. [29] used recurrent neural networks and graph convolutional networks to model rumor propagation trees to capture hidden feature representations, respectively.

Compared to other deep learning models, GCN is better at capturing global structural features in graphs. Therefore, Emrah Inan [13] used graph attention networks to filter potential fake users. Based on the existing literature, Ma et al [14] considered the interaction between events and built a multilayer graph convolution model to obtain multilayer features for rumor detection. Cui et al. [30] proposed a novel adaptive graph comparison learning method for rumor detection, showing that rumor propagation trees usually have a broad structure and most of the nodes are flat. Xiang et al. [31] proposed a graph autoencoder-based rumor detection model that achieves a comprehensive understanding of rumor propagation by capturing both local semantic changes and global semantic evolution information of events during the propagation process. Zhang et al. [32] proposed a heterogeneous subgraph converter-based approach, which uses graph convolutional networks to capture news, user behavior and social relationships, and other multi-source heterogeneous information. Natali et al. [33] proposed a model called CSI to extract rumor features. Bian et al. [12] mined structural features of rumor tree propagation and diffusion using Bi-GCN. Wu et al. [34] introduced an attention mechanism that dynamically adjusts the weight of each node to further improve detection performance.

In summary, existing rumor detection methods don't take into account the heterogeneity of nodes, we consider the heterogeneity of nodes on the basis of graph convolutional neural networks to better detect the authenticity of information.

3. PROBLEM DESCRIPTION

Our main task is to make more accurate judgments about the veracity of information. To describe the problem more clearly, we have described the rumor detection task as follows. Some people deliberately create and spread false information for the purpose of gaining attention or creating confusion [28], thus causing rumors to proliferate. The spread of events is shown in Fig. 2.

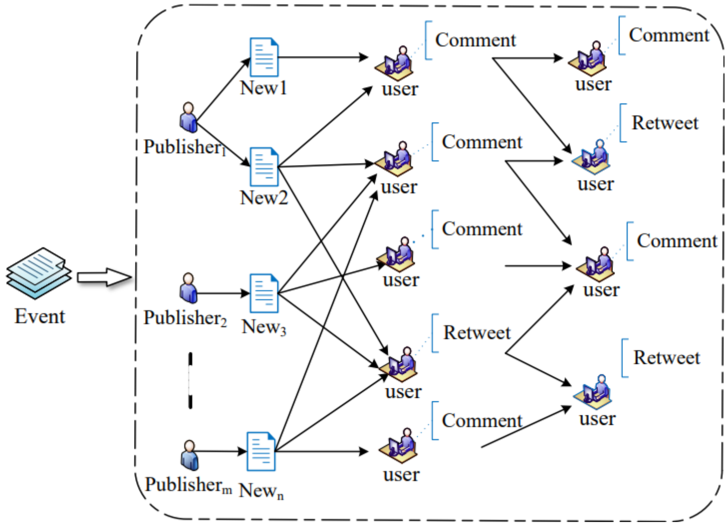


Fig. 2. Rumor propagation tree

From Figure 2, we can find that events are often composed of a group of related posts. Therefore, we denote rumor events as$E=\{E_1,E_2,\ldots,E_N\}$, where E_i is the i-th event and N is the total number of events.Each event contains therelated post set $R_i$and user set U_i,i.e. $E_i=\{R_i,U_i\}$.Each rumor post set $R_i$is composed of a source blog r_0^i and n_i-1 retweets or comments on the blog, denoted as $R_i=\{r_0,r_1,\ldots,r_{n_i-1}\}$, where $r_j(j\in[1,n_i-1])$ denotes the j-th retweet or comment on the source blog $r_0$in R_i. The user set $U_i$is the information about the users corresponding to each blog, denoted as $U_i=\{u_0,u_1,\ldots,u_{n_i-1}\}$,$u_j$ denotes the information related to the$j$-th user involved in the retweet.

Each event E_i corresponds to a category label $y_i\in\{F,T\}$(rumor and non-rumor). Rumor detection aims to learn the mapping $f$from the rumor event E_i to the label and thus determine whether the event isrumor. The mapping function can be expressed as

$$f\colon E_i\to y_i \tag{1}$$

4. THE PROPOSED MODEL

The general framework of the rumor detection model based on weighted graph convolutional neural networks is shown in Fig. 3.

Our model can be divided into three parts.

