

Research Article

GCNRDM: A Social Network Rumor Detection Method Based on Graph Convolutional Network in Mobile Computing

Dawei Xu ^{1,2}, Qing Liu,² Liehuang Zhu,¹ Zhonghua Tan,³ Feng Gao ¹ and Jian Zhao ²

¹School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China

²College of Cybersecurity, Changchun University, Changchun 130022, China

³College of International Education, Hainan Normal University, Haikou 571000, China

Correspondence should be addressed to Dawei Xu; xudw@ccu.edu.cn

Received 19 June 2021; Accepted 13 September 2021; Published 8 October 2021

Academic Editor: Lihua Yin

Copyright © 2021 Dawei Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile computing is a new technology emerging with the development of mobile communication, Internet, database, distributed computing, and other technologies. Mobile computing technology will enable computers or other information intelligent terminal devices to realize data transmission and resource sharing in the wireless environment. Its role is to bring useful, accurate, and timely information to any customer at anytime, anywhere, and to change the way people live and work. In mobile computing environment, a lot of Internet rumors hidden among the huge amounts of information communication network can cause harm to society and people's life; this paper proposes a model of social network rumor detection based on convolution networks, the use of adjacency matrix between the nodes represent user and the relationship between the constructions of social network topology. We use a high-order graph neural network (K-GNN) to extract the rumor posting features. At the same time, the graph attention network (GAT) is used to extract the association features of other nodes of the network topology. The experimental results show that the method of the detection model in this paper improves the accuracy of prediction classification compared with deep learning methods such as RNN, GRU, and attention mechanism. The innovation of the paper proposes a rumor detection model based on the graph convolutional network, which lies in considering the propagation structure among users. It has a strong practical value.

1. Introduction

In the 5G communication network environment, the number of data transmission is increasing, and there are many types of data. Different types of data storage methods are different. Therefore, it is difficult to collect, store, analyze, and query big data. The commonly used big data analysis and collection technology cannot meet the development needs of all walks of life when applied. When the technology is improved and optimized, the appropriate data mining algorithm should be selected to extract the effective information. After the analysis of the mined big data, it should be presented to users in the form of visualization of data charts, and it should be evaluated quantitatively. In order to further improve the existing data mining technology, we can auto-

matically extract relevant information from valid data through the artificial intelligence algorithm and semantic search engine design, so as to improve the ability of data collection and screening.

As mobile computing expands into every aspect of our lives, arm-based smartphones, laptops, wearables, and other smart devices are everywhere. The number of compute-intensive use cases for these devices is rising every year, performing tasks we could only dream of in the past. Mobile computing is a new technology that covers many disciplines and a wide range. It appears with the development of mobile communication, Internet, database, distributed computing, and other technologies. Mobile computing technology will enable computers or other information intelligent terminal devices to realize data transmission and resource sharing in

the wireless environment. Its role is to bring useful, accurate, and timely information to any customer at anytime, anywhere, and to change the way people live and work.

The internet technique has been rapidly developed for the recent half-century, which causes the social media to become a convenient online platform for users to obtain information, express their opinions, and communicate. Increasingly, people are eager to participate in the discussion on some hot topics and exchange their points via social media. Therefore, some false information has been disseminated [1]. Due to the scale of social media users being large and the information on social media being easy to access for everyone, rumors can be spread rapidly in a nuclear fission manner through social media, which often triggers many instability factors and makes a great impact on economy and society. Therefore, it is particularly urgent to identify rumors on social media effectively and early to deal with the panic and threat.

Traditional rumor detection methods mainly rely on semisupervised learning of automatically labeled features, such as user features, message content features, microblog topic features, location information features, and network-type features [2]. But the abovementioned feature extraction methods are not only time-consuming and labor-intensive but also their extracted feature information is insufficient, while such methods fail to reflect the deep social network topology. Hence, they are not sufficient to judge rumors.

Since traditional machine learning rumor detection methods result in the aforementioned drawbacks, researchers have been conducting some researches about deep learning methods into introducing a rumor detection model in recent years. Typical deep learning models include recurrent neural networks (RNN), gated recurrent units (GRU), and recurrent neural networks [3]. Although these methods are able to learn time series features from rumor propagation, they ignore the effects of rumor propagation for the reason that their temporal structure features only focus on the serial propagation of rumors.

The graph attention network (GAT) is a new type of convolutional neural network which operate on graph-structured data using hidden self-attentive layers. The graph attention layer used in GAT is computationally efficient (does not require complex matrix operations with parallel computation over all nodes in the graph). In GAT, each node in the graph can be assigned a different weight based on the characteristics of its neighbors and does not rely on prior knowledge of the entire graph structure. It allows the model to reduce a large amount of physical memory during intermediate computations and enhance the efficiency of the operation of model.

Another graph neural network, K-GNN, is a generalization of GNN based on k-WL. This new model is stronger than GNN in distinguishing nonisomorphic (sub)graphs and is able to distinguish more graph attributes [4]. Triangle counting is an algorithm for counting graph structures, where the number of triangles indicates the degree of association and the tightness of organization of nodes in graph. This counting method is often used as an identification method for social network topological graphs, which can

distinguish the properties of graph structures. K-GNN has better results for triangle counting problems. Therefore, in this work, we adopt K-GNN as a convolutional layer.

