Hindawi
Security and Communication Networks
Volume 2022, Article ID 4840997, 13 pages
https://doi.org/10.1155/2022/4840997

Research Article

Rumor Detection with Bidirectional Graph Attention Networks

Xiaohui Yang, Hailong Ma, and Miao Wang

School of Cyberspace Security and Computer, Hebei University, Baoding 071000, China

Correspondence should be addressed to Miao Wang; 2629101909@qq.com

Received 15 September 2021; Accepted 23 December 2021; Published 18 January 2022

Academic Editor: Xingsi Xue

In order to extract the relevant features of rumors effectively, this paper proposes a novel rumor detection model with bidirectional graph attention network on the basis of constructing a directed graph, named P-BiGAT. Firstly, this model builds the propagation tree and diffusion tree through the tweet comment and reposting relationship. Secondly, the improved graph attention network (GAT) is used to extract the propagation feature and the diffusion feature through two different directions, and the multihead attention mechanism is used to extract the semantic information of the source tweet. Finally, the propagation feature, diffusion feature, and semantic information representation of the source tweet are connected together through a fully connected layer, and the mapping function is used to determine the authenticity of the information. In addition, this paper also proposes a new node update method and applies it to the model in order to select neighbor node information effectively. Specifically, it can select the neighbor information node with larger weight to update the node according to the weight of the neighbor node. The results of the experiment show that the model is better than the baseline method of comparison in accuracy, precision, recall, and F1 measure on the public datasets.

1. Introduction

With the development of technology and the popularization of the Internet, a large number of rumors have also been born in recent years. Social software based on Sina Weibo, Twitter, and Facebook has become the main birthplace of rumors. In 2021, the official report released by Sina Weibo showed that 5332 rumors were dealt with in April alone [1]. In order to have a healthy network environment, scholars at home and abroad have begun to carry out related research on text information in social networks. They usually focus on the text content [1–6], user information [1, 7], and other related features [8, 9].

From a social perspective, in order to attract the public's attention and gain social attention, rumors are usually disguised as close to real information, which will affect the society to a certain extent and may even create a bad social atmosphere and cause social unrest. From a national perspective, the impact of rumors on the country cannot be ignored. Although some rumors cannot directly endanger national security, these rumors will indirectly affect national security by affecting personal and social security. In order to have a healthy living environment, most scholars have begun to study advanced rumor detection technology at home and abroad, which has gradually improved the accuracy of rumor detection.

Due to the large amount of rumor detection data, the method based on deep learning has become the choice of most scholars [10]. Compared with traditional manual feature extraction methods, it can process large amounts of data more effectively and extract data features for detection. With the emergence of graph neural networks, the powerful feature representation ability of graph structure is gradually replacing the methods of traditional deep learning. Since graph neural network can effectively extract features in the graph domain and has good performance and interpretability, it has become a widely used feature extraction method [11]. Some scholars have begun to study the usability of graph neural networks for rumor detection. The latest rumor detection methods are mainly divided into two categories: methods based on traditional deep learning and methods based on graph neural networks.

Researchers use deep learning tools to obtain relevant features for rumor detection. For example, Ma et al. [8]

proposed a new model for rumor detection based on recurrent neural networks, which can capture contextual features that change over time and effectively improve the performance of rumor detection by adding complex recursive units and hidden layers. Wu et al. [12] proposed an SVM classifier based on a graphics kernel, which can effectively capture text information features, propagation features, and user attribute information. Wang et al. [13] combined reinforcement learning and deep learning and proposed an early rumor detection model based on Q-learning, which can effectively improve the timeliness and accuracy of early rumor detection. Chen et al. [14] can better judge the importance of different texts by focusing the attention mechanism on different texts. Gao et al. [15] used the specific task features based on the bidirectional language model and the superimposed LSTM to express the text information and time information of the input source tweets. In addition, they also introduced a multilayer attention model to learn the embedding of the context. Alsaeedi and Al-sarem [16] proposed a model architecture based on convolutional neural network and discussed its hyperparameter settings in detail to achieve the best detection effect. Azri et al. [17] proposed an end-to-end deep hybrid rumor detection model by fusing image features, text features, and emotional features of posts. However, most of these methods focus on extracting the text features of rumors, ignoring the propagation features of rumors in real life, or modeling the propagation features as a sequence structure, which cannot effectively express the propagation features of rumors.

The method based on graph neural network is different from the traditional deep learning method. The method based on graph neural network focuses on discovering the difference between the propagation features and the diffusion features of real information and false information. For example, Yuan et al. [18] proposed a new heterogeneous graph network model, which extracts features by constructing different source tweets and ancillary information. Bian et al. [19] obtained the propagation and diffusion features of rumors through top-down and bottom-up node updates and proposed a rumor detection model with bidirectional graph convolutional network. Ke et al. [20] proposed constructing the global relationship between all source tweets, response tweets, and users as a heterogeneous graph to obtain interactive feature information. Dou et al. [21] proposed a rumor detection method based on a multirelational propagation tree, which comprehensively considers the interlayer dependencies and intralayer dependencies of nodes. Yang et al. [21] proposed a graph convolutional network model fused with gating units, which selectively enhanced the information representation of nodes. Lotfi et al. [22] constructed user graphs and tweet graphs and captured the user behavior characteristics of rumors and the global features of tweets through graph convolutional neural networks, which greatly improved the accuracy of rumor detection. Although these methods improve the effectiveness of obtaining propagation features and diffusion features, most graph neural network methods use directed propagation on undirected graphs, ignoring the practical significance of directed graphs in rumor detection.

In order to solve the above problems, this paper proposes a rumor detection model (P-BiGAT) based on GAT. In addition, we also consider the phenomenon that spammers usually post advertisements under the source tweets which are not related to the source tweets and propose a node update method based on threshold P and apply it to the model. Experiments show that the model has improved accuracy, precision, recall, and F1 measure in rumor detection compared to the baseline method.

The main contributions of this paper are as follows:

(1) We propose a new method for node update, which can effectively use neighbor nodes with larger weights to update nodes through bottom-up and top-down directions.

(2) Different from other models, based on the premise of constructing directed graphs, this paper applies GAT, which is good at processing directed graphs, to rumor detection and proposes a new rumor detection model (P-BiGAT).

2. Preliminaries

First, we introduce the relevant definitions involved and then lead to the problems of this paper and the symbols used in the paper:

Definition 1 (rumor): the true value cannot be verified or deliberately false statement [23].

Definition 2 (source tweet): it refers to the original tweet, not response to any other tweet. This paper uses r_i to represent the source tweet of the i-th event.

Definition 3 (response tweets): reply to the source tweet or other response tweets, including comments on the tweet and reposted content. This paper uses w_{ij} to represent the j-th related response tweet in the i-th event.

Definition 4 (node): the feature vector of source tweet and response tweet conversion. This paper uses h_i^t to represent the feature vector after the t-th update of node i.

Definition 5 (propagation tree/diffusion tree): the feature vector stored in the node constructs the edge through the index method to generate a tree structure. The top-down pointing direction of the node represents the propagation tree; the bottom-up pointing direction of the node represents the diffusion tree.

