Recurrent Graph Convolutional Network for Rumor Detection

Song Wu, Hailing Xiong*, Ye Yang, Jinming Zhang, Chenwei Lin
College of Computer and Information Science, Southwest University, Chongqing, China
wusong5543@gmail.com

***Abstract*—The development of social media has provided an ideal platform for sharing information, but it has also become a hotbed for rumor posting and spreading. Existing rumor detection methods mainly look for clues from the textual content, static propagation structures. However, existing studies do not make full use of user information while ignoring the dynamic change of the communication structure. Therefore, this paper proposes a rumor detection model based on dynamic propagation structure called Recurrent Graph Convolutional Network (Re-GCN). In terms of rumor content representation, both the textual content of rumors and user information are considered. In terms of propagation structure, the dynamic propagation structure of rumors is considered. A dynamically changing propagation structure representation is constructed by dividing rumor propagation into multiple stages, and the dynamically changing propagation features are captured using a Bi-directional Recurrent Neural Network (Bi-RNN) and Graph Convolutional Networks (GCN). Finally, the attention mechanism is introduced to fuse the information of each stage to classify rumors. Experimental results on two real-world datasets show that the proposed method has significant improvement compared to the state-of-the-art benchmark methods.**

Keywords-rumor detection; graph convolutional network; dynamic propagation structure; social media

I. Introduction

Mobile Internet has become an integral part of people's lives. The openness and convenience of social media platforms such as Twitter, Facebook, and Weibo have lowered the threshold of information exchange, but it has also become a hotbed for rumors to spread. An official report released by Weibo shows that the Weibo platform effectively dealt with 76,107 rumors in 2020[1]. Rumors seriously mislead people's minds and even cause substantial economic losses, and endanger public safety. Therefore, there is an urgent need for fast and effective methods to detect social media rumors in response to the potential threats that rumors may pose.

Most early rumor detection methods mainly used manually extracted features such as text content[2] [3], user features[2], and propagation methods [4] [5] to train supervised classifiers, e.g., decision trees[2], support vector machine (SVM)[6]. Some studies have mined more effective features, such as temporal structure features[7], sentiment attitude of posts[8]. However, such methods rely heavily on feature engineering, which is time-consuming and suffers from insufficient robustness.

Recent studies have mainly employed deep learning methods to fully exploit the deep features of rumors by learning high-level representations of rumor texts through various neural networks, such as RNN and Gated Recurrent Unit(GRU)[10], to detect rumors. Some studies also considered the propagation structure of rumors and constructed top-down propagation trees and bottom-up propagation trees based on the retweet-reply relationships among posts in social media, as shown in Fig.1. The node "1" represents the source post of the rumor, and the other nodes represent the retweeted messages related to the source post. Then, the structural features of rumor propagation are captured by Recursive Neural Networks(RvNN)[11] and GCN[12][15].

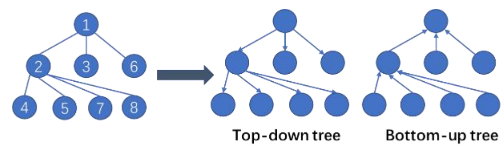

Figure 1. A bottom-up/top-down propagation tree. Node "1" represents the source post, i.e., the earliest post, and the other nodes represent the re-posted comment messages related to the source post. The number in the node indicates the order in which the messages are posted, and the larger the number, the later the message is posted.

Although the above studies have achieved good results in rumor detection tasks, there are still two shortcomings. The first is in the representation of the content of the rumor, these methods mainly consider rumor text content and ignore the user information contained in social media, which proved to be an essential clue for rumor detection in earlier studies[2]. The second is in the representation of the structure of rumor propagation, they consider the static structure of rumor propagation and ignore the change of rumor propagation structure over time, i.e., the dynamic propagation structure. Taking Fig.2 as an example, the global propagation structures of the propagation trees (a) and (b) are the same, but there are differences in their propagation structures when different cutoff times are chosen.

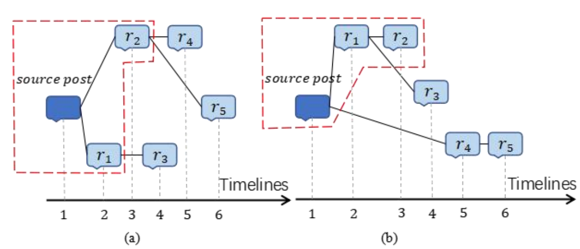

Figure 2. When the selected cutoff time is "6", which is the case of global propagation structure, the propagation trees (a) and (b) have the same structure. However, in the case of not global propagation structure, for example, when the cutoff time is "3", the structure of (a) can be described as "$s(source\ post) \rightarrow r_1, s \rightarrow r_2$" and the structure of (b) can be described as "$s \rightarrow r_1 \rightarrow r_2$", which are different.