In order to mine the difference between rumor and non-rumor implied layer structure features with better performance, we propose a two-layer graph convolutional attention network, which obtains the propagation and dispersion properties through two parts of top-down and bottom-up GAT, respectively [5]. K-GNN obtains the information of the parent node of a node in the rumor tree. GAT aggregates the information of the children of a node in the rumor tree. Then, the propagation and dispersion representations converged at the embedding of K-GNN and GAT are combined by full concatenation to obtain the final result. Meanwhile, we connect the root features of the rumor tree with the hidden features of each graph convolution layer to enhance the influence of the root of the rumor. In addition, we use drop edge [6] in the training phase to avoid the overfitting problem of the model. The main contributions of this paper are as follows:

- (1) In this paper, we solve the obstacle of traditional convolution methods when it cannot extract structural features of social networks by applying the graph convolution method for rumor detection
- (2) For the problem of extracting the effective feature of different data, we obtain the more effective features by using two-layer graph convolution GAT and K-GNN as the implicit layer
- (3) The detection is performed on two public datasets. The experimental results show that the results are better than the accuracy of existing schemes

2. Related Work

Automatic detection of rumors on social media has attracted a large amount of attention in recent years. Previous works on rumor detection focus on extracting rumor features from text content, user configurations and propagation structures, learning classifiers from labeled data [7–11], etc. Jing et al. [12] used time series to classify rumors, simulating changes in handcrafted social context features. Yin et al. [13] combined RBF kernels with random traversal-based graph kernels to propose a graph kernel-based hybrid SVM classifier. Ma et al. [14] constructed a rumor propagation tree kernel to detect rumors by evaluating the similarity between rumor propagation tree structures. The aforementioned works are less efficient and heavily rely on manual feature engineering to extract information feature sets.

To implement the automatic learning of high-level features, there are some rumor detection methods based on deep learning models that have been proposed recently. Yu et al. used recurrent neural networks (RNNs) to capture hidden representations from temporal content features [15]. Tong et al. [16] improved this approach by combining attention mechanisms with RNNs in order to text features with different attention. Su et al. [17] proposed a convolutional neural network- (CNN-) based approach to learn key

features scattered in the input sequence form high-level interactions between important features. Ke et al. [18] combined RNN and CNN to obtain user features based on time series. Recently, Ma et al. [19] used an adversarial learning approach to improve the performance of a rumor classifier, where the discriminator acts as a classifier and the corresponding generator based on the design of the generative model improves the discriminator by generating conflicting noise. In addition, Ma et al. constructed a tree-structured recurrent neural network (RNN) to capture the hidden representation of the propagation structure and text content [20]. However, these methods are less efficient in learning the structural features of the spread of rumors and ignore the global structural features of rumor propagation.

Compared with the deep learning models, GCN captures global structural features from graphs or trees. Su et al. [21] theoretically analyzed a graph convolution method for undirected graphs based on spectral graph theory. Subsequently, Defferrard et al. [22] developed a method called Chebyshev spectral CNN and used Chebyshev polynomials as filters. Kipf and Welling [23] proposed a new semisupervised classification method based on graph structure data. Based on the GCN model, researchers used an efficient hierarchical propagation rule, which is based on a first-order approximation of the spectral convolution on the graph. Experiments on a large number of network datasets show that the proposed GCN model is able to encode graph structure and node features in a way that facilitates semisupervised classification. After that, Veličković et al. [24] proposed the graph attention network (GAT), which operates on graph-structured data and utilizes a hidden self-attention layer allowing different importance (implicitly) assigned to different nodes. In the process of rumor propagation, it is often the important nodes of social networks that play a key role. GAT can increase the weight of important nodes, so the rumor detection process can be convolved to dig out the implied malicious nodes. We can use the Weisfeiler-Leman algorithm [25] to determine if two graphs have the same structure. In the rumor spreading process, emotional rumors are able to make rumor audiences produce similar positive and negative emotions through emotional infection, and under the influence of emotions, audiences lack rational analysis of information, thus increasing the forwarding of rumors. The emotions of rumor audiences play a mediating effect in rumor forwarding. We think that the structure of the feature map extracted between rumors should be isomorphic. Morris et al. [4] in 2019 proposed a k -order GNN based on the ensemble k -WL algorithm. Thus, we use GAT, K -GNN double-layer convolution for the rumor detection process.

3. Preliminary

3.1. Graph Attention Network. Recently, there has been an increasing interest in extending convolution to the graph domain. Graph convolution (GCN) is the first proposed model, of which the convolution operation is considered as a general “message passing” architecture, as follows:

$$H_k = M(A, H_{k-1}; W_{K-1}), \quad (1)$$

where $H_k \in \mathbb{R}^{n \times v_k}$ is the hidden feature matrix computed by the graph convolution layer and M is the message propagation function depending on the adjacency matrix A . H_{k-1} and W_{K-1} are the hidden layer feature matrix and the parameters for training, respectively.