Definition 6 (weight): the importance of neighbor nodes in the tree to their own nodes. This paper uses the attention mechanism to calculate α_{ij} to indicate the importance of neighbor node j to node i.

Tweets on the Internet can be divided into real information and false information. The purpose of rumor detection is to detect false information posted on the Internet based on the relevant features of the tweets, and its essence is a text classification problem. The formal definition is as follows: Consider a given rumor detection dataset $C = \{c_1, c_2, \ldots, c_m\}$, where c_i represents the i-th event: $c_i = \{r_i, w_{i1}, w_{i2}, \ldots, w_{iz}\}$. The rumor category label is L, the mapping function is Y, and the process of judging is $Y\colon C \longrightarrow L$.

3. P-BiGAT Rumor Detection Model

The rumor detection model P-BiGAT proposed in this paper is shown in Figure 1. The model includes three parts: the extraction of propagation feature and diffusion feature, the representation of semantic information, and the classification of rumor detection. The extraction of propagation feature and diffusion feature module uses the improved GAT model for feature extraction by constructing a directed propagation tree and diffusion tree; the representation of semantic information module uses a multihead attention mechanism to represent the semantic information of the source tweet; the classification of rumor detection module connects the propagation features, diffusion features, and the semantic information of the source tweet and uses the mapping function to determine whether the source tweet is a rumor.

3.1. Feature Representation of Propagation and Diffusion

3.1.1. Tree Construction. Both source tweets r_i and response tweets w_{ij} are text type data. For text type data, we use the BERT [24] model to convert text type data into vector encoding. Compared with other methods, BERT can extract contextual semantic information through a deep bidirectional pretraining model. For different tasks, it does not require substantial modifications to the model, and the effect has been greatly improved.

This paper obtains the propagation features and diffusion features in the BiGCN [19] model under the premise of constructing a directed graph. First, we embed the extracted text feature vector into the node; then, if there is a direct forwarding or comment relationship between the nodes, we think that there is a connection between the two nodes, thereby establishing a connection; finally, according to the direction of the parent node and the child node, the direction of the parent node pointing to the child node represents the propagation tree of the rumor as shown in Figure 2(a), and the direction of the child node pointing to the parent node represents the diffusion tree of the rumor as shown in Figure 2(b).

3.1.2. Node Update. This paper is based on the GAT model to update the feature vector of the node. We iteratively update the nodes through the locally coded graph structure and node feature vectors, thereby obtaining a vector representation with global features. In the step of iteratively updating a single node, first the neighbor node obtains the vector representation of neighbor node information through different types of relationship paths; then the own node gathers the neighbor node information and its own node information to update the representation of its own node information; finally, the newly obtained node information contains both neighbor information and self-information to indicate the propagation or diffusion features of the entire tweet. When calculating the corresponding hidden layer information of each node, the attention mechanism is introduced as the weight to express the importance of nodes.

The specific update process of the node is shown in Figure 3. The t-th update of node $h_1^{(t-1)}$ is to obtain its own node $h_1^{(t-1)}$ and neighboring node $(h_2^{(t-1)}, h_3^{(t-1)}, \ldots, h_n^{(t-1)})$ after the $(t-1)$-th update of the node is completed. These nodes are, respectively, multiplied by their own weights and then connected or averaged to obtain the h_1^t node. The weight $(\alpha_{11}, \alpha_{12}, \alpha_{13}, \ldots, \alpha_{1n})$ on the edge is automatically output by the attention network.

Taking the t-th node update process of the propagation tree as an example, we assume that the input layer is

$$\left(h_1^{(t-1)}, h_2^{(t-1)}, \ldots, h_n^{(t-1)}\right), \quad h_i^{(t-1)} \in R^F, \tag{1}$$

where $i \in [1, n]$, n is the number of nodes, F is the number of features, and R^F indicates that each node is a feature space of size F. The output layer is

$$\left(h_1^t, h_2^t, \ldots, h_n^t\right), \quad h_i^t \in R^{F'}. \tag{2}$$

For each pair of neighbor nodes, there is α_{ij} to describe the importance of node i and neighbor node j. The specific calculation steps are shown in Figure 4.

Firstly, the attention coefficient e_{ij} of node i and its neighbor node j is calculated through the attention function:

$$e_{ij} = \text{attention}\left(Wh_i^{(t-1)}, Wh_j^{(t-1)}\right), \tag{3}$$

where $W \in R^{F\times F\prime}$ is the weight matrix that the node can train, which represents the relationship between the input features F and the output features F'; $j \in N_i$, where N_i represents all neighbor nodes of node i.

Secondly, we use the softmax function to regularize the attention coefficient e_{ij} in order to make the attention coefficient easier to calculate and compare. The results of obtaining weight α_{ij} are as follows:

$$\begin{aligned}\alpha_{ij} &= \text{softmax}\left(e_{ij}\right)\\ &= \frac{\exp\left(e_{ij}\right)}{\sum_{k\in N_i}\exp\left(e_{ik}\right)}.\end{aligned} \tag{4}$$

In the experiments of this paper, weight α_{ij} is calculated and output by a single-layer feedforward neural network, and the Leaky ReLU activation function is used in the output layer of the neural network. The Leaky ReLU activation function is an improvement of the ReLU activation function. By introducing a nonzero slope in the assignment area, it solves the “death” problem of the ReLU function in the negative area and has a good performance in practical

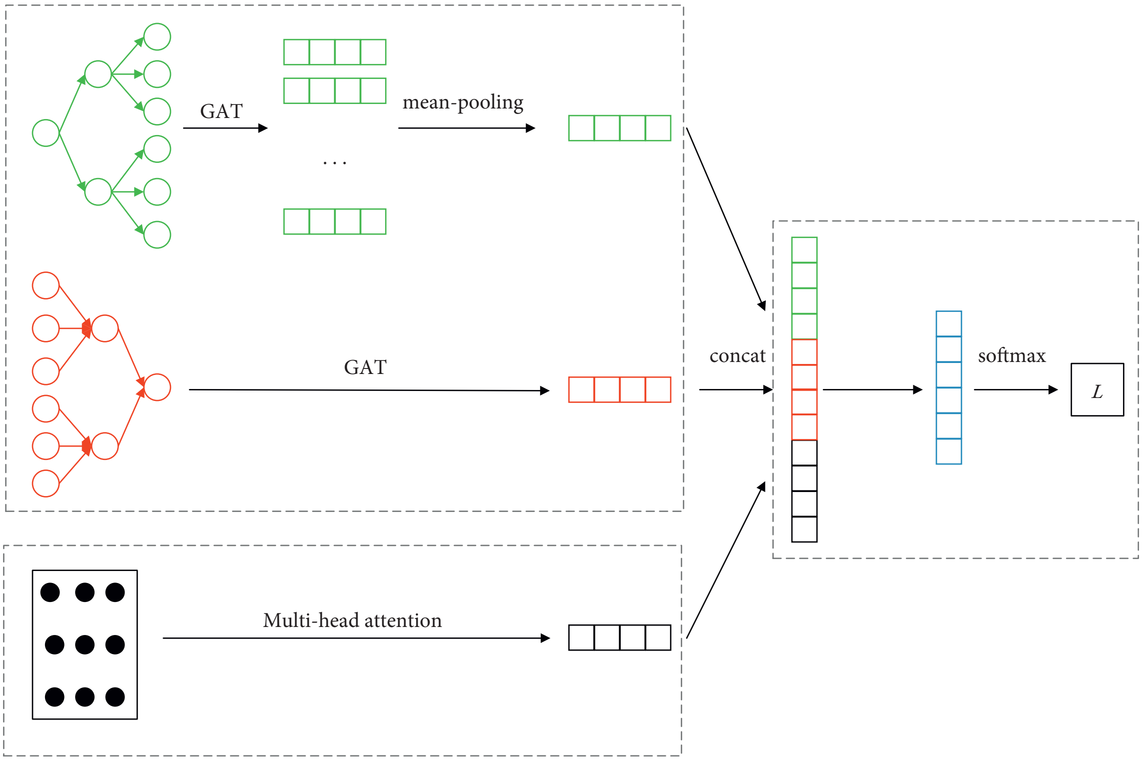


FIGURE 1: The structure of P-BiGAT model.