(A) Structural feature extraction module.We used the directed weighted graphs to reflect the topological relationship of information dissemination in social networks. Users in the social network correspond to nodes in the weighted graph, while the sentiment difference of posted content between neighboring nodes can be expressed as the weight of connected edges. Finally, a weighted graph convolutional neural network is used to extract the structural features of this weighted graph.

(B) User feature extraction module. In addition to rumor structure features, we also consider the role of user features for rumor detection. The combination of structure features and user features makes the results of rumor detection more accurate.

(C) Classifier. After feature extraction, we introduced attention mechanisms to enhance contextual semantic information and global dependencies. The specific implementation of these components is detailed in the following sections.

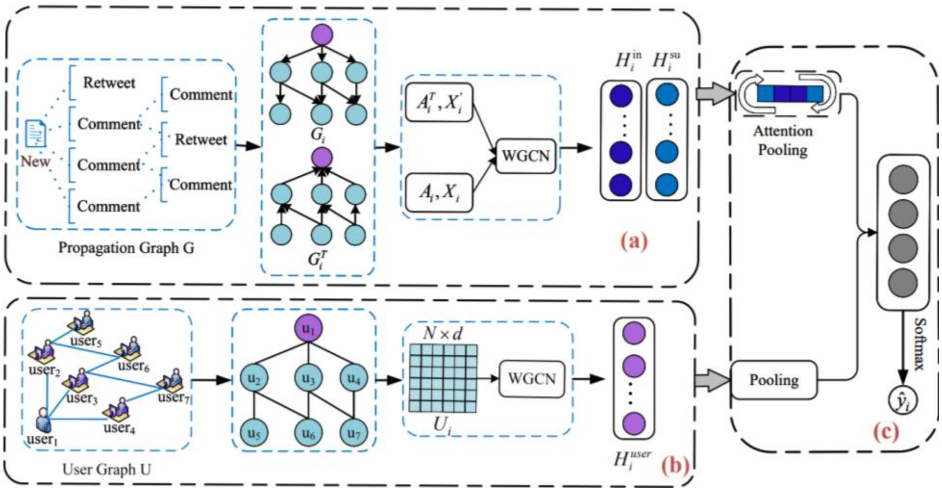


Fig. 3. The overall framework of the CR-GCN model. (a) structure feature extraction module;(b) user feature extraction module; (c) rumor detection classifier.

4.1 Weighted Graph Convolutional Neural Networks

The basic idea of the GCN is to update its own node representation by aggregating information on neighboring nodes. GCN tended to treat all neighboring nodes equally.However,since users have different cognitive scopes and interests, different neighboring nodes have different degrees of influence on a particular node. Thus, we considered the heterogeneity of nodes based on the GCN, and the specific implementation of the weighted graph convolutional neural network (WGCN) is as follows.

The interactive behavior of information is represented as a heterogeneous weighted graph $\mathrm{G} =< V, E, W >$, where Wdenotes the strength of correlation between nodes.The adjacency matrix $\mathrm{A} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$is constructed based on the connectivity of the graph G. The WGCN is used to aggregate the features of neighboring nodes to update the vector representation of the nodes. To avoid overfitting and over-smoothing problems, we refer the DropEdge method [35] to randomly deactivate the edges in the graph.Given the adjacency matrix A and the deactivation probability η, $\mathrm{N_e}$ is the number of edges in the graph, and $\eta \times \mathrm{N_e}$ edges are randomly sampled in the set of edges of the given graph to form $\mathrm{A_{drop}}$, and the processed adjacency matrix A' is

$$\mathrm{A}' = \mathrm{A} - A_{\mathrm{drop}} \tag{2}$$

To better understand the forward propagation law of WGCN, we represented it as a vector form. Thus, for node i, we have

$$\tilde{h}_i^{(l+1)} = \sigma\left(\left(\sum_{j\in N(i)} a_{ij} \frac{1}{\sqrt{d_{ii} d_{jj}}} h_j^{(l)}\right) W^{(l)}\right) \tag{3}$$

$$h_i^{(l+1)} = \tilde{h}_i^{(l+1)} + h_i^{(l)} \tag{4}$$

where a_{ij} is the element corresponding to the adjacency matrix, $N(i)$ denotes the set of neighbors of node i, $\sigma(\bullet)$ is the activation function, $W^{(l)}$ is the weight matrix, $\tilde{h}_i^{(l+1)}$ is the node feature representation denoting the output of layer l, $h_i^{(l+1)}$ is the input of layer $l+1$. d_{ii} denotes the sum of the weights of the neighboring nodes directly connected to node i.