Veličković et al. [24] proposed the graph attention network (GAT), which for each node implements a self-attentive mechanism. The attention correlation coefficient is

$$e_{ij} = a\left(W\vec{h}_i, W\vec{h}_j\right), \quad (2)$$

where e_{ij} is the attention correlation coefficient between node i and node j ; W is the matrix parameter for training; \vec{h}_i and \vec{h}_j are the feature vectors of nodes i and j , respectively; and the corresponding weights are assigned to different neighboring nodes without either matrix operations or prior knowledge of the graph structure. For simplifying the calculation and comparison among correlation coefficients, softmax is introduced to regularize all neighboring nodes as follows:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}. \quad (3)$$

The attention mechanism α is a single-layer feedforward neural network added LeakyRelu nonlinear activation with $\vec{a} \in \mathbb{R}^{2F'}$ determined by the weight vector. Thus, we finally obtained the attention activation function as follows:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyRelu}\left(\vec{a}^T \left[W\vec{h}_i \parallel W\vec{h}_j \right] \right)\right)}{\sum_{k \in N_i} \exp\left(\text{LeakyRelu}\left(\vec{a}^T \left[W\vec{h}_i \parallel W\vec{h}_k \right] \right)\right)}. \quad (4)$$

3.2. K -GNN Convolutional Layer. The Weisfeiler-Leman algorithm [25] is used to determine whether two graphs are isomorphic, and the basic idea is to determine the independence of the current central node by iteratively aggregating the information of neighboring nodes to update the coded representation of the whole graph with the following updated formula:

$$c_l^{(t)}(v) = \text{HASH}\left(\left(c_l^{(t-1)}(v), \left\{c_l^{(t-1)}(u) \mid u \in N(v)\right\}\right)\right), \quad (5)$$

where HASH is a mapping of graph structure nodes. By executing the above function on two graphs, one can determine whether the two graphs are isomorphic.

The GNN-based base model [4] can be implemented by the following equation:

$$f^{(t)}(v) = \sigma\left(f^{(t-1)}(v) \bullet W_1^{(t)} + \sum_{\omega \in N(v)} f^{(t-1)}(\omega) \bullet W_2^{(t)}\right). \quad (6)$$

In each layer, we compute a new eigenvector $\mathbb{R}^{1 \times e}$ for node v . $W_1^{(t)}, W_2^{(t)}$ is the matrix of weight parameters updated by $\mathbb{R}^{d \times e}$, and σ is a nonlinear activation function, such as rectified linear unit (ReLU) or Sigmoid.

According to the work of Gilmer et al. [26], it is also possible to replace the summation defined on the neighborhood in the above equations by a substitution invariant differentiable function, or to replace the external summation by a column vector tandem or LSTM-style update step. Thus, in the fully general case, the computation of a new identity $f^{(t)}(v)$ can be expressed as

$$f_{\text{merge}}^{W_1} \left(f^{(t-1)}(v), f_{\text{aggr}}^{W_2}(v), \left\{ f^{(t-1)}(\omega) \mid \omega \in \mathbb{N}(v) \right\} \right), \quad (7)$$

where $f_{\text{aggr}}^{W_2}$ aggregates the features of neighborhood nodes and $f_{\text{merge}}^{W_1}$ aggregates the neighborhood features representation calculated in the previous step. We can analogize $f_{\text{merge}}^{W_1}$ and $f_{\text{aggr}}^{W_2}$ to $W_1^{(t)}$ and $W_2^{(t)}$ of the GNN base formula. Then, we can conclude that there exists a specific set of GNN models whose effects are fully equal to the Weisfeiler-Leman algorithm (WL algorithm).

Drawing on the expansion of first-order WL to higher-order WL, the GNN is expanded to K-GNN by the following equation:

$$f_{k,L}^{(t)}(s) = \sigma \left(f_{k,L}^{(t-1)}(s) \cdot W_1^{(t)} + \sum_{\omega \in \mathbb{N}_L(s)} f_{k,L}^{(t-1)}(\omega) \cdot W_2^{(t)} \right), \quad (8)$$

where s denotes the subgraph consisting of k nodes and u is the neighboring subgraph of this subgraph, for a given k , we consider all k -element subsets $[V(G)]^k$ over $V(G)$. Let $s = \{s_1, \dots, s_k\}$ be a k -set in $[V(G)]^k$, then, we define the neighborhood of s as follows:

$$\mathbb{N}(s) = \left\{ t \in [V(G)]^k, |s \cap t| = k - 1 \right\}. \quad (9)$$

That is, a subgraph consisting of k nodes must have and only $k - 1$ common nodes in its neighboring subgraphs. With such an idea in mind, we can consider more higher-order information sets when modeling tasks with multilayer graph structures like social networks:

3.3. DropEdge. DropEdge is a new method to reduce overfitting of training models based on graph convolutional networks [27]; in each training cycle, some edges are randomly removed from the input graph and different deformation structures are generated at a certain rate, as shown in Figure 1. Thus, this method increases the randomness and diversity of the input data. Assuming that the total number of edges in graph A is N_e and the drop rate is set to p , the adjacency matrix after DropEdge calculation is shown as follows:

$$A' = A - A_{\text{drop}}, \quad (10)$$

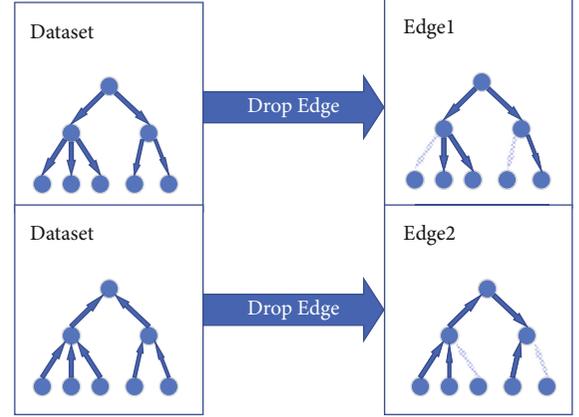


FIGURE 1: Schematic diagram of DropEdge process.

where A_{drop} is a matrix constructed using $N_e \times p$ edges randomly sampled from the original edges.

4. Model

4.1. Rumor. Rumor has three different meanings: words fabricated without the existence of facts, unacknowledged legends, and words circulated by folk to comment on current affairs [28]. The research of this paper is social network rumors, which refer to rumors spread through online media (e.g., microblogs, foreign websites, online forums, social networking sites, and chat software) without factual basis and offensive, purposeful discourse. They are mainly related to emergencies, public health, food and drug safety, political figures, subversion of tradition, and deviance.

Rumors spread suddenly and quickly and therefore have a negative impact on the normal social order. The rumor mill is not able to prevent the spread of rumors because of the misappropriation of concepts and generalization; the herd mentality accelerates the spread of rumors because it is better to believe in them than not to believe in them. Internet rumors, especially political rumors, can easily cause serious social problems and even social unrest and political instability due to their indistinguishability and confusing nature [29]. Many countries have made combating online political rumors an important part of rumor management and have taken comprehensive measures to crack down on them.

4.2. Social Network Rumor Detection. The current mainstream approaches treat social network rumor detection as a dichotomous problem, which is formally defined as follows.

The tweets in the social network are treated as a set $P = \{p_1, p_2, p_3, \dots, p_i\}$, where p_i represents a tweet. Each tweet is given a label $L = \{l_1, l_2\}$, where l_1 and l_2 represent rumor and nonrumor, respectively. The task of social network rumor detection is to learn a classifier model M that maps tweet p_i into a category label l_j . The input of the model is an event containing several tweets, and the output is the rumor or nonrumor label corresponding to the event.

The social network rumor detection usually includes four stages: data processing, feature selection and extraction, model training, and rumor detection.

Data processing includes the collection of raw data and data annotation. The purpose of data collection is twofold: one is to build a dataset for training models and another is to monitor and obtain information to be detected, such as user interaction information. Data annotation is to label the data according to different needs. The data mostly is labeled as rumors or nonrumors. The experimental data in this paper are derived from two publicly available datasets. The datasets have been annotated with the correctness of each data item, and the user interaction information can be extracted from the datasets.

Feature selection and feature extraction is to select and construct the set of feature vectors that represent the data from the collected raw data optimally. For machine learning methods, feature selection and extraction are even more important than model selection. Therefore, the important work based on the machine learning method is to find more effective features to improve the accuracy of rumor detection. Rumor detection based on deep learning has a strong feature learning capability, which can obtain more high-dimensional, complex, and abstract feature data than traditional machine learning without manual feature extraction. In this paper, the signs are extracted by top- k topic word selection, and then, feature vectors are constructed based on whether the word occurs in a sentence. Although this method seems relatively simple, the complex selection of feature vectors is easy to over fit the later model training.

Model training refers to the process of selecting a model from existing classification models according to a specific problem scenario and adjusting the parameters to find an optimal model based on the classification performance of the model on the training dataset. For the social network rumor problem, it is the toughest challenge to train a classifier with accuracy in the massive data which is full of noise and still unbalanced. The main part of the model training in this paper is to adjust the parameters. The adjustment of different parameters will make different effects on the model and deploy the next parameter adjustment based on the model's performance solution.

Rumor detection is to identify the information authenticity of the information spread in social networks based on the rumor classifier obtained from model training. Our goal is to build a binary classifier to determine if a sentence is a rumor or not a rumor.

4.3. Symbols. In the following, the notation used in the model of this paper will be defined uniformly.

Let $C = \{c_1, c_2, \dots, c_m\}$ be the rumor dataset, c_i be the i th tweet, and M be the total number of tweets. $C_i = \{r_i, w_1^i, w_2^i, \dots, w_{n_i-1}^i, G_i\}$, where n_i is the reply or retweet of c_i tweets, and r_i is the source post tweets. Each w_j^i denotes the j th relevant reply or retweet tweet, and G_i is the propagation structure of the tweet. For G_i is defined as a graph structure $\langle V_i, E_i \rangle$ [13, 14], r_i as the root node, $V_i = \{r_i, w_1^i, w_2^i, \dots, w_{n_i-1}^i\}$ and $E_i = \{e_{st}^i \mid s, t = 0, \dots, n_i - 1\}$ denotes the

set of edges from the replied post to the forwarded post or the replied post, for example, suppose w_2^i has a response to w_1^i , then there exists a directed edge $w_1^i \longrightarrow w_2^i$, which is e_{12}^i , and if w_1^i has a response to r_i , then there exists a directed edge $r_i \longrightarrow w_1^i$, which is e_{01}^i . Define $A_i \in \{0, 1\}^{n_i \times n_i}$ as the adjacency matrix, where

$$a_{st}^i = \begin{cases} 1, & \text{if } e_{st}^i \in E_i, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Define $X_i = [x_0^i, x_1^i, \dots, x_{n_i-1}^i]^T$ as a feature matrix from c_i , where x_0^i denotes the feature vector of r_i and each x_j^i denotes the feature vector of the corresponding row w_j^i .