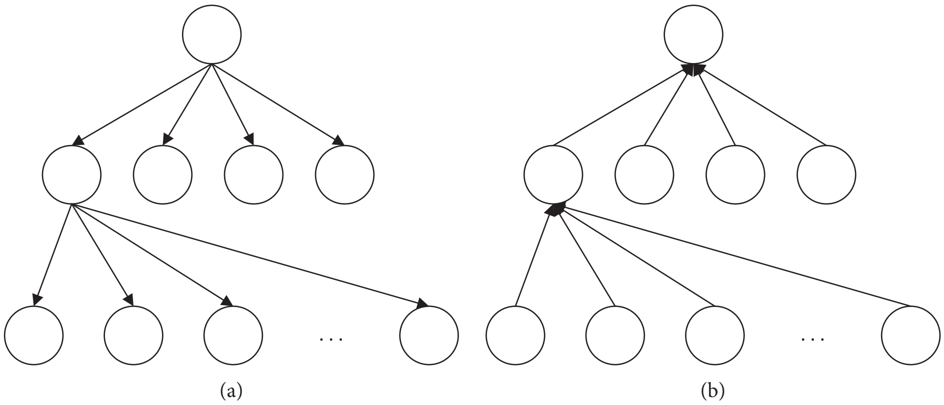


FIGURE 2: The structure of rumor tree. (a) Propagation tree of the rumor; (b) diffusion tree of the rumor.

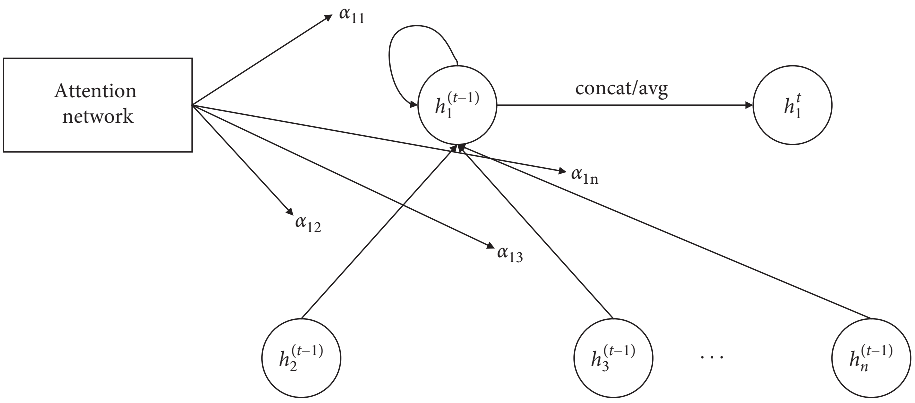


FIGURE 3: The process of node update.

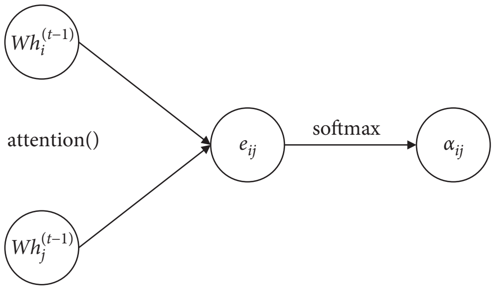


FIGURE 4: The calculation process of the weight α_{ij}.

applications. Finally, combining (3) and (4), weight α_{ij} is as follows:

$$\alpha_{ij} = \text{softmax}\left(e_{ij}\right) = \frac{\exp\left(\text{LeakyRelu}\left(\overrightarrow{a}^T\left[Wh_i^{(t-1)}\|Wh_j^{(t-1)}\right]\right)\right)}{\sum_{k\in N_i}\left(\text{expLeakyRelu}\left(\overrightarrow{a}^T\left[Wh_i^{(t-1)}\|Wh_k^{(t-1)}\right]\right)\right)}, \tag{5}$$

where $\overrightarrow{a} \in R^{2F'}$ represents the weight matrix between layers in the neural network; T represents the transposition operation; $\|$ represents the connection operation.

After calculating weight α_{ij} of the neighbor nodes, the final output feature vector of each node h_i^t can be calculated. The equation is as follows:

$$h_i^t = \sigma\left(\sum_{j\in N_i}\alpha_{ij}Wh_j^{(t-1)}\right), \tag{6}$$

where σ is a nonlinear activation function. In order to make the model more stable, we use a multilayer attention mechanism to merge multiple results, that is, calculate multiple self-attention mechanisms at the same time (as shown in equation (6)) and then obtain h_i^t by averaging. Assuming that there are K self-attention mechanisms that execute equation (6), the specific equation for merging using the multihead attention mechanism is as follows:

$$h_i^t = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^k W^k h_j^{(t-1)}\right), \tag{7}$$

where W^k represents the trainable weight matrix of the node under the k-th feature vector and α_{ij}^k represents the importance of the neighbor node j to node i under the k-th feature vector.

In order to filter out neighbor nodes that have nothing to do with nodes, improve model performance, and prevent oversmoothing and overfitting, this paper proposes a node update process based on threshold P. First, we sort all neighbor nodes including the target node from large to small according to the node weight and set threshold P. We select neighbor nodes greater than or equal to threshold P for node update and discard neighbor nodes with a weight less than threshold P. After deleting the edge with the smaller point weight, we redistribute the weight to calculate the node update. In addition, this paper uses the residual connection method to apply to the process of node update, connecting the feature vector of the root node at time *t-1* with the feature vector of each node's hidden layer at time t. It not only better enhances the root node features but also better prevents overfitting problems. The equation is as follows:

$$h_i^t = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^k W^k h_j^{(t-1)}\right) + h_{\text{root}}^{(t-1)}\,(\alpha > P). \tag{8}$$