4.2 Structural feature extraction module

We constructed both a rumor propagation graph G_i and arumor diffusion graph G_i^T based on the content and propagation structure data of a given blog, taking full account of the depth and breadth of rumor propagation.

Take the rumor propagation graph $G_i =< V, E, W_i >$ as an example. We represent each blog post as a node to form node sets of the propagation graph, $V = \{r_0, r_1, \ldots, r_{n_i-1}\}$. Based on the existence of comments or retweets between blog posts, the set of edges that form the propagation graph is $E_i = \{e_{st} | s, t = 0,1,2, \ldots, n_i - 1\}$.

When we cannot judge the authenticity of a blog post, we often make a choice based on the content of the comments since the content posted by users contains their preferences and opinions. We used the snownlp method to perform sentiment analysis on the content posted by each node, quantifying the degree of support of the responding node for its parent node's views of textual sentiment differences between nodes. Assuming that the sentiment score of blog post r_s is S_s and the sentiment score of responding node r_t is S_t, the support of r_t to r_s is expressed as

$$w_{st} = 1 - |S_s - S_t| \tag{5}$$

Taking the support as the weight of the edge e_{st}, the set of weights can be expressed as $W_i = \{w_{st} | s, t = 0,1,2, \ldots, n_i - 1\}$. The adjacency matrix A_s is constructed according to the connection relationship between the upper and lower nodes of the propagation graph, as shown in (6).

$$A_s = \begin{cases} w_{st} & \text{if } e_{st}^i \in E_i, \\ 0 & \text{others.} \end{cases} \tag{6}$$

We used the BERT model to preprocess the text of event R_i to capture a more accurate representation of the text features. We used the d-dimensional feature vector extracted by the BERT model as the feature matrix X_i for this event. Let $X_i = [x_0, x_1, \ldots, x_{n_i-1}]^T \in \mathbb{R}^{n_i \times d}$, where x_0 is the feature vector of the source blog r_0 and x_j is the feature vector of the j-th retweet or comment blog r_j.

The feature matrix and adjacency matrix are used as the input of the graph convolution to fuse the neighbor information of different nodes so that the neighborhood features under different dependencies can be learned. Finally, we obtained the feature representations $H^{in} = h_{in}^{(l+1)}$ and $H^{su} = h_{su}^{(l+1)}$ of each node in the rumor propagation graph G_i and the rumor diffusion graph G_i^T, respectively.

4.3 User feature extraction module

When users receive rumors, they can express their own opinions and may even become new rumor spreaders. Rumor propagation and users are mutually influential and inextricably linked. We introduced a user feature extraction module to further analyze the impact of user interaction information on rumor detection.

We construct the user social network $G_{U_i} =< U, E, W_u >$based on the propagation structure and user information, where $E = \{e_{st} | s, t = 0,1,2, \ldots, n_i - 1\}$,$e_{ij}$ denotes the interaction between user $i$and j. $U = \{u_0, u_1, \ldots, u_{n_i-1}\}$ is the users set, each user corresponds to a user vector $q_j \epsilon \mathbb{R}^{d_u}$. We considered 8 user features in terms of 2 aspects, background information and social information, which are described in detail in Table1 and Table 2.

Table 1. User Profiling - Background Information

Features	Description
Username (j_0)	Nickname of the user, unique identification
Certification degree (j_1)	Official certification of Weibo
IP Location (j_2)	Region of the user
Length of accession (j_3)	Time since users joined Weibo

Table 2. User Profiling - Social Information

Features	Description
Credit value (j_4)	User credit value after integrated consideration
Followers' number (j_5)	Number of users following others
Fans' number (j_6)	Number of followers of the user
Blogs'number (j_7)	Total number of blog made by users

Each user vector contains the above eight features, denoted as $q_j = [q_{j_0}, q_{j_1}, q_{j_2}, q_{j_3}, q_{j_4}, q_{j_5}, q_{j_6}, q_{j_7}]$. The feature vectors of all users involved in rumor spread are stitched together to form the user's feature matrix $X_u = [q_0, q_1, q_2, \ldots, q_{n_i-1}]^{\mathrm{T}} \in \mathbb{R}^{n_i \times d_u}$.

The stronger the relationship between users is, the greater the influence of friend's behavior on the user is. We defined a similarity-based approach to measure the social relationships between users, as shown in (7).

$$sim(\mathrm{u}_i, \mathrm{u}_\mathrm{j}) = \frac{\mathrm{F}_i \cap \mathrm{F}_\mathrm{j}}{\mathrm{F}_i \cup \mathrm{F}_\mathrm{j}} \tag{7}$$

where $sim(\cdot)$ denotes the friend similarity between users and F_i denotes the i-th user's friend sets.