Moreover, each source-posted tweet is associated with a real label $y_i \in \{F, T\}$, and the goal of rumor detection is to learn a classifier that

$$f = C \longrightarrow Y, \quad (12)$$

where C and Y are the set of events and labels, respectively, and the labels of the event are predicted based on the textual content, user information, and propagation structure constructed from the related posts of the event.

4.4. My Model. In this subsection, the rumor detection model proposed in this paper will be described. The core idea is to extract features from root rumors by the K-GNN layer and obtain more features of neighboring subgraphs with GAT layer. We call this model GAT_GNN. My model in this paper is shown in Figure 2.

We first discuss how to apply the GAT_GNN model to one event; X denotes the original feature matrix input to the GAT_GNN model and Edge1 and Edge2 are the matrices obtained after DropEdge processing. $X1_{\text{root}}$ and $X2_{\text{root}}$ is the first row of the $X1$ matrix and $X2$ matrix, After convolving $X1$ and $X2$ in two layers, we get $Y1$ and $Y2$. The final detection result is obtained by putting the matrix of $Y1$ and $Y2$ stitching into the binary classifier FC.

We can obtain in the rumor dataset the information of the original text and its retweets and comments. We integrate all the texts into a text database; firstly, we use jieba to split the words of this database and then use top- k to extract 5000 high-frequency words. Each rumor (root) and its forwarding and commenting message is represented by a vector of 5000 rows, each column represents a word, and the word is recorded as n if it appears n times in the text, and as 0 if it does not appear in the text message.

Each rumor and its forwarded comments form subgraph G_i , which is defined as a graph structure $\langle V_i, E_i \rangle$, and V_i is an $n \times d$ matrix, where n is the total number of users who post rumor information and its forwarded comments, and d is the number of feature vectors introduced above, and the comparison experiments show that the experimental results obtained by taking 5000 for d are better. The first row of V_i is the feature vector of rumor posting users, and we will mark this row of each subgraph to facilitate the subsequent part of the model to read this vector.

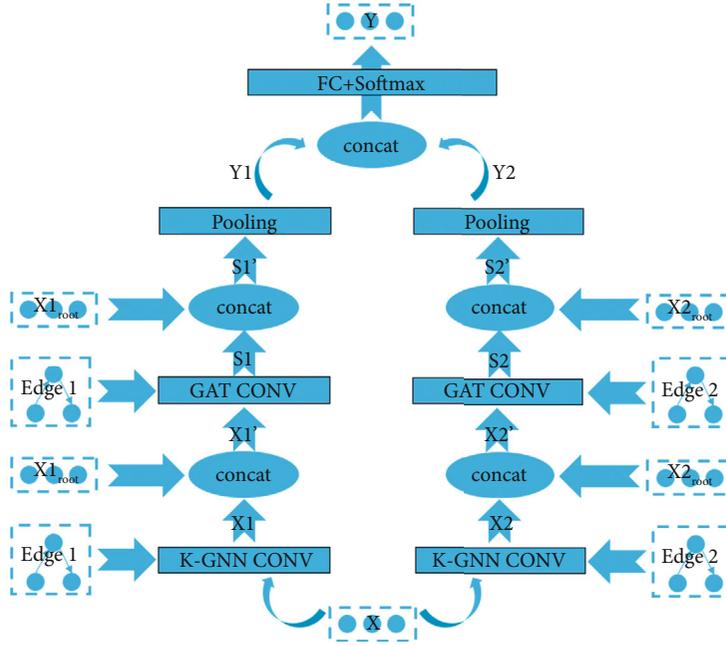


FIGURE 2: GAT_GNN rumor detection model.

The user information in the subgraph is extracted to get E_i . Each node represents a user, where the source node of the published rumor is used as the root node. The root node has no parent node, and the parent and child nodes of a node can be obtained from the dataset (for example, in set of microblog review, the blogger is parent node and the reviewer are child nodes.) We take the directed graph of the parent node pointing to the child nodes as $EdgeD$, and the child node pointing to the parent node is $EdgeU$. We do not consider the relationship between user nodes under different rumors, but only consider the relationship of each node in a subgraph. The rumor dataset includes a large number of roots to child relationship and a small number of child-to-child relationship. A large number of users choose to review and transponder, which causes unbalanced distribution of samples in the dataset. Therefore, the subgraph does not have a deep hierarchy. For this feature, we reduce the percentage of P edges by Equation (10) to generate two new adjacency matrices $Edge1$ and $Edge2$, which can avoid the overfitting problem of the model.