DOI reference number: 10.18293/SEKE2022-063

To address the above issues, our research is concerned with (1) Considering both textual content and user information in the rumor detection task. (2) modeling the dynamic propagation structure of rumors and applying it to rumor detection. In this paper, we propose a novel Recurrent Graph Convolutional Network (Re-GCN) for rumor detection. First, we consider both textual content and user information during rumor propagation and extract features from them as the content representation of rumors. Second, we constructed a representation of the dynamic propagation structure by dividing the propagation tree into multiple stages at the same number interval, and Fig.3 shows an example of dividing a rumor propagation tree into 4 stages. We also propose the Re-GCN model to model the dynamic propagation structure and classify the rumors. The main contributions of this work are as follows.

- We consider both textual content and user information in social media, and encode them separately using different representations.
- We analyze and introduce the dynamic propagation structure of rumors. A multi-stage propagation structure representation method is proposed and a Recurrent Graph Convolutional Network (Re-GCN) is designed to model the dynamic propagation structure.
- Experiments on the Weibo and PHEME datasets show that our model improves significantly on the rumor detection task compared to some state-of-the-art models.

The rest of this paper is organized as follows. In Section II, we describe the related work on rumor detection. The problem statement is discussed in Section III. In Section IV, we describe the details of the proposed Re-GCN model. In Section V, we present the experiments and analyze the experimental results. In Section VI, we summarize the research of this paper.

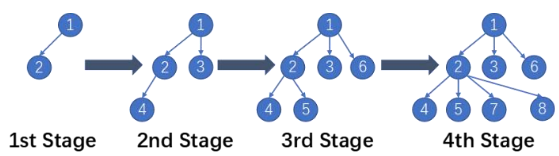

Figure 3. The rumor propagation is divided into 4 stages based on the number of nodes interval to represent the dynamic propagation of rumors, and the difference in the number of nodes between each stage is the same.

II. RELATED WORK

In recent years, research on automatic rumor detection on social media has attracted much attention. For the sake of description, this paper adopts the concept of "event", which consists of a source post and retweeted posts related to the source post. The rumor detection task aims to determine whether the event is a rumor based on information of the source post and related retweeted comments. Related research can be divided into two main categories: (1) Feature engineering-based approaches; (2) Deep learning-based approaches.

Most early studies mainly used feature engineering-based approaches to extract hand-crafted features from event-related post messages and classify events using classifiers such as decision trees, random forests, and SVM. Castillo et al. [2] considered features such as the average sentiment score of messages, and the proportion of messages with URLs in their research work on Twitter news topic credibility assessment. Yang et al.[6] considered two new features in their work on rumor detection: the client used by the user to post the message and the location of the event mentioned in the message content; Kwon et al.[16] proposed a method for rumor detection based on rumor propagation time, structure, and linguistic features; Zhang et al.[17] considered four implicit content-based features: popularity, internal and external consistency, sentiment polarity and comment opinion. Feature engineering-based methods usually require extensive pre-processing and feature design processes, which are less efficient. Hand-crafted features also have the limitation of not being robust enough.

As deep learning methods have achieved remarkable results in various natural language processing tasks, many scholars have also applied deep learning models to rumor detection. Ma et al.[10] first applied deep learning models to the field of rumor detection, using RNN and GRU to model rumor propagation, and achieved significant improvements compared to previous benchmark methods. Chen et al.[18] improved the RNN-based approach by combining attentional mechanisms with RNN and introducing attentional mechanisms to capture text features. Yu et al.[9] used Doc2vec method to obtain text vectors of tweets in periods, stitched the text vectors of each period into a feature matrix of events, and used CNN to learn the hidden layer representation of events. Bian et al.[12] first applied GCN to rumor detection research and proposed a Bi-Directional Graph Convolutional Neural Network(Bi-GCN) model that takes into account the top-down and bottom-up propagation structure of rumors. Wei et al.[13] considered the propagation uncertainty of rumor detection and proposed an Edge-enhanced Bayesian Graph Convolutional Network(EBGCN) to capture robust structural features. Lao et al.[14] considered both linear temporal sequence and the non-linear diffusion structure, capturing linear sequence features by LSTM and nonlinear structural features by GCN. For the limited rumor labeling data, He et al.[15] designed three event enhancement strategies and proposed the Rumor Detection on social media with Event Augmentations(REDA) framework to learn event representation by self-supervised pre-training. However, most of these methods focus on the textual content of rumors and ignore other types of information in social media. In contrast, the representation of the rumor propagation structure mainly focuses on the static structure of rumor propagation and ignores the dynamic change of the rumor propagation structure with the sequential posting of posts.

Previous work has provided valuable ideas for our study. To address the deficiencies in content representation, we consider both textual content and user information. In terms of propagation structure, we introduce a dynamic propagation structure representation and design a Re-GCN model for rumor detection.