To take into account the difference in the influence of different nodes, we use node similarity as the weight of connected edges, then the adjacency matrix A_u of the user's social network can be expressed as

$$A_u = \begin{cases} sim(\mathrm{u}_i, \mathrm{u}_\mathrm{j}) & sim(\mathrm{u}_i, \mathrm{u}_\mathrm{j}) > e. \\ 0 & \text{Others.} \end{cases} \tag{8}$$

where e is the similarity threshold. If the similarity between two nodes is over the threshold, the connection is established, and the similarity is used as the weight value of the connected edges.

We use WGCN to fuse neighboring nodes with different degrees of similarity to obtain a feature representation of the user node as $H^{user} = h_u^{(l+1)}$.

4.4 Classifier

To avoid features being smoothed, we use the attention mechanism as the body of the pooling layer to capture the influence between nodes, and the attention weights are calculated as shown in (9) and (10).

$$t_j = \tanh(Q \cdot W)^T \cdot H_j \tag{9}$$

$$\alpha_j = \frac{\exp(t_j)}{\sum_{i=1}^{|t|} \exp(t_i)} \tag{10}$$

where t_j is the attention score, H_j is the hidden feature vector representation of j, α_j is the attention weight, and $|t|$ denotes the total number of features.

The feature vector representation and the corresponding attention weights α_j are combined to obtain the final attention feature vector.

$$\widehat{H} = \sum_{i=1}^{|t|} \alpha_i \cdot H_i \tag{11}$$

We compute the feature vectors H^{in} and H^{su} to obtain the attention feature vectors $\widehat{H}^{in}$ and $\widehat{H}^{su}$. This method can retain more important features than traditional methods.

The different features are concatenated into the final feature matrix H_c.

$$H_c = \mathrm{concat}\left(\widehat{H}^{in}, \widehat{H}^{su}, \widehat{H}^{user}\right) \tag{12}$$

The final feature matrix H_c is fed into the SoftMax function to predict the category to which the rumor event E_i belongs.

$$y_i = \mathrm{softmax}(W_c H_c + b) \tag{13}$$

where W_c and b are the parameters to be learned. We train to optimize all parameters in this model by minimizing the cross-entropy of the predicted and true labels for all events.

5. EXPERIMENTAL ANALYSES

To demonstrate the effectiveness of our proposed CR-GCN model for rumor detection, a series of experimental analyses of the CR-GCN model are conducted in this section.

5.1 Dataset& Experimental Setup

Most existing models related to rumor detection use the Twitter15 and Twitter16 datasets, but these datasets are old and contain fewer feature categories, so this paper draws on the CHECKED dataset from 2021 [36], this dataset contains Weibo data related to COVID-19 from December 2019 to August 2020, which contains a total of 1760 non-rumor and 344 rumor data. Due to the large difference in the amount of data between the two types, poor generalization ability and overfitting problems may occur during model training. In order to avoid the above problems, the dataset is extended and collected by the crawler. For the detailed statistics of the dataset, please refer to Table 3.

Table 3. Dataset Statistics

Statistics	Rumor	Non-rumor	Total
Number of blogs	755	344	1099
Number of retweets	1605337	40 054	1645391
Number of comments	1 169 246	16 456	1 185 702

To reduce the chance of the experimental results, we performed a 5-fold cross-validation on the dataset by randomly dividing it into five mutually exclusive subsets of similar size, randomly selecting four of them as the training set and the remaining subset as the test set, and using the average of the metrics obtained from the experiments as the detection results of the model.

By fine-tuning the pre-trained BERT model to ensure that the input of the model is adjusted and optimized, the text information of the blog post is extracted, and the feature dimension of the text is set to 1200 [9]. The model is trained on a two-layer weighted graph convolutional network model, with the dimension of the hidden feature vector of nodes in each layer being 64 and the dropout of each layer being 0.2 [12]. The Adam algorithm is used for model training, the number of iterations is 200 epochs, and early stopping is applied when the validation loss stops decreasing by 10 epochs.

5.2 Baseline models & Evaluation indicators

To test the performance of our model, we compared with existing state-of-the-art methods.

• SVM-TK [20]: A support vector machine classifier based on the kernel function of a propagation tree simulating rumor propagation to extract structural features.