The detection model we built is a binary prediction of the root rumor information. The information of rumors and retweeted comments are used to construct the subgraph G_i , which also translates into a binary classification problem for the subgraph G_i . In the following, we present the whole model.

First, $Edge1$ and X are put into the K-GNN CONV layer for convolution, and the formula used is Equation (6) above. The original 5000 features of each node are extracted to 32; a large number of features are not conducive to node classification; we use this convolution for feature compression to obtain $X1$. Then, we extract the first row of $X1$ to note as $X1_{root}$, which is the result of compressed features of root rumor. In order to enhance the impact of the root rumor,

we splice the feature vectors of the compressed root rumor into each original feature by using the horizontal splicing, which only increases the number of features and does not change the number of nodes. Each subgraph becomes a matrix H_k with $N \times 5032$. This new feature matrix is called $X1$. So, we can still use $Edge1$ as the adjacency matrix for calculation. The formula for the splicing process is shown below:

$$H_k = \text{concat}(X1, X1_{root}). \quad (13)$$

After that we perform the second layer of convolution by putting $Edge1$ and $X1'$ into the GAT layer for convolution, and the formulas used are those in Equations (1)–(4) above. Here, the 5032 features of each node are compressed to 32, and the $N \times 32$ feature matrix $S1$ is obtained, in order to further enhance the effect of root rumors. We splice $X1_{root}$ with $S1$ horizontally again, which is used to increase the number of features, to obtain $N \times 64$ feature matrix $S1'$. The formula for the splicing process is shown below:

$$S1' = \text{concat}(S1, X1_{root}). \quad (14)$$

Finally, using mean pooling, N rows in $S1'$ are turned into 1 row to obtain a 1×64 eigenvector to represent a rumor subgraph $Y1$. In exactly the same way, $Edge2$ is processed with the data according to the above steps to obtain $Y2$, which should also be a 1×64 eigenvector, calculated as

$$\begin{aligned} Y_1 &= \text{MEAN}(S1'), \\ Y_2 &= \text{MEAN}(S2'). \end{aligned} \quad (15)$$

The two information representations are then combined:

$$W = \text{concat}(Y_1, Y_2). \quad (16)$$

Finally, the label \hat{y} of event y is calculated by the fully connected layer and softmax:

$$\hat{y} = \text{softmax}(FC(W)), \quad (17)$$

where $\hat{y} \in \mathbb{R}^{1 \times C}$ is the probability vector used to predict all classes of event labels. In this experiment, the model parameters of this paper are trained by minimizing the cross-entropy through the real distribution of labels. The L_2 regularizer is used in the loss function of all model parameters.

5. Experiment

The performance of the model proposed in this paper is first empirically evaluated, then compared with several other baseline models. Finally, the ability of the method in this paper verified for other rumor-type detections.

The datasets chosen for the experiments in this paper are the publicly available datasets Chinese_Rumor_Dataset [30], Twitter15, and Twitter16 [31]. In the experimental dataset, nodes represent users, edges represent retweet and response relationships, and features are extracted from text messages after data processing in TF-IDF values of top 5000 words. The Twitter dataset contains two tags, namely, false rumors (F) and true rumors (T). the Twitter15 and Twitter16 datasets contain four tags: nonrumor (N), false rumor (F), true rumor (T), and unconfirmed rumor (U). Each event in Weibo is labeled according to the Sina Community Management Center, which reports all kinds of false information. Each event in Twitter15 and Twitter16 are labeled according to the authenticity labels of articles in disinformation sites (e.g., <http://snopes.com>, <http://Emergent.info>). The statistical results of the three datasets are shown in Table 1.

5.1. Contrasting Models. We compare the proposed approach with some state-of-the-art baseline models, including the following models:

- (i) DTC [32]: a rumor detection method that uses decision tree classifiers based on various handcrafted features to obtain information credibility
- (ii) SVM-TS [33]: a linear SVM classifier that uses handcrafted features to construct a time series model
- (iii) GRU [34]: a RNN-based model that learns temporal linguistic patterns from user comments
- (iv) cPTK [35]: a SVM classifier based on propagation tree kernels is proposed based on the propagation structure of rumors
- (v) RvNN [36]: a rumor detection method based on the tree recurrent neural network with GRU units,

TABLE 1: Dataset statistics.

Statistics	Weibo	Twitter15	Twitter16
Posting	2062501	331612	204820
User	1265387	276663	173487
Events	3387	1490	818
True rumor (T)	1538	374	205
False rumors (F)	1849	370	205
Unconfirmed rumors (U)	0	374	203
Not a rumor (N)	0	372	205
Maximum number of post retweets	56155	1768	2765
Minimum number of post retweets	5	55	81
Length of time (hours)	68064	1337	848

which learns rumor representation by the propagation structure

- (vi) PPC_RNN+CNN [37]: a rumor detection model combining RNN and CNN, which learns rumor representation by the user's features in the rumor propagation path
- (vii) GAT_GNN [38]: constructs a generalized network rumor detection model based on the GAT and GNN layers using a bidirectional propagation structure

For a fair comparison, we randomly divide the dataset into 5 parts and perform a 5-fold cross-test to obtain more stable results. On this dataset, this paper evaluates the accuracy (Acc.), recall (Rec.), and F1 value (F1) of each classification. The stochastic gradient descent algorithm was used to update the model parameters, and the model was optimized using Adam's algorithm [39]. The dimensionality of the feature vector hidden by each node was 64. The parameter of DropEdge was 0.1, and the parameter of dropout was 0.5. The training process was iterated for 30 cycles and applied to the early stop when the test loss stop was reduced by 5 cycles.