III. PROBLEM STATEMENT

In this paper, the rumor detection dataset is defined as $C = \{c_1, c_2, ..., c_m\}$where c_i represents the i-th event and m is the number of events. The ground-truth label of all events defined as $Y = \{y_1, y_2, ..., y_m\}$ where y_i denotes the label of c_i. c_i contains several posts, $c_i = \{r_i, w_1^i, w_2^i, ..., w_{n_i-1}^i,\}$,where n_i denotes the number of posts contained in c_i, r_i denotes the source post, and w_j^i denotes the j-th related reply or retweet post. We denote the propagation structure of c_i by the directed graph $G_i =< V_i, E_i >$. where V_i denotes the set of nodes of G_i and E_i denotes the set of edges of G_i. As shown in Fig.1, each post is a node, and if the post w_1^i replies to r_i, there is a directed edge from r_i to w_1^i.

The problem to be solved in this paper can be summarized as follows: given a rumor dataset, learn a classifier f that maps each event in C to the ground-truth label(rumor/non-rumor)

$$f: C \rightarrow Y.$$

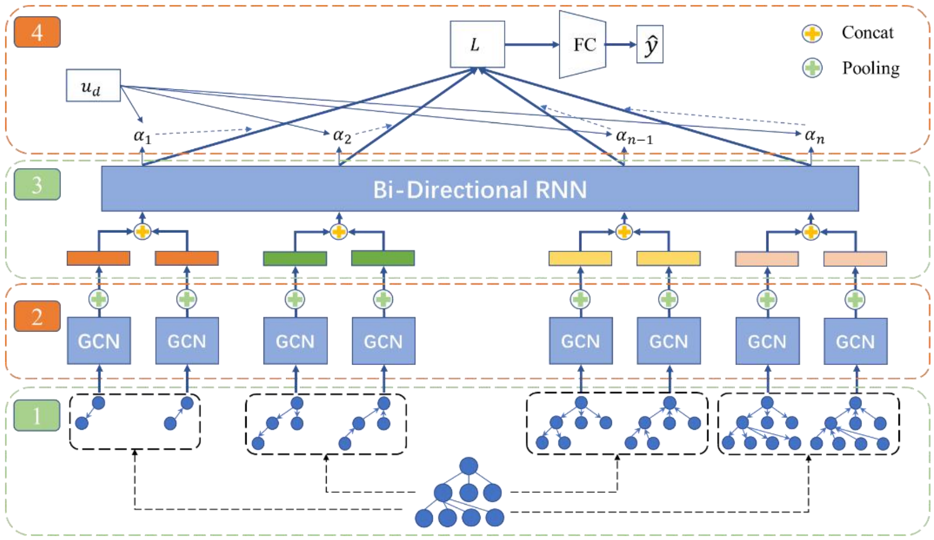

Figure 4. Our Re-GCN rumor detection model.

IV. RE-GCN RUMOR DETECTION MODEL

This section proposes a dynamic propagation structure-based rumor detection method called Recurrent Graph Convolutional Network (Re-GCN). The core idea of Re-GCN is to learn the appropriate high-level representation from the propagation structure of rumors at each stage. Re-GCN consists of four components: post content representation module, GCN module, Bi-RNN module, feature fusion and classification module. Specifically, the event representation module describes mapping the post text and user information in the event to the vector space and the construction process of the dynamic propagation structure representation of the event. The GCN module obtains the high-level node representation of each stage of event propagation by GCN. The Bi-RNN module models the dynamic changes of the propagation structure by using the high-level node representations of the event phases obtained from the GCN module as the input sequence of the Bi-RNN. The feature fusion and rumor detection module use the attention mechanism to fuse the high-level features of the events in each stage and perform classification. Fig.4 shows the structure of the proposed model.

We discuss how to use the Re-GCN to detect an event c with post count n. The same calculation is used for all other events.

A. Event Representation

The event representation of rumors is mainly divided into post content representation and dynamic propagation structure representation. We arrange all posts in the event according to the posting time from early to late, dividing them into N copies according to the number equally, which corresponds to the N stages of rumor propagation, and N takes the value of 6 in this paper. The propagation structure of the t-th stage contains the first t number of posts, i.e., $\frac{tn}{N}$ posts.

1) Post Content Representation

For the post content, taking a post in an event as an example, we consider both the user information and the text content of the post. User information features mainly refer to the content of the user account registration and some statistical features under that account. Table I and Table II show the user information in the selected Weibo[10] and PHEME[19] datasets.

TABLE I. USER INFORMATION EXTRACTED FROM WEIBO.

No.	Feature	Type
1	GENDER	Binary
2	VERIFIED	Binary
3	USER_GEO_ENABLED	Binary
4	BI_FOLLOWERS_COUNT	Integer
5	FRIENDS_COUNT	Integer
6	FOLLOWERS_COUNT	Integer
7	STATUSES_COUNT	Integer
8	FAVOURITES_COUNT	Integer
9	COMMENTS_COUNT	Integer

TABLE II. USER INFORMATION EXTRACTED FROM PHEME.