3.1.3. Feature Representation. In the bottom-up propagation tree, we use HBU to represent the propagation features of tweets. In this model, since the nodes all point to the root node, we use the convergence output result h_{root}^t of the root node r_i to represent the propagation features of the entire event c_i. The equation is as follows:

$$\text{HBU} = h_{\text{root}}^t. \tag{9}$$

In the top-down diffusion tree, we use HTD to represent the diffusion features of tweets. In this model, since all nodes point to leaf nodes, we use the convergence output result $(h_{v1'}^t, h_{v2'}^t, \ldots, h_{vn'}^t)$ of leaf node $vi\prime$ to represent the diffusion features of the entire event c_i. Since the feature vectors of multiple leaf nodes are output, we use the mean pooling operation to aggregate the information of each leaf node, which is expressed as follows:

$$\text{HTD} = \text{MEAN}\left(h_{v1'}^t, h_{v2'}^t, \ldots, h_{vn'}^t\right). \tag{10}$$

3.2. Representation of Semantic Information. Since the multihead attention mechanism can capture the internal dependencies of the text sequence, it has a good effect on the representation of the semantic information of the text. In order to be able to more accurately determine whether the source tweet r_i is a rumor, this paper uses the multihead attention mechanism proposed by Ashish et al. [25] to further extract the semantic features of the source tweet and symbol T_i is used to represent it. The principle of the multihead attention mechanism is scaled dot-product attention. The specific equation is as follows:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{\text{QK}^T}{\sqrt{d_k}}\right)V, \tag{11}$$

where Q, K, and V are, respectively, the feature mapping of the feature vectors under three different weight coefficient matrices W^Q, W^K, W^V, representing the query vector, key vector, and value vector; d_k is the dimension of the key vector; $1/\sqrt{d_k}$ acts as an adjustment function to prevent the inner products of Q and K^T from being too large and to ensure the stability of the final value. The score of each key vector is obtained through the query vector, and then the corresponding weight QK^T is calculated according to the score, and the weight obtained is multiplied by the value vector and the value vector is weighted and averaged.

In order to better represent the ability of different text positions and improve the learning ability of the subspace representation of the attention unit, the multihead attention mechanism improves the scaled dot-product attention. The specific process is shown in Figure 5.

First, the input part is changed from the original Q, K, V to $\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V$. The specific process is as follows:

$$\begin{aligned}\left[Q_1, Q_2, \ldots, Q_h\right] &= \left[Q_1 W_1^{Q_1}, Q_2 W_2^{Q_2}, \ldots, Q_h W_h^{Q_h}\right],\\ \left[K_1, K_2, \ldots, K_h\right] &= \left[K_1 W_1^{K_1}, K_2 W_2^{K_2}, \ldots, K_h W_h^{K_h}\right],\\ \left[V_1, V_2, \ldots, V_h\right] &= \left[V_1 W_1^{V_1}, V_2 W_2^{V_2}, \ldots, V_h W_h^{V_h}\right],\end{aligned} \tag{12}$$

where W^Q, W^K, W^V represent the initialization weights of Q, K, V, which are continuously updated in the iterative update process.

Secondly, we concatenate the results and multiply the concatenated results by W^0 to get the sentence matrix. The specific process is as follows:

$$\begin{aligned}\text{wherehead}_i &= \text{attention}\left(\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V\right),\\ \text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right) W^0.\end{aligned} \tag{13}$$

Finally, the model learns the representation of the source tweet r_i through the multihead attention mechanism, so as to obtain the representation T_i of the semantic information of the source tweet r_i.

3.3. Classification of Rumor Detection. Through the above methods, we obtain the propagation feature HBU, the diffusion feature HTD, and the representation T_i of the semantic information of the source tweet to be detected and then concatenate the above feature vectors:

$$H = \text{Concat}\left(\text{HBU}, \text{HTD}, T_i\right). \tag{14}$$

Finally, we pass the feature vector H through a fully connected layer, the purpose of which is to reduce the dimensionality of the feature vector and delete redundant feature information. After leaving the new feature $\widehat{H}$, we use the mapping function Y to determine the category label of event c_i, the mapping function Y is expressed as follows:

$$Y = \text{softmax}(W\widehat{H} + b), \tag{15}$$

where $W \in R$ is the weight parameter and $b \in R$ is the bias term.

Finally, the model is trained through the cross-entropy loss function. In order to reduce the complexity of the model and prevent overfitting, a regular term is added after the loss function:

$$\text{Loss} = -\sum_{n=1}^{N} \sum_{l=1}^{L} y_l\left(x_n\right) \log Y_l\left(x_n\right) + \lambda \left\|\theta^2\right\|, \tag{16}$$

where N is the number of training samples, L is the label of rumors that define the rumor, $y_l(x_n)$ represents the true value of the n-th sample as l class, $Y_l(x_n)$ represents the probability that the n-th sample is class l, λ represents the regularization coefficient, and $\|\theta^2\|$ represents the L_2 regular term. During the training process, all model parameters are updated using an effective backpropagation algorithm.

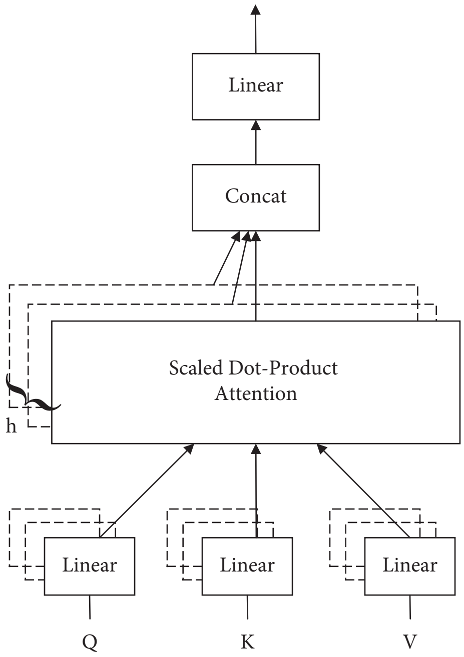


Figure 5: The diagram of multihead attention mechanism structure.

4. Experiment

In order to verify the performance of the proposed model, this paper compares the rumor detection performances of several baseline models on the public datasets and verifies the effectiveness of the proposed node update method based on threshold P. Finally, this paper also compares and analyzes the performances of the P-BiGAT model and the baseline method in early rumor detection.

4.1. Datasets. This paper uses three public datasets, Weibo [8], Twitter15, [26] and Twitter16 [26], to evaluate the performance of the model. The Weibo dataset includes two types of data labels: false rumors (F) and true rumors (T). The Twitter15 and Twitter16 datasets contain four types of data labels: nonrumors (NR), false rumors (FR), true rumors (TR), and unverified rumors (UR). The detailed statistics of the three datasets are shown in Table 1.

This paper uses 10% of the data as the verification dataset, and the remaining datasets are divided into training datasets and test datasets with a ratio of 3 : 1. For irrelevant information such as image references and hyperlinks in the dataset, we delete them through regular expressions to reduce noise.

4.2. Evaluation Index. For the Weibo dataset, we use standard model evaluation indicators: accuracy (Acc), precision (Prec), recall, and F1 measure. For the Twitter15 and Twitter16 datasets, we only consider the accuracy to evaluate the 4 label types and use the F1 measure to evaluate each label type. The specific calculation equations are as follows:

Table 1: Statistics of the datasets.