• RvNN [28]: A recurrent neural network model modelling the bottom-up and top-down propagation directions, respectively, and learning the feature vector representation of rumors.

• PPC_RNN [3]: A rumor detection model that combines RNN and CNN to learn the rumor representation by combining user features along the rumor propagation path.

• CSI [33]: An LSTM classifier that extracts features such as text content, user response and source user from the propagation graph.

• Bi-GCN [12]: A bi-directional graph convolutional network model based on the propagation direction and diffusion direction of the propagation tree, transforming the rumor classification task into a graph classification task.

• GCAN [37]: A co-attention network that detects true and false rumors based on the content of the source tweet and its propagation-based users.

We used the same evaluation metrics as in previous work, i.e. accuracy, precision, recall and $F1$ score, to achieve an assessment of the performance of the rumor detection model.

5.3 Analysis of experimental results

To test whether the CR-GCN model could achieve better rumor detection performance, we compared it with the baseline approach and the performance of all models on the dataset is shown in Table 4.

Table 4. Experimental Results of the Models

Models	Acc	TR			FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1
SVM-TK	0.759	0.686	0.859	0.764	0.847	0.639	0.730
RvNN	0.838	0.824	0.861	0.851	0. 842	0.822	0.831
PPC_RNN	0.847	0.829	0.875	0.851	0.868	0.820	0.847
CSI	0.861	0.862	0.866	0.872	0.866	0.872	0.869
Bi-GCN	0.899	0.866	0.892	0.904	0.876	0.871	0.894
GCAN	0.880	0.871	0.861	0.885	0.866	0.889	0.896
CR-GCN	0.918	0.904	0.941	0.931	0.922	0.902	0.911

From the results shown in Table IV, we can see that the traditional machine learning-based method (SVM-TK) performs significantly lower than other methods on the dataset, with a more limited detection capability, and cannot better identify rumors in social networks. While propagation tree-based methods (RvNN, PPC_RNN) get some improvement in accuracy compared to machine learning methods. These methods also show that the propagation structure contains a lot of important information about rumors and capturing the propagation structure helps the rumor detection task. CSI performs slightly better than PPC RNN because it scores user behavior while integrating user features, while Bi-GCN investigates both rumor propagation and diffusion, compared to GCAN which models structural information from user similarity matrices rather than propagation networks. According to the results, Bi-GCN performs much better than GCAN because it takes into account the comment information.

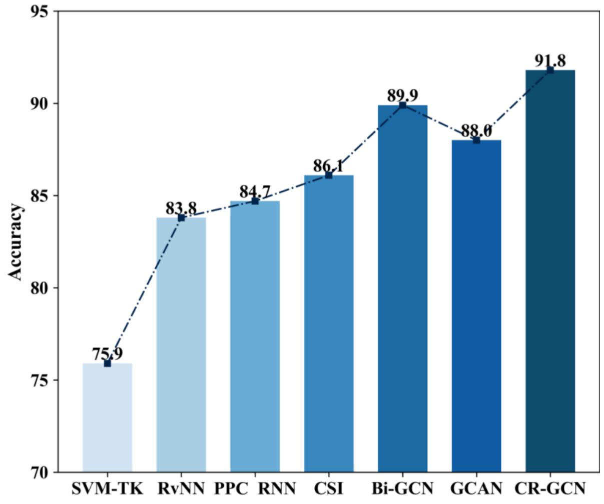


Fig. 4. Comparison results of the models

Figure 4 shows more visually the performance of the models on different evaluation metrics. From Fig. 4, we can find that each model gradually gets some improvement in rumor detection, and our proposed model can mostly outperform the other comparison methods. In terms of accuracy, our model outperforms SVM-TK, RvNN, PPC_RNN, CSI, Bi-GCN and GCAN by 16%, 8%, 7%, 5%, 2% and 3% respectively on the dataset. The F1 values for rumor samples and non-rumor samples were 3% and 2% higher than Bi-GCN, respectively. In addition, the increase of F1 values indicates the robustness of the CR-GCN model, which can reduce the misclassification rate of rumor events. Obviously, our model can improve the ability to classify rumor information.

5.4 Ablation experiments

In this section, we conduct a series of ablation experiments to visually illustrate the effectiveness of the components of our model. This subsection discusses the following aspects.

(A) With or without bi-directional propagation graph. In the rumor feature extraction module, we compared the proposed method with top-down and bottom-up variants of the structure, and fully considered the effectiveness of the features of rumor propagation and diffusion structures for rumor classification.