No.	Feature	Type
1	VERIFIED	Binary
2	USER_GEO_ENABLED	Binary
3	FAVOURITES_COUNT	Integer
4	RETWEET_COUNT	Integer
5	FOLLOWERS_COUNT	Integer
6	LISTED_COUNT	Integer
7	STATUSES_COUNT	Integer
8	FRIENDS_COUNT	Integer

Then we describe the preprocessing of these features. We preprocess them using One-Hot encoding for binary features such as "VERIFIED" in user information. For integer features such as "FRIENDS_COUNT" in user information, we use normalization to process them. The discrete features are then spliced with the numerical features as the user information feature representation. For text content, BERT[20] is an excellent algorithm to obtain text vector representation, and we use the pre-trained BERT-base model to extract the representation vector of text content. Finally, the user information feature representation is stitched with the text content representation to obtain the post content representation vector.

We construct the content feature matrix $X = \{X_1, X_2, ..., X_N\}$ for each stage of the event based on the N stages of rumor propagation, where X_t denotes the content feature matrix of the t-th stage, which consists of the content representation vectors of the first $\frac{tn}{N}$ posts.

2) Dynamic propagation structure representation

Taking stage t of rumor propagation as an example, we define the adjacency matrix A_t corresponding to the top-down propagation structure based on the forward-reply relationship between posts in the t stage. The values of the adjacency matrix A_t are defined in Eq.1. We consider both the top-down and bottom-up propagation structures, and the adjacency matrix corresponding to the top-down propagation structure is denoted as $A_t^{TD} = A_t$. The adjacency matrix corresponding to the bottom-up propagation structure is denoted as $A_t^{BU} = A_t^T$, A_t^{BU} is the transpose of A_t.

$$(A_t)_{ij} = \begin{cases} 1, & if\ tweet\ i\ replies\ to\ tweet\ j \\ 0, & otherwise \end{cases} \tag{1}$$

We construct the adjacency matrix $A^{TD} = \{A_1^{TD}, A_2^{TD}, ..., A_N^{TD}\}$ and $A^{BU} = \{A_1^{BU}, A_2^{BU}, ..., A_N^{BU}\}$ for both top-down and bottom-up propagation structures of N stages.

B. GCN module

GNN-based methods have achieved impressive results in many fields (e.g., click-through rate prediction[22], text classification[23]), among which GCN is a very effective method for processing graph structured data by updating the node's embedding based on its neighbors. GCN can capture information from direct and indirect neighbors of a node through a stacked hierarchy. There exist several types of message propagation functions for GCN. In this paper, we use the message propagation function defined by GCN in the first-order approximation of ChebNet[21]. For a top-down propagation structure, the 2-layers GCN is calculated as

$$H_{t_0}^{TD} = \sigma\left(\hat{A}_t^{TD} X_t W_{t_0}^{TD}\right) \tag{2}$$

$$H_{t_1}^{TD} = \sigma\left(\hat{A}_t^{TD} H_{t_0}^{TD} W_{t_1}^{TD}\right) \tag{3}$$

Where $\hat{A}_t^{TD} = D^{-\frac{1}{2}} \tilde{A}_t^{TD} D^{-\frac{1}{2}}$ is the normalized adjacency matrix, $\tilde{A}_t^{TD}$ is A_t^{TD} plus the self-loop, D is the degree matrix, and $W_{t_0}^{TD}$ and $W_{t_1}^{TD}$ denote the GCN weight parameters of the first and second layers, respectively. For the bottom-up propagation structure, the same computational procedure can be used to obtain $H_{t_1}^{BU}$ .We use $ReLU$ as the activation function. To prevent overfitting, we apply the Dropout[24] regularization method to the GCN.

We use mean pooling to aggregate information from these two sets of node representations and then stitch the top-down propagation features and bottom-up propagation features to obtain the final representation of the propagation features. It is formulated as

$$S_t^{TD} = MEAN\left(H_{t_1}^{TD}\right) \tag{4}$$

$$S_t^{BU} = MEAN\left(H_{t_1}^{BU}\right) \tag{5}$$

$$S_t = concat(S_t^{TD}, S_t^{BU}) \tag{6}$$

For each of the N stages of rumor propagation, we obtain its propagation characteristics, which are denoted as $S = \{S_1, S_2, \dots, S_N\}$ for all stages.