	Twitter15	Twitter16	Weibo
Total number of source tweets	1490	818	4664
Total number of users	276663	173487	2746818
Total number of tweets	331612	204820	3805656
Amount of real information	374	205	2351
Number of false rumors	370	205	2313
Number of true rumors	372	207	0
Number of unconfirmed rumors	374	201	0

$$\begin{aligned}\text{accuracy} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}},\\ \text{precision} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},\\ \text{recall} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\\ F1 &= \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}.\end{aligned}\tag{17}$$

The specific meanings of TP, TN, FP, and FN are shown in Table 2.

4.3. Parameter Setting. This model sets the number of samples selected for one training Batch_size = 64, the learning rate $\eta = 0.02$, and the regularization term coefficient $\lambda = 0.001$ in the regularization L_2. A two-layer GAT model is adopted, and both layers include $K = 4$ attention heads. In the training process, the stochastic gradient descent method is used to update the model parameters. In addition, we use the Adam [27] optimizer to train the P-BiGAT model, because it comprehensively considers the first-order moment estimation and the second-order moment estimation and can iteratively update the weight of the neural network based on the training data, adaptively adjust the learning rate, and perform efficient calculations with less memory. Among them, β_1 and β_2 are set to 0.9 and 0.999, respectively.

After setting the above parameters, threshold P is added in the node update process, and the selection of the P value is evaluated with accuracy. Firstly, we set $P = 0$ as the initial value, with 0.01 as the step width, until $P = 1$, and find the predicted label corresponding to each P value. Secondly, all the predicted labels obtained are compared with the real labels. Finally, we find the accuracy rate corresponding to each threshold P, so as to find threshold P corresponding to the optimal accuracy rate among many accuracy rates. In the Weibo dataset, when $P = 0.24$, the accuracy reaches the maximum; in the Twitter15 and Twitter16 datasets, when $P = 0.18$, the accuracy reaches the maximum.

4.4. Performance Analysis

4.4.1. Baseline Method Selection. In order to verify the performance of the P-BiGAT model, we choose some baseline methods for comparison with the methods in this paper:

Table 2: The concrete representation of the symbol.

Predicted value	Actual value	
	1	0
1	TP	FP
0	FN	TN

GLAN [18]: a rumor detection method based on heterogeneous graphs, which combines local semantic information and global structural information to capture the features of rumor detection.

BiGCN [19]: a rumor detection method based on graph convolutional neural network (GCN). On the basis of constructing an undirected tree, GCN is used to perform top-down and bottom-up convolution of the tree structure to express the propagation and diffusion of rumors.

KZWANG [20]: a rumor detection method based on graph neural network and attention mechanism.

DTC [1]: a rumor detection method that manually extracts multiple global features and uses decision trees for classification.

SVM-TS [28]: a method of extracting the manual features of Weibo events dynamically changing over time and using a linear SVM classifier for rumor detection.

PPC [29]: a rumor detection model that combines recurrent neural networks and convolutional neural networks to capture the dynamic changes of user features along the propagation path.

RvNN [30]: a rumor detection model based on a tree-shaped recurrent neural network, which can capture the dynamic changes of contextual information over time.

4.4.2. Overall Performance. It can be seen from Tables 3, 4, and 5 that the P-BiGAT model is superior to other methods in the three datasets. As shown in Table 3, the P-BiGAT model has the highest accuracy rate of 96.7% among all methods in the Weibo dataset. The recall rate and F1 measure are also better than those in the baseline method of comparison. This is because the GAT-based method has a natural advantage in dealing with directed graphs, while the GCN-based method has an advantage in dealing with undirected graphs. It can be seen from Tables 4 and 5 that the rumor detection method GLAN based on heterogeneous graph neural network and the rumor detection method KZWANG based on GCN have high rumor detection performance. However, these models are lower than the P-BiGAT model in various evaluation indicators. It can be seen that the method proposed in this paper is better than other baseline methods in the three datasets.

In addition, the method of extracting features based on deep learning is obviously better than the method of

Table 3: The results of rumor detection on Weibo dataset.

Method	Class	Acc	Prec	Recall	F1
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.847	0.830
SVM-TS	F	0.857	0.839	0.885	0.861
	T		0.878	0.830	0.857
RVNN	F	0.910	0.876	0.956	0.914
	T		0.952	0.864	0.906
PPC	F	0.921	0.896	0.962	0.923
	T		0949	0.889	0.918
GLAN	F	0.946	0.943	0.948	0.945
	T		0.949	0.943	0.946
KZWANG	F	0.950	0.945	0.954	0.949
	T		0.954	0.945	0.950
BiGCN	F	0.960	0.959	0.963	0.960
	T		0.961	0.960	0.959
P-BiGAT	F	**0.967**	**0.966**	**0.969**	**0.967**
	T		**0.970**	**0.964**	**0.966**

Bold values indicate the best experimental data among all compared baseline methods.

Table 4: The results of rumor detection on Twitter15 dataset.

Method	Acc	NRF1	FRF1	TRF1	URF1
DTC	0.454	0.733	0.355	0.317	0.415
SVM-TS	0.544	0.796	0.472	0.404	0.483
RVNN	0.723	0.682	0.758	0.821	0.654
PPC	0.842	0.811	0.875	0.818	0.790
GLAN	0.905	0.924	0.917	0.852	0.927
KZWANG	0.911	0.928	0.920	0.850	0.931
BiGCN	0.886	0.891	0.860	0.907	0.864
P-BiGAT	**0.918**	**0.932**	**0.929**	**0.930**	**0.938**

Table 5: The results of rumor detection on Twitter16 dataset.

Method	Acc	NRF1	FRF1	TRF1	URF1
DTC	0.465	0.643	0.393	0.419	0.403
SVM-TS	0.574	0.755	0.420	0.571	0.526
RVNN	0.737	0.662	0.743	0.835	0.708
PPC	0.863	0.820	0.898	0.843	0.837
GLAN	0.902	0.921	0.869	0.847	0.968
KZWANG	0.907	0.926	0.837	0.850	**0.971**
BiGCN	0.880	0.847	0.869	**0.947**	0.865
P-BiGAT	**0.913**	**0.929**	**0.915**	0.847	0.951

Bold values indicate the best experimental data among all methods.

manually extracting features; in particular, the related methods based on graph neural network have higher performance in the field of rumor detection. This is because the propagation and diffusion of rumors can be expressed by the directed path of the graph. The graph neural network is a tool that specializes in processing graph domain information, which can well extract the propagation and diffusion features of tweets on the graph. Among the related methods of graph neural network, the method[[parms resize(1),pos(50,50),size(200,200),bgcol(156)]] based on GCN and heterogeneous graph neural network show strong performance in rumor detection. Since the propagation and diffusion of rumors are actually directional, the method based on graph attention network proposed in this paper has a great improvement in performance compared with the other methods. However, the method based on graph convolutional neural network cannot handle directed graphs and cannot dynamically assign weights and can only do directed convolution on undirected graphs. Although the method based on the heterogeneous graph neural network adds user information and other ancillary information and comprehensively considers various features, the extracted propagation and diffusion features also have obvious defects, which have an impact on the final result. The method based on graph attention network has natural advantages in dealing with directed graphs, while the propagation feature and diffusion feature of rumor detection are actually based on different directions of the feature graph for node update. Compared with other graph neural network methods, graph attention network is more suitable for extracting global features based on different directions. The P-BiGAT model uses tools that specialize in directed graphs for rumor detection, improves the node update method, and effectively improves the performance of the model in rumor detection.