(B) Whether to consider node heterogeneity. We analyzed the effectiveness of rumor detection by removing weights and considering the degree of influence of different nodes.

Ablation experiments were conducted for the effects of node heterogeneity and propagation direction on rumor detection performance, and the experimental results are shown in Fig. 5.

We compared the GCN methods with bidirectional propagation, considering only rumor propagation (UD-GCN), and considering only rumor diffusion (BU-GCN), respectively. The bi-directional graph convolution method (Bi-GCN) significantly outperformed the other two groups of methods, with an improvement of up to 5% in accuracy

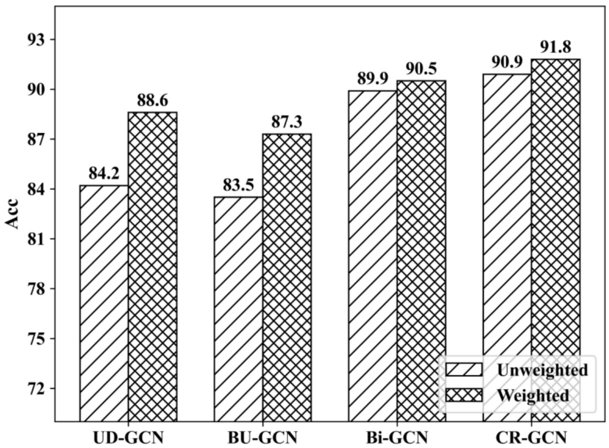


Fig. 5. Results of weighted heterogeneous graph ablation experiments

rate. This indicates that the use of directed propagation graph can better extract the features of rumor propagation, while the extraction and fusion of features from both aspects of rumor propagation and spread can further improve the effectiveness of rumor detection.

Subsequently, we investigated the effect of node variability on rumor detection performance. The experimental results show that the accuracy of this model decreases by about 1% when node discrepancy is ignored. In addition, the impact of node discrepancy on each model was considered on the basis of the above ablation experiments, and the accuracy was improved to different degrees. Among them, Bi-GCN can increase the accuracy of rumor detection to 90.5% when node variability is considered. This proves the effectiveness of considering node variability for rumor detection methods.

5.5 Early detection of rumors

Early detection of rumors is an important goal of rumor detection. The longer a rumor is posted on social media, the more far-reaching its impact becomes and the more difficult it is to intervene it. To verify the performance of our proposed model in early rumor detection, we evaluated the early rumor detection capability of the proposed method in this paper and other baseline methods by two delay strategies, limiting the number of retweets of the source blog post and its posting elapsed time, respectively. The experimental results are shown in Fig. 6 and Fig. 7.

The performance of the models was evaluated by controlling the number of user retweets since the source blog post was published and calculating the accuracy of rumor detection and the experimental results are shown in Fig. 6, with the horizontal axis representing the number of blog posts retweeted by users. With the increasing number of retweets, the performance of different rumor detection models is improved to some extent. CR-GCN and Bi-GCN achieve high accuracy very early after the initial broadcast by the rumor source node. This indicates that graph convolutional networks can effectively aggregate information about neighboring nodes and learn accurate node feature vector representations for improving the model's ability to detect rumors at an early stage. The performance of CR-GCN outperforms that of Bi-GCN, which can be attributed to the fact that CR-GCN can effectively aggregate and learn representations of neighboring

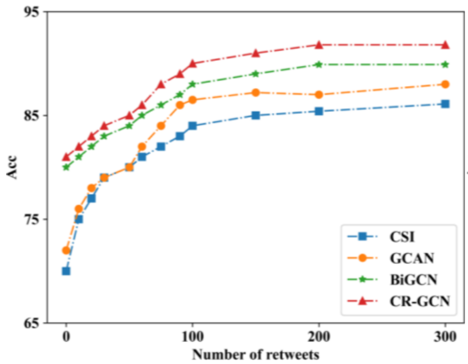


Fig. 6. Result change with number of retweets

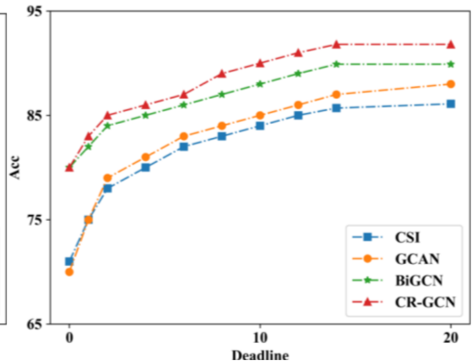


Fig. 7. Result change with deadline

nodes by taking into account the different intra-layer dependencies formed between different blog posts, thus enabling achieve accurate identification of rumors in the propagation network.