C. Bi-RNN module

RNN-based approaches are effective in modeling sequential data. We use Bi-RNN to model the dynamically changing propagation structure with S as the input sequence of Bi-RNN. Bi-RNN consists of forward RNN and backward RNN, and the computation procedures of forward RNN and backward RNN are shown in Eq. 7and Eq. 8, respectively. Then we splice the output of forwarding RNN and backward RNN as the output of the current moment, as shown in Eq. 9.

$$\overrightarrow{h_t} = \tanh\left(\overrightarrow{U} S_t + \overrightarrow{W} \overrightarrow{h}_{t-1} + \overrightarrow{b}\right) \tag{7}$$

$$\overleftarrow{h_t} = \tanh\left(\overleftarrow{U} S_t + \overleftarrow{W} \overleftarrow{h}_{t+1} + \overleftarrow{b}\right) \tag{8}$$

$$h_t = concat\left(\overrightarrow{h_t}, \overleftarrow{h_t}\right) \tag{9}$$

Where $(\overrightarrow{U}, \overrightarrow{W})$ and $(\overleftarrow{U}, \overleftarrow{W})$ are the weight parameters inside the forward RNN($\overrightarrow{RNN}$) and backward RNN($\overleftarrow{RNN}$) hidden layer cells, $\overrightarrow{b}$ and $\overleftarrow{b}$ are the bias of the $\overrightarrow{RNN}$ and $\overleftarrow{RNN}$. S_t denotes the propagation characteristics of the event at stage t, tanh is the activation functions of $\overrightarrow{RNN}$ and $\overleftarrow{RNN}$, respectively, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the outputs of the $\overrightarrow{RNN}$ and $\overleftarrow{RNN}$ at moment t, and h_t denotes the splicing of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ as the outputs of the Bi-RNN at moment t.

D. Feature fusion and classification

In order to enable the model to make focused use of the information in each stage of rumor propagation, we use the attention mechanism to assign attention weights to the output of the hidden state by the Bi-RNN at each moment. The attention mechanism is calculated as

$$o_t = \tanh\left(W_d h_t + b_d\right) \tag{10}$$

$$\alpha_t = \frac{\exp\left(o_t{}^T u_d\right)}{\sum_t \exp\left(o_t{}^T u_d\right)} \tag{11}$$

$$L = \sum_t \alpha_t o_t \tag{12}$$

Where o_t is the hidden layer representation of h_t after a hidden layer with $tanh$ as the activation function, W_d and b_d are the weight parameter matrix and bias, respectively. u_d is the random initialized weight parameter, α_t is the calculated weight value of the hidden state at each moment of the Bi-RNN, and L is the sequence representation obtained by weighting and summing the hidden state values at each moment. Finally, a fully connected layer and a $softmax$ layer are used to calculate the prediction results $\hat{y}$ of the events.

$$\hat{y} = Softmax\big(FC(L)\big) \tag{13}$$

We train all parameters of the Re-GCN model by minimizing the cross-entropy of the ground-truth labels and predicted labels. To speed up the convergence of the model, we use the Adam[25] optimization algorithm to update the model parameters.

V. EXPERIMENTS

In this section, we compare the performance of the proposed Re-GCN model with that of some state-of-the-art benchmark models.

A. Datasets

We evaluate our proposed method on two real-world datasets: Weibo[10] and PHEME[19]. In all datasets, nodes refer to posts, and edges represent retweet or reply relationships. Both Weibo and PHEME contain two binary labels, false rumors (F) and true rumors (T). The tags of each event in Weibo are annotated by the Sina Community Management Center, which reports various error messages[10]. PHEME contains relevant rumor and non-rumor posts that appeared on Twitter during the period when some breaking news was released. The statistics for both datasets are shown in Table III.

TABLE III. STATISTICS OF THE DATASETS

Statistic	PHEME	Weibo
Of source tweets	5802	4664
Of tree nodes	30376	3805656
Of non-rumors	3803	2351
Of rumors	1972	2313
Avg. time length/tree	18 Hours	2461 Hours
Avg. of posts/tree	6	816
Max of posts/tree	228	59318
Min of posts/tree	3	10

B. Baselines

We compare the proposed model with some state-of-the-art benchmarking methods, including feature engineering-based methods and deep learning-based methods.

The comparison methods are as follows. DTC[2]: a rumor detection method that uses a decision tree classifier based on various manual features to obtain the credibility of information. SVM-TS[26]: a linear SVM classifier that uses hand-crafted features to construct a time series model for rumor detection. DTR[27]: a decision tree based ranking model for rumor detection by querying phrases. GRU[10]: a recurrent neural network-based rumor detection model with GRU units that learns rumor representations by modeling the sequential structure of related posts. RvNN[11]: a rumor detection method based on tree-structured recurrent neural networks with GRU units that learn rumor representation by propagation structure. Bi-GCN[12]: a GCN-based rumor detection model that captures the features of rumor propagation through top-down and bottom-up rumor propagation structures.