Compared with other methods, the P-BiGAT method proposed in this paper shows a better effect on the rumor detection problem of social platforms, which verifies the effectiveness of the P-BiGAT model.

4.5. Ablation Study

4.5.1. Analysis of the Influence of Threshold P on the Results. This article conducted a series of ablation experiments on three datasets to further explore the influence of threshold P on the results of rumor detection. The P-BiGAT model proposed in this article and the variant model BiGAT were selected for experimental comparison.

BiGAT is A variant model of this article. This model does not use the node update method based on threshold P proposed in this paper when updating nodes.

It can be seen from Figure 6 that the node update method P-BiGAT model with threshold P added has a higher accuracy in identifying rumors than the node update method BiGAT model without threshold P. This is because the P-BiGAT rumor detection model can effectively filter unimportant neighbor nodes, select neighbor nodes with larger weights for node update, filter out neighbor nodes with smaller weights, and improve accuracy. In the Weibo dataset, the amount of data is relatively large and the tree structure has many layers, so it can be seen that the BiGAT model has a significant decline in the accuracy of identifying rumors. In the Twitter15 and Twitter16 datasets, the amount of data is relatively small, and the tree structure constructed is relatively small. Therefore, it can be seen that the BiGAT model has a significant decrease in the accuracy of identifying rumors.

Studies have shown that many response tweets are not related to the source tweets, and they are still involved in node update, which greatly reduces the accuracy of identifying rumors. The P-BiGAT model proposed in this paper can effectively solve this shortcoming. In the actual application process, the larger the amount of data, the more powerful the P-BiGAT rumor detection model based on

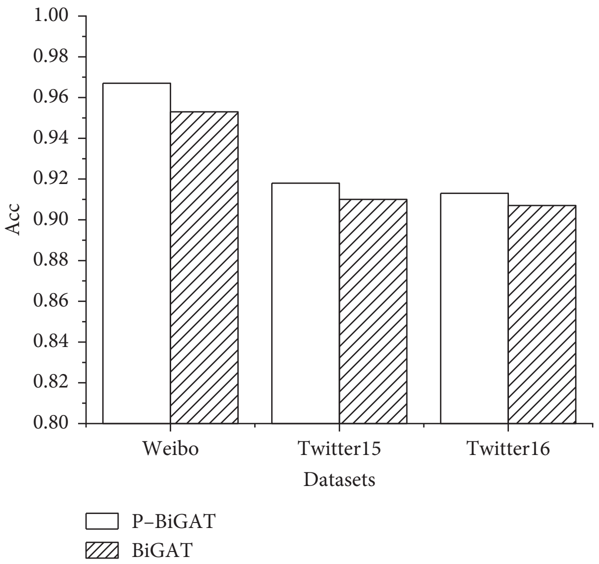


Figure 6: Comparison results of node update method based on threshold (P) on three datasets.

threshold P-based node update method can be. Constructing a rumor propagation tree and a diffusion tree with a relatively large number of layers can improve the accuracy of rumor recognition.

4.5.2. Analysis of the Influence of Different Variant Models on the Results of Rumor Detection. In order to study the influence of different parts of the model on rumor detection, this article compares the proposed P-BiGAT method with the variant models TD-GAT, BU-GAT, and w/*o* attention and w/*o* GAT were compared, using accuracy as the criterion. The experimental results are shown in Figure 7.

TD-GAT: this model only uses a top-down tree structure to update nodes to detect rumors

BU-GAT: this model only uses a bottom-up tree structure to update nodes to detect rumors

w/*o* attention: this model does not consider the semantic information extraction module and only conducts rumor detection by constructing a directed tree, extracting propagation features and diffusion features

w/*o* GAT: this model does not consider the propagation feature and the diffusion feature extraction module and only detects rumors on the semantic information representation model

It can be seen from Figure 7 that w/*o* attention is always better than TD-GAT and BU-GAT, which verifies the effectiveness of the combination of rumor spreading and walking features. Secondly, the accuracy of the w/*o* GAT variant model in identifying rumors has been significantly reduced, because the multihead attention mechanism can extract important semantic information, making rumors and nonrumors highly cohesive and low-coupling. Although the w/*o* attention model is higher than the w/*o* GAT model, the GAT model can effectively aggregate comment forwarding information and capture the relationship between the source tweet and the response tweet. However, the accuracy of identifying rumors is still lower than the method proposed in this article. Finally, the P-BiGAT model proposed in this paper is always better than all other models, verifying the importance of the propagation features of the rumor, the diffusion features, and the semantic information representation of the source tweets.

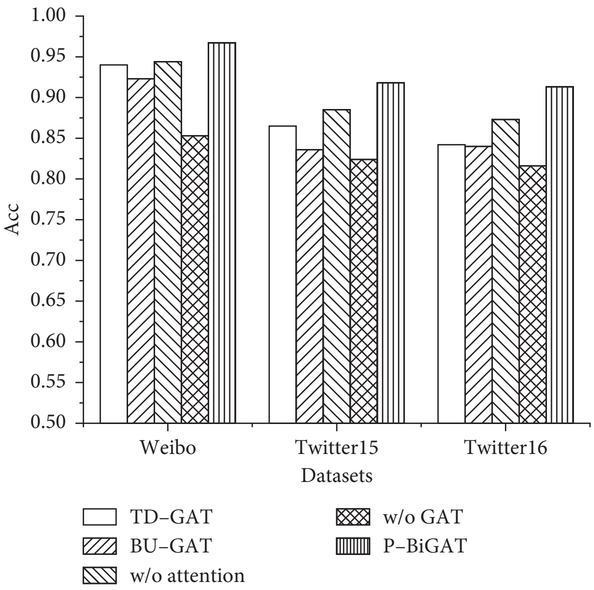


Figure 7: Rumor detection results of different variant models.

4.5.3. The Influence of Different Text Encoders on Rumor Detection Results. This article further explores the effectiveness of the BERT encoder selected by this model. The two most commonly used text encoders (word2vec and TF-IDF) will be selected to replace the BERT text encoder of this model, and the accuracy of rumor detection will be used as the criterion. The experimental results are shown in Figure 8.

In order to be able to more accurately judge the impact of different coding models on this model and achieve higher detection accuracy of the comparison group, this article uses the jieba tool to segment the experimental data and then remove the stop words and punctuation and other operations. The processed data are used as input to the word2vec and TF-IDF models, respectively. After training, the vector encoding of the text is obtained, and the BERT model is replaced to obtain the accuracy of rumor detection of different text encoders.

For word2vec, through a single-layer or multilayer neural network, each word is implied as a single word vector in one-hot form, thereby forming a dense text feature vector.

For TF-IDF, calculate the text feature vector of the word according to the number of times the word appears in the document and the frequency of the entire corpus.