5.2. Experimental Results. Figures 3 and 4 give the performance of the methods in this paper and all comparative methods on the Twitter and Weibo datasets, respectively. First, in the benchmark algorithm, we observe that the deep learning method performs significantly better than those using handcrafted features. This is because deep learning methods are able to learn the high-level representations of rumors to capture effective features. This illustrates the importance and necessity of studying deep learning for rumor detection. Second, the method in this paper outperforms the PPC RNN+CNN method in all performance metrics, demonstrating the effectiveness of introducing discrete structures for rumor detection. Since RNNs and CNNs cannot process data with graph structure, PPC RNN+CNN

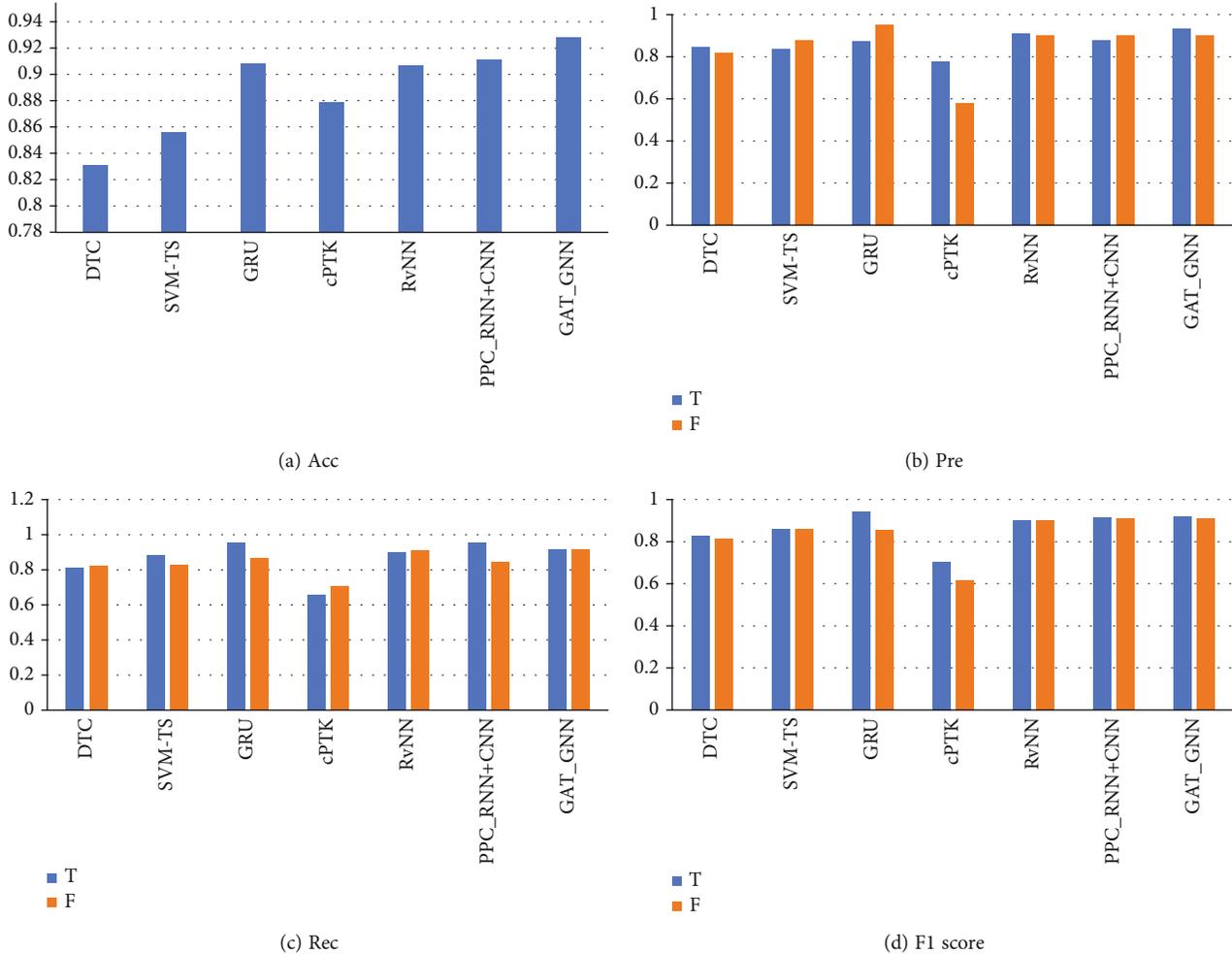


FIGURE 3: Rumor detection results of Weibo dataset (F: false rumor; T: true rumor).

ignores the important structural features of rumor scattering. This makes it impossible to obtain an efficient high-level representation of rumors, which leads to poor performance of rumor detection. Finally, the GAT_GNN method clearly outperforms the RvNN method. Since RvNN only uses the hidden feature vectors of all leaf nodes, it is heavily influenced by the information of the latest posts. However, the latest posts are always missing information such as comments and just follow the previous posts. Unlike RvNN, root feature augmentation makes the proposed method more focused on the information of source posts, which helps to further improve our model.