As in the original paper[12], we randomly partitioned the dataset into five sections and performed 5-fold cross-validation to obtain stable results. For Bi-GCN, which also uses graph structures, we conducted experiments using the same Bert-based semantic vectors as this paper. For the Weibo and PHEME datasets, we evaluated the accuracy (Acc.) of the two categories and the precision (Prec.), recall (Rec.), and F1 value (F1) of each category. The hidden feature vector dimension of each node was 64, and the dropout was set to 0.5. The training procedure was iterated for 100 epochs, and early stopping[28] was used when the validation loss stopped decreasing by 7 epochs.

C. Results and analysis

Table IV and Table V show the experimental results of the proposed method with all the comparison methods on the Weibo and PHEME datasets.

TABLE IV. RUMOR DETECTION RESULTS ON WEIBO DATASET

Method	Acc.	rumor			Non-rumor		
		Prec.	***Rec.***	***F1***	***Prec.***	***Rec.***	***F1***
DTC	0.831	0.847	0.815	0.831	0.815	0.847	0.830
SVM-TS	0.857	0.839	0.885	0.861	0.878	0.830	0.857
DTR	0.732	0.738	0.715	0.726	0.726	0.749	0.737
GRU	0.899	0.865	0.946	0.904	0.940	0.852	0.894
RvNN	0.928	0.914	0.951	0.932	0.934	0.905	0.919
Bi-GCN	0.954	0.949	**0.956**	0.952	0.951	0.955	0.953
Re-GCN	**0.968**	**0.982**	0.954	**0.967**	**0.954**	**0.982**	**0.967**

TABLE V. RUMOR DETECTION RESULTS ON PHEME DATASET

Method	Acc.	rumor			Non-rumor		
		Prec.	***Rec.***	***F1***	***Prec.***	***Rec.***	***F1***
DTC	0.670	0.572	0.435	0.494	0.687	0.837	0.755
SVM-TS	0.717	0.318	0.541	0.405	0.832	0.786	0.814
DTR	0.657	0.472	0.239	0.317	0.695	0.867	0.772
GRU	0.775	0.667	0.643	0.658	0.825	0.840	0.832
RvNN	0.820	0.733	0.741	0.731	0.869	0.857	0.867
Bi-GCN	0.828	**0.771**	0.714	0.736	0.858	**0.887**	0.871
Re-GCN	**0.845**	0.767	**0.781**	**0.772**	**0.886**	0.879	**0.882**

First, it can be clearly observed that the deep learning-based methods (GRU, RvNN, Bi-GCN, and Re-GCN) significantly outperform the manual feature-based methods (DTC, SVM-TS, and DTR). This is mainly due to the ability of deep learning methods to learn advanced representations of rumors and thus capture more effective features. This proves the importance of studying deep learning methods for rumor detection.

Secondly, among the deep learning-based methods, RvNN, Bi-GCN, and Re-GCN perform better relative to GRU since they consider the rumor propagation structure, while GRU ignores important rumor propagation structure features. It can also be observed that RvNN performs poorly relative to Bi-GCN due to the fact that RvNN is tree-structured by nature and it uses the hidden features of all leaf nodes as the final event representation, so it is more susceptible to the latest postings, resulting in the loss of more information from previous posts.

Finally, our proposed Re-GCN method is significantly better than the Bi-GCN method. There are two main reasons. First, Bi-GCN only considers the static propagation structure, and Re-GCN takes into account the important factor of the dynamically changing propagation structure of rumors. Second, compared with Bi-GCN, which only uses textual content to represent events, Re-GCN considers both textual content and user information. This demonstrates the effectiveness of introducing user information and dynamically changing the propagation structure for rumor detection.

D. Ablation study

In order to analyze the effect of introducing user information and dividing the number of stages N, we compared different values of N and the use of text-only content with the introduction of user information.

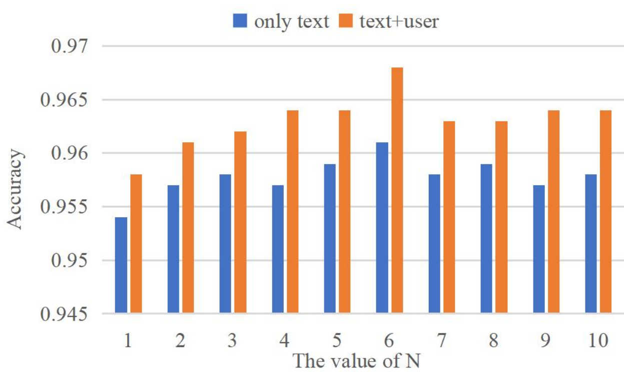

Figure 5. Experimental results for different values of N.