As shown in Figure 8, the word2vec text encoder is better than the TF-IDF text encoder, because the word vector calculated by the word2vec model pays more attention to the

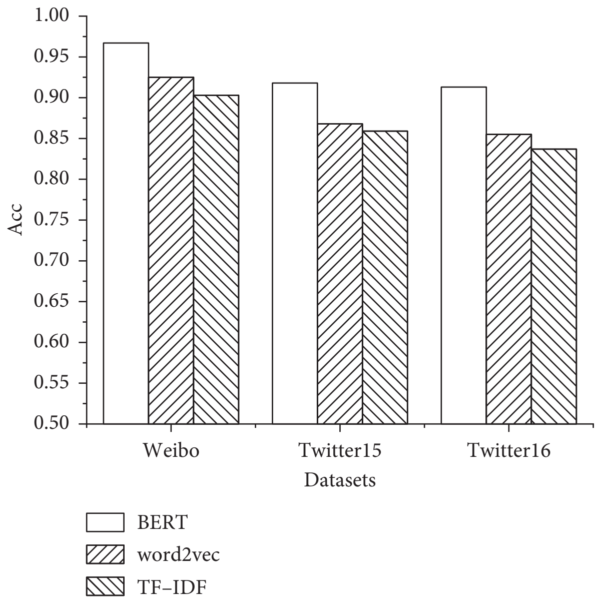


Figure 8: The influence of different text encoders on rumor detection results.

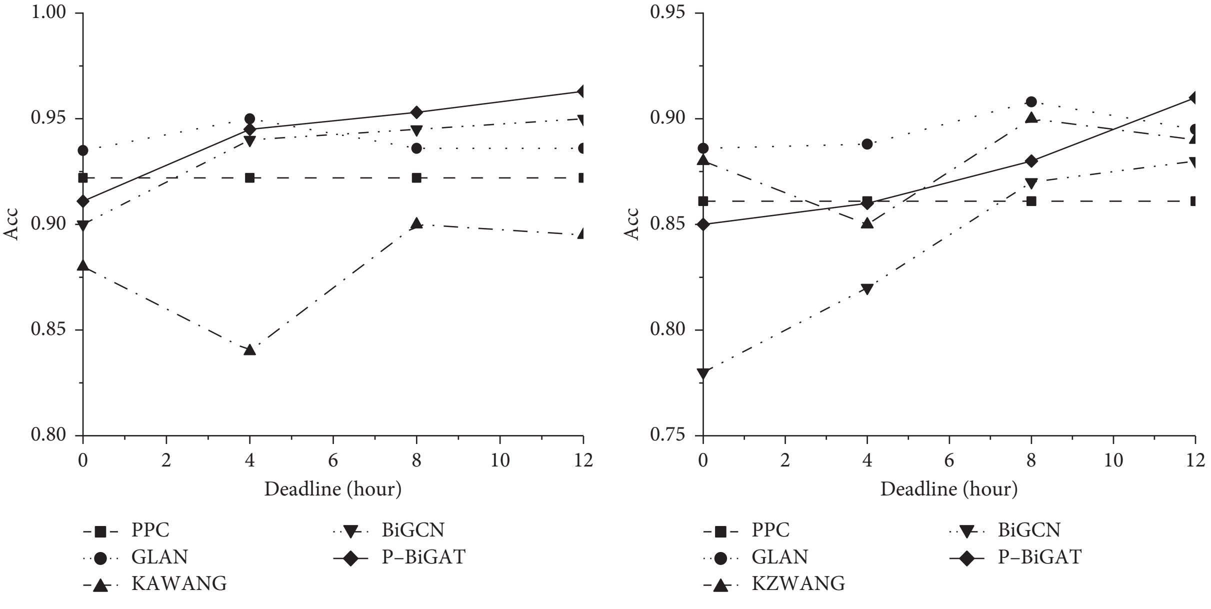
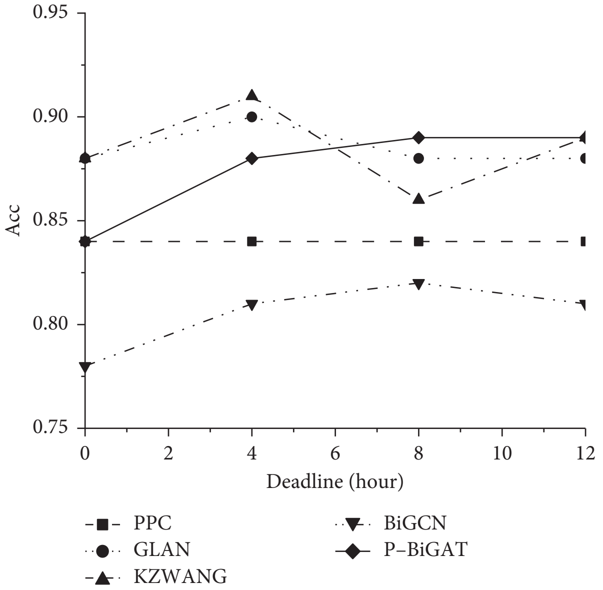


Figure 9: Continued.



(c)

Figure 9: Early rumor detection results on three datasets. (a) Weibo dataset. (b) Twitter15 dataset. (c) Twitter16 dataset.

context and semantic relationship; the TF-IDF model pays more attention to the frequency of words appearing in the document. In addition, the TF-IDF text encoder maps text to sparse feature vectors, and the word2vec text encoder maps text to dense feature vectors of fixed dimensions, which has better computational efficiency. However, whether it is the word2vec encoder or the TF-IDF encoder in this model, the accuracy of the final rumor detection is slightly lower than the BERT model encoder used in this article. Research shows that the BERT text encoder can better improve the rumor detection model of this article. This is because the BERT model uses a multilayer transformer structure to more thoroughly capture the semantic relationship in the text.

4.6. Early Rumor Detection. In order to build an early rumor spreading model, this paper sets a 12-hour deadline and compares the P-BiGAT model with the 4 best-performing baseline methods. We conducted early rumors detection on three datasets. The experimental results are shown in Figure 9.

It can be seen from Figure 9 that, within the first 4 hours, this method has an accuracy of 0.94 in the Weibo dataset, which is slightly lower than the GLAN method. As time goes up, the accuracy of P-BiGAT also increases. After the deadline hours were increased to 12 hours, the model in the early phase rumors detection accuracy is better than the others. In the Twitter15 dataset and Twitter16 dataset, the accuracy of the early rumor detection of the P-BiGAT model in the first 4 hours is lower than that of the baseline method of comparison. But when the deadline hours are increased to within 12 hours, the accuracy of the method gradually increases. This is because the amount of data within 4 hours is relatively small, and the built-up propagation tree and diffusion tree are relatively small, resulting in nodes that cannot be updated effectively. As the number of response tweets increases, the number of neighbor nodes of the node also increases, and the numbers of propagation trees and diffusion tree layers built also increase. The P-BiGAT model is based on GAT and can effectively filter out nodes with useless review information, select useful review nodes for node update, and improve the accuracy of rumor recognition.