The performance of different rumor detection models is judged by controlling the propagation duration of rumors after they are published, as shown in Fig. 7. Fig. 7 shows the accuracy of rumor detection corresponding to different cutoff times. We can find that the detection effect of controlling the propagation duration has basically the same trend as the detection effect of controlling the number of user retweets. With the passage of time, the rumor detection methods are both gradually improved to a certain extent, which also indicates that as more and more users perform the propagation of this message with the growth of time, the connectivity of the propagation graph is subsequently enhanced, making the propagation structure further strengthened. Thus, compared with other baseline models, the graph-based rumor detection model can capture a more comprehensive structural feature of rumor propagation and has better detection performance. This again demonstrates the importance of the interaction between posts for the rumor detection task and the effectiveness of the model.

6. CONCLUSIONS

To address the current challenges in rumor detection, this paper proposes a rumor detection model based on a weighted graph convolutional neural network. The method models the dependency relationships formed between parent and child nodes in the propagation tree through a weighted graph convolutional network to capture the complex features of the propagation structure. Meanwhile, the feature vector representation of nodes in the propagation tree is enhanced using an attention mechanism to learn the rumor feature vector representation more accurately. In this paper, we used the microblogging dataset related to the COVID-19 as experimental data, and the proposed method in this paper has higher rumor detection performance than other baseline methods and also has good detection effect in the early propagation stage of rumors. We will consider enhancing the detection of rumors using multimodal information in future work.

ACKNOWLEDGMENT

This work was supported by The National Social Science Fund of China (No. 20BXW096).

REFERENCES

[1] Kwon, Sejeong, et al. "Prominent features of rumor propagation in online social media." *2013 IEEE 13th international conference on data mining*. IEEE, 2013.

[2] Jieun Shin, Lian Jian, Kevin Driscoll, and Francois Bar, "The diffusion of misinformation on social media: Temporal pattern, message, and source." *Computers in Human Behavior*, Vol. 83, 2018, pp. 278–287.

[3] Liang, Gang, et al. "Rumor identification in microblogging systems based on users' behavior." *IEEE Transactions on Computational Social Systems* 2.3 (2015): 99-108.
[4] Soroush Vosoughi, Mostafa Mohsenvand, and Deb Roy, "Rumor gauge: Predicting the veracity of rumors on twitter," *ACM Trans. Knowl. Discov. Data*,Vol. 11, no. 4, 2017, pp. 50:1--50:36.
[5] Kwon, Sejeong, Meeyoung Cha, and Kyomin Jung. "Rumor detection over varying time windows." *PloS one* 12.1 (2017): e0168344.
[6] Ke Wu, Song Yang, and Kenny Q Zhu, "False rumors detection on sinaweibo by propagation structures," in *2015 IEEE 31st international conference on data engineering*, 2015, pp. 651–662.
[7] Ma, Jing, et al. "Detecting rumors from microblogs with recurrent neural networks." in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 3818–3824.
[8] Somya Ranjan Sahoo and Brij B Gupta, "Multiple features-based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, Vol. 100, 2021, pp. 106983.
[9] Kaliyar, Rohit Kumar, Anurag Goswami, and Pratik Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." *Multimedia tools and applications* 80.8 (2021): 11765-11788.
[10] Kaliyar, Rohit Kumar, et al. "FNDNet–a deep convolutional neural network for fake news detection." *Cognitive Systems Research* 61 (2020): 32-44.
[11] Justin Sampson, Fred Morstatter, Liang Wu, and Huan Liu, "Leveraging the implicit structure within social media for emergent rumor detection," In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 2377–2382.
[12] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang, "Rumor detection on social media with bi-directional graph convolutional networks." in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34, 2020, pp. 549–556.
[13] Inan, Emrah. "ZoKa: a fake news detection method using edge-weighted graph attention network with transfer models." *Neural Computing and Applications* 34.14 (2022): 11669-11677.
[14] Yuanyuan Ma, Shouzhi Xu, and Fangmin Dong, "A multilevel graph convolution neural network model for rumor detection." In *IEEE 2nd International Conference on Power, Electronics and Computer Applications* (*ICPECA*), 2022, pp. 1225–1229.
[15] Juan Cao, Junbo Guo, Xirong Li, Zhiwei Jin, Han Guo, and Jintao Li, "Automatic rumor detection on microblogs: A survey." *arXiv preprint arXiv:1807.03505*, 2018.
[16] Pan, Liangming, Yunxiang Zhang, and Min-Yen Kan. "Investigating zero-and few-shot generalization in fact verification." *arXiv preprint arXiv: 2309.09444* (2023).
[17] Shi, Yuhui, et al. "Ten Words Only Still Help: Improving Black-Box AI-Generated Text Detection via Proxy-Guided Efficient Re-Sampling." *arXiv preprint arXiv: 2402.09199* (2024).