The experimental results on the Weibo dataset are shown in Fig.5, where the range of N is set to 1-10, “only text” means using only text as the post content representation, and “text+user” means splicing text and user information as the post content representation. According to Fig.5, firstly, it is evident that the accuracy rate of both datasets increases to different degrees after the introduction of user information, which indicates the effectiveness of introducing user information. Second, starting from a value of N of 1, the accuracy of rumor detection increases gradually as N's value increases, reaching the best result when N is taken as 6. When N is greater than 6, the model's accuracy decreases slightly but still outperforms all compared benchmark models, which indicates the effectiveness of the propagation structure considering dynamic changes.

In order to investigate the effect of the number of graph convolution layers on the model detection, we conducted experiments with the number of graph convolution layers set to 1, 2, 4 and 6, and the experimental results on the Weibo dataset are shown in Fig.6. According to Fig.6, it is not the case that the larger the number of graph convolution layers is, the better the overall performance is when the number of graph convolution layers is set to 2. Since increasing the depth of the graph convolution layers brings more parameters to the model, and the amount of data used for training is relatively insufficient, resulting in a possible overfitting of the model, the detection accuracy no longer improves with the increase in the number of graph convolution layers.

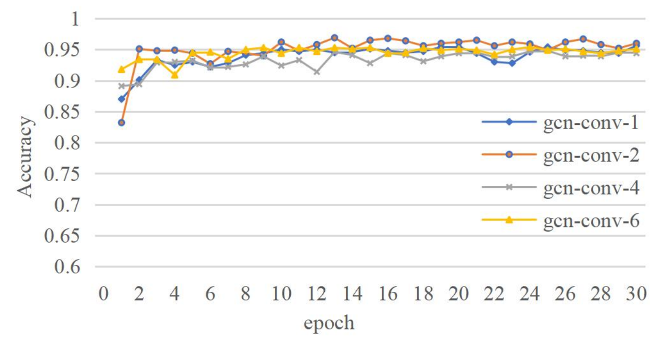

Figure 6. Experimental results of different graph convolution layers.

E. Early rumor detection

Early rumor detection is one of the important metrics for evaluating rumor detection methods, aiming to assess the ability of the method to detect rumors at an early stage of dissemination. We set a series of cutoff times and use only posts published before the cutoff time to compare the proposed method with the benchmark method.

Fig.7 shows the performance of our Re-GCN method and some benchmark methods for the Weibo dataset with different cutoff times. As can be seen from the Fig.7 Re-GCN achieves high rumor detection accuracy early in the propagation and outperforms the compared benchmark method at different cutoff times. This shows that considering user characteristics and dynamic propagation structure not only facilitates long-term rumor detection, but also helps to detect rumors early in propagation.

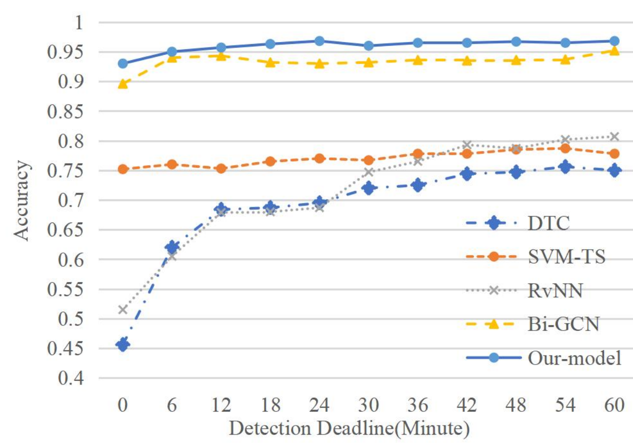

Figure 7. Experimental results of early rumor detection.

F. Case study

Take as an example the event in the PHEME dataset whose source tweet id is 500235112785924096, the propagation structure and some text contents of the posts in this event are shown in Fig.8. The event is labeled as a true rumor, which is incorrectly classified as a non-rumor by Bi-GCN using a static propagation structure, and correctly classified as a true rumor by our Re-GCN.

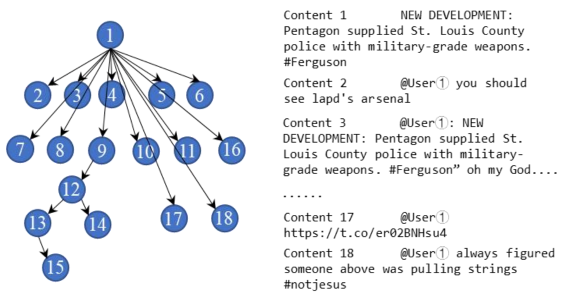

Figure 8. Propagation structure of the event with source twitter id 500235112785924096.

We extracted the attention values of the 6 stages when Re-GCN classified the event, as shown in Table VI. It can be seen that Re-GCN correctly classifies the event with the main clues of 3rd and 5th stage of the propagation of this event. As described in subsection *D* of Section IV, our proposed model is able to focus on diffierent stage of propagation features with the attention mechanism, which is beneficial to improve the effectiveness of rumor detection.