5. Conclusions

This paper proposes a rumor detection model based on a two-way graph attention network, which comprehensively considers multiple scale features of rumors based on the premise of constructing a directed graph. In addition, in order to filter out the useless text comment information, this paper proposes a node update method, which arranges neighbor nodes according to their weights, selects nodes with higher weights, and ignores nodes with lower weights. Experiments show that the model of node update method with threshold P can effectively improve the performance of rumor detection. In the first few hours of early rumor detection, due to the relatively small number of response tweets, the node update method based on the threshold P proposed in this paper cannot play a great role, resulting in the accuracy of rumor detection and the comparison baseline method. However, with the increase of time, more and more tweets are responded to, and the amount of data is getting larger and larger. The node update method based on threshold P can select important nodes to participate in node update, so that the accuracy rate has increased significantly.

In future work, we should first consider how to conduct rumor detection more efficiently in the early stage. Then, we should integrate more user social network information, further enrich the map structure information, and consider how to use the pictures, videos, and other information contained in Weibo more efficiently. Finally, we should pay more attention to heterogeneous graphs, because heterogeneous graphs can contain multiple types of nodes, expressing feature information that is different from isomorphic graphs. Therefore, in future experiments, it is planned to encode pictures and videos to build a large heterogeneous graph network.

Data Availability

The experimental data consist of three public datasets: Weibo [8], Twitter15 [26], and Twitter16 [26].

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2017YFB0802300 and the Natural Science Foundation of Hebei Province under Grant F2021201052.

References

[1] Weibo, "Weibo Monthly Rumor Refutation Work Report," 2021, https://weibo.com/ttarticle/p/show?id=2309404637789206741299&mod=zwenzhang.

[2] C. Castillo, M. Mendoza, and B. Poblete, "Information Credibility on Twitter," in *Proceedings of the 20th International Conference on World Wide Web (WWW)*, pp. 675–684, Hyderabad, India, April2011.

[3] V. Qazvinian, E. Rosengren, D. Radev, and Q Mei, "Rumor has it: identifying misinformation in microblogs," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1589–1599, Edinburgh, UK, July 2011.

[4] K. Popat, "Assessing the credibility of claims on the web," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 735–739, Perth, Australia, April 2017.

[5] G. Xu, D. Zhou, and J. Liu, "Social Network Spam Detection Based on ALBERT and Combination of Bi-LSTM with Self-Attention," *Security and Communication Networks*, vol. 2021, Article ID 5567991, 11 pages, 2021.

[6] L. GuangJun, S. Nazir, H. U. Khan, and A. U Haq, "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms," *Security and Communication Networks*, vol. 2020, Article ID 8873639, 6 pages, 2020.

[7] F. Yang, Y. Liu, X. Yu, and M Yang, "Automatic Detection of Rumor on Sina Weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, pp. 1–7, Beijing, China, August 2012.

[8] J. Ma, W. Gao, P. Mitra, S Kwon, B. J Jansen, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3818–3824, USA, July 2016.

[9] J. Sun, "Research on the Credibility of Social Media Information Based on User Perception," *Security and Communication Networks*, vol. 2021, Article ID 5567610, 10 pages, 2021.

[10] A. Zubiaga, A. Aker, K. Bontcheva, M Liakata, and R Procter, "Detection and resolution of rumours in social media: a survey," *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–36, 2018.

[11] J. Zhou, G. Cui, S. Hu et al., "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[12] K. Wu, S. Yang, and K. Q. Zhu, "False Rumors Detection on Sina Weibo by Propagation Structures," in *Proceedings of the 2015 IEEE 31st International Conference On Data Engineering (ICDE)*, pp. 651–662, Seoul, South Korea, April 2015.

[13] W. Wang, Y. Qiu, S. Xuan, and W Yang, "Early Rumor Detection Based on Deep Recurrent Q-Learning," *Security and Communication Networks*, vol. 2021, Article ID 5569064, 13 pages, 2021.

[14] T. Chen, X. Li, H. Yin, and J. Zhang, "Call attention to rumors: deep attention based recurrent neural networks for early rumor detection, Lecture Notes in Computer Science," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 40–52, Berlin, Germany, June 2018.

[15] J. Gao, S. Han, X. Song, and F Ciravegna, "R. P.-D. N. N.: A Tweet Level Propagation Context Based Deep Neural Networks for Early Rumor Detection in Social media," 2020, https://arxiv.org/abs/2002.12683.

[16] A. Alsaeedi and M. Al-sarem, "Detecting rumors on social media based on a CNN deep learning technique," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, Article ID 10813, 2020.

[17] A. Azri, C. Favre, N. Harbi, J. Darmont, and C. Noûs, "Calling to CNN-LSTM for Rumor Detection: A Deep Multi-Channel Model for Message Veracity Classification in Microblogs," in *Proceedings of the European Conference On Machine Learning And Principles And Practice Of Knowledge Discovery In Databases (ECML PKDD 2021)*, pp. 497–513, Bilbao, Spain, September 2021.

[18] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu, "Jointly Embedding the Local and Global Relations of Heterogeneous Graph for Rumor Detection," in *Proceedings of the 2019 IEEE International Conference On Data Mining (ICDM)*, pp. 796–805, Beijing, China, September 2019.

[19] T. Bian, X. Xiao, T. Xu et al., "Rumor detection on social media with bi-directional graph convolutional networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 549–556, NY, USA, February 2020.

[20] Z. Ke, Z. Li, C. Zhou, J. Sheng, W. Silamu, and Q. Guo, "Rumor detection on social media via fused semantic information and a propagation heterogeneous graph," *Symmetry*, vol. 12, no. 11, 2020.

[21] H. Dou, W. Lingwei, Z. Wei, H. Xiaoyong, H. Jizhong, and H. Songlin, "A rumor detection approach based on multi-relational propagation tree," *Journal of Computer Research and Development*, vol. 58, no. 7, pp. 1395–1411, 2021.

[22] S. Lotfi, M. Mirzarezaee, M. Hosseinzadeh, and V. Seydi, "Detection of rumor conversations in Twitter using graph convolutional networks," *Applied Intelligence*, vol. 51, no. 7, pp. 4774–4787, 2021.

[23] N. Difonzo and P. Bordia, *Rumor Psychology: Social and Organizational Approaches*, American Psychological Association, , Tamilnadu, 2007.

[24] J. Devlin, M. W. Chang, K. Lee, and K Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, https://arxiv.org/abs/1810.04805.

[25] A. Vaswani, N. Shazeer, J. Parmar et al., “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[26] J. Ma, W. Gao, and K. F. Wong, “Detect rumors in microblog posts using propagation structure via kernel learning,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 708–717, Vancouver, Canada, July 2017.

[27] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 2014, https://arxiv.org/abs/1412.6980.

[28] J. Ma, W. Gao, Z. Wei, and Y. Lu, “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 (CIKM)*, pp. 1751–1754, Melbourne, Australia, October 2015.

[29] Y. Liu and Y. F. B. Wu, “Early Detection of Fake News on Social media through Propagation Path Classification with Recurrent and Convolutional Networks,” in *Proceedings of the Thirty-Second AAAI Conference On Artificial Intelligence*, pp. 354–361, New Orleans, Louisiana, USA, December 2018.

[30] J. Ma, W. Gao, and K. F. Wong, “Rumor detection on twitter with tree-structured recursive neural networks,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 1980–1989, Melbourne, Australia, July 2018.