[18] Rosa Sicilia, Stella Lo Giudice, Yulong Pei, Mykola Pechenizkiy, and Paolo Soda, "Twitter rumor detection in the health domain." *Expert Systems with Applications*, Vol. 110, 2018, pp. 33–40.
[19] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong, "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*, 2015, pp. 1751– 1754.
[20] Jing Ma, Wei Gao, and Kam-Fai Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proceedings of ACL*, 2017.
[21] Yuan, Sida, and Qiong Yang. "Network Rumor Detection Using Attention Mechanism and BiGRU Neural Network in Big Data Environment." *Journal of Circuits, Systems and Computers* 33.01 (2024): 2450009.
[22] Jiang, Gongyao, et al. "Fake news detection via knowledgeable prompt learning." *Information Processing & Management* 59.5 (2022): 103029.
[23] Lin, Hongzhan, et al. "Zero-shot rumor detection with propagation structure via prompt learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 4. 2023.
[24] Yang, Chang, et al. "Rumor detection on social media with crowd intelligence and chatgpt-assisted networks." *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.
[25] Tan, Li, et al. "Research status of deep learning methods for rumor detection." *Multimedia Tools and Application*. 82.2 (2023): 2941-2982.
[26] Liu, Tianrui, et al. "Rumor Detection with a novel graph neural network approach." *arXiv preprint arXiv: 2403.16206* (2024).
[27] Jing Ma, Wei Gao, and KamFai Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning." In *The world wide Web conference*, 2019, pp. 3049–3055.
[28] Jing Ma, Wei Gao, and Kam-Fai Wong, "Rumor detection on twitter with tree-structured recursive neural networks." in *Proceedings of Association for Computational Linguistics*, 2018.
[29] Lu, Yi-Ju, and Cheng-Te Li. "GCAN: Graph-aware co-attention networks for explainable fake news detection on social media." *arXiv preprint arXiv: 2004.11648* (2020).
[30] Cui, Chaoqun, and Caiyan Jia. "Propagation Tree Is Not Deep: Adaptive Graph Contrastive Learning Approach for Rumor Detection." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. No. 1. 2024.
[31] Tao Xiang, et al. "Semantic Evolvement Enhanced Graph Autoencoder for Rumor Detection." *Proceedings of the ACM on Web Conference 2024*. 2024.
[32] Zhang, Yuchen, et al. "Heterogeneous Subgraph Transformer for Fake News Detection." *Proceedings of the ACM on Web Conference 2024*. 2024.
[33] Natali Ruchansky, Sungyong Seo, and Yan Li, "Csi: A hybrid deep model for fake news detection." In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 797–806.
[34] Zhiyuan Wu, Dechang Pi, JunfuChen, Meng Xie, and Jianjun Cao, "Rumor detection based on propagation graph neural network with attention mechanism." *Expert systems with applications*, Vol. 158, 2020, pp. 113595.

[35] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification." in *International Conference on Learning Representations*, 2019.
[36] Chen Yang, Xinyi Zhou, and Reza Zafarani, "Checked: Chinese covid-19 fake news dataset." *Social Network Analysis and Mining*, Vol. 11(1), 2021, pp. 1–8.
[37] Liu, Yang, and Yi-Fang Wu. "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. No. 1. 2018.

Xue Hanreceived her Bachelor's degree and Master's degree in computer science from Hunan University of Science and Technology, Xiangtan, China, in 2002 and in 2005, respectively. She is currently an associate professor in School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. Her research interests include complex system, multi-agent systems, distributed computing and social intelligence.

Lin-Lin Gu received the B.S. degree in computer science and technology from the Hunan University of Science and Technology, Xiangtan, China,where she is currently pursuing the master’s degree in electronic information. Her research interestsinclude social networks, artificial intelligence, cooperative computing.

Jianyong Yu received his Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013. He is currently an associate professor in School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include multiplex networks, information propagation, and artificial intelligence.