TABLE VI. ATTENTION VALUES IN 6 STAGES OF EVENT PROPAGATION

Terms	Stage					
	1st	*2nd*	*3rd*	*4th*	*5th*	*6th*
Attention Values	0	0	24.1	0	75.7	0.15

VI. CONCLUSION

This paper proposes a dynamic propagation structure-based rumor detection model named Re-GCN. First, compared with previous studies that mainly consider textual content, we encode both user information and textual content as post content representation, which provides richer information for the rumor detection model. Second, we introduced a dynamic propagation structure and constructed a dynamically changing propagation structure representation by dividing rumor propagation into multiple stages. Finally, the dynamically changing propagation features are captured using Bi-RNN and GCN, and an attention mechanism fuses the dynamic propagation features. Experimental results on two real-world datasets show that the proposed approach outperforms other state-of-the-art models for the rumor detection task.

REFERENCES

[1] Weibo:The 2020 Annual Report On counter-Rumor Work. https://weibo.com/1866405545/K0QaImwsK

[2] Castillo C, Mendoza M, Poblete B. Information credibility on twitter[C]//Proceedings of the 20th international conference on World wide web. 2011: 675-684.

[3] Popat K. Assessing the credibility of claims on the web[C]//Proceedings of the 26th International Conference on World Wide Web Companion. 2017: 735-739.

[4] Sampson J, Morstatter F, Wu L, et al. Leveraging the implicit structure within social media for emergent rumor detection[C]//Proceedings of the 25th ACM international on conference on information and knowledge management. 2016: 2377-2382.

[5] Ma J, Gao W, Wong K F. Detect rumors in microblog posts using propagation structure via kernel learning[C]. Association for Computational Linguistics, 2017.

[6] Yang F, Liu Y, Yu X, et al. Automatic detection of rumor on sina weibo[C]//Proceedings of the ACM SIGKDD workshop on mining data semantics. 2012: 1-7.

[7] Wu K, Yang S, Zhu K Q. False rumors detection on sina weibo by propagation structures[C]//2015 IEEE 31st international conference on data engineering. IEEE, 2015: 651-662.

[8] Liu X, Nourbakhsh A, Li Q, et al. Real-time rumor debunking on twitter[C]//Proceedings of the 24th ACM international on conference on information and knowledge management. 2015: 1867-1870.

[9] Yu F, Liu Q, Wu S, et al. A Convolutional Approach for Misinformation Identification[C]//IJCAI. 2017: 3901-3907.

[10] Ma J, Gao W, Mitra P, et al. Detecting rumors from microblogs with recurrent neural networks[J]. 2016.

[11] Ma J, Gao W, Wong K F. Rumor detection on twitter with tree-structured recursive neural networks[C]. Association for Computational Linguistics, 2018.

[12] Bian T, Xiao X, Xu T, et al. Rumor detection on social media with bi-directional graph convolutional networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 549-556.

[13] Wei L, Hu D, Zhou W, et al. Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection[J]. arXiv preprint arXiv:2107.11934, 2021.

[14] Lao A, Shi C, Yang Y. Rumor detection with field of linear and non-linear propagation[C]//Proceedings of the Web Conference 2021. 2021: 3178-3187.

[15] He Z, Li C, Zhou F, et al. Rumor Detection on Social Media with Event Augmentations[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 2020-2024.

[16] Kwon S, Cha M, Jung K, et al. Prominent features of rumor propagation in online social media[C]//2013 IEEE 13th international conference on data mining. IEEE, 2013: 1103-1108.

[17] Zhang Q, Zhang S, Dong J, et al. Automatic detection of rumor on social network[M]//Natural Language Processing and Chinese Computing. Springer, Cham, 2015: 113-122.

[18] Chen T, Li X, Yin H, et al. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection[C]//Pacific-Asia conference on knowledge discovery and data mining. Springer, Cham, 2018: 40-52.

[19] Zubiaga A, Liakata M, Procter R. Learning reporting dynamics during breaking news for rumour detection in social media[J]. arXiv preprint arXiv:1610.07363, 2016.

[20] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[21] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[22] Fang W, Lu L. Deep Graph Attention Neural Network for Click-Through Rate Prediction[C]//SEKE. 2020: 483-488.

[23] Liu X, You X, Zhang X, et al. Tensor graph convolutional networks for text classification[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8409-8416.

[24] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

[25] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[26] Ma J, Gao W, Wei Z, et al. Detect rumors using time series of social context information on microblogging websites[C]//Proceedings of the 24th ACM international on conference on information and knowledge management. 2015: 1751-1754.

[27] Zhao Z, Resnick P, Mei Q. Enquiring minds: Early detection of rumors in social media from enquiry posts[C]//Proceedings of the 24th international conference on world wide web. 2015: 1395-1405.

[28] Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning[J]. Constructive Approximation, 2007, 26(2): 289-315.