6. Discussion

Our solution is compared with other solutions in 4 aspects: accuracy, recall, precision, and F1 value. In the Weibo dataset, the GAT-GNN model is 1.2 percentage points higher in accuracy than the highest solution among other models. Some models, such as GRU, have higher recall and precision than us in the T classification but do not perform well in the F classification. In rumor detection, the classification of F,

which we define as untrue rumors, is more important. This is because the goal of automated rumor detection is to save labor costs, nonrumors still account for a large proportion of text messages in the entire social network. Our model of GAT-GNN has a higher classification accuracy and performs more consistently in other judging metrics. This facilitates the application of the model to real-world detection scenarios.

While in the Twitter dataset, our model performs better relative to other models. Twitter is a four-category dataset, so we only compared the accuracy relationships under each category. Although there are some models such as cPTK, GRU, and RvNN that can be close to our judgment in a certain class of rumors, they do not perform as well in other classes of rumors. Since rumors are time-sensitive, multicategorizing rumor detection helps us to analyze where the spread will go next. Therefore, our model obtains a relatively good performance in the problem of four classifications; it depends on the ability of bilayer graph convolution to analyze complex problems. The stability of our model in experiments also paves the way for systematizing automatic rumor detection.

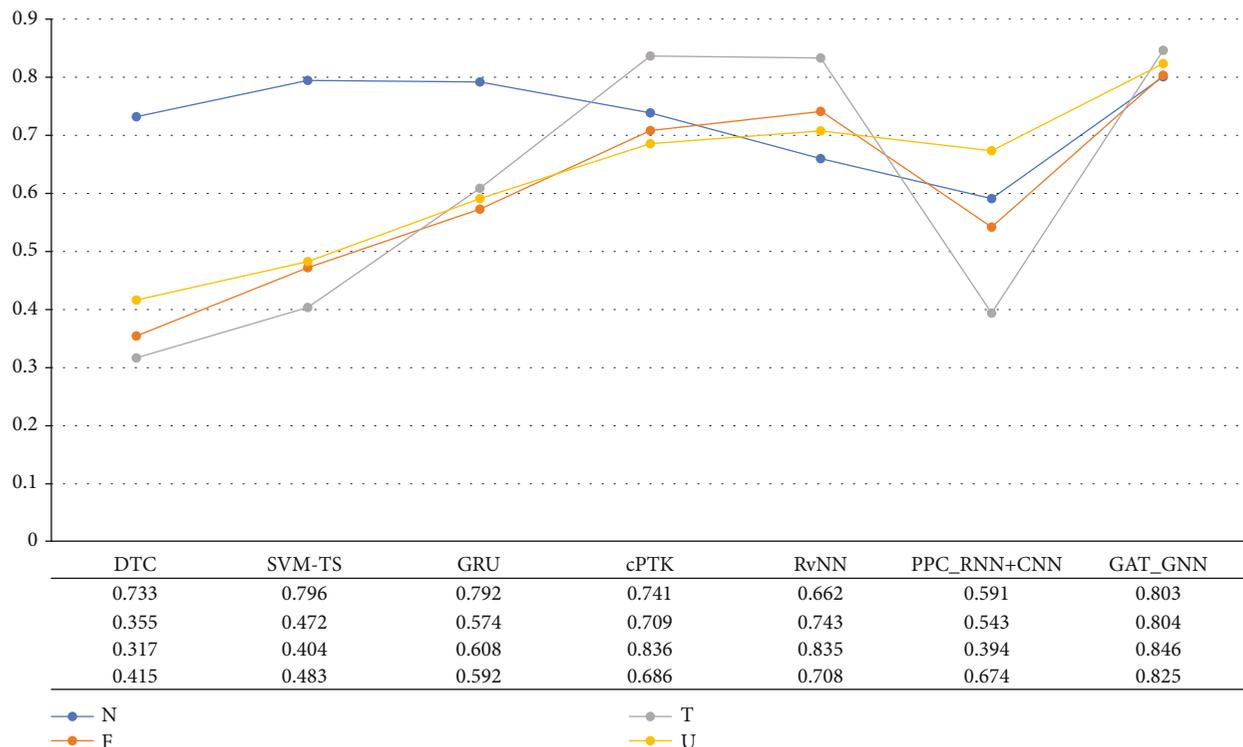


FIGURE 4: Rumor detection results on Twitter15 and Twitter16 datasets (N: non-rumor; F: false rumor; T: true rumor; U: unconfirmed rumor).

7. Conclusion and Future Work

In this paper, we propose a social media rumor detection model based on GAT and K-GNN, called GAT_GNN, where the graph convolution model has the ability to handle graph or tree structures, making the model more conducive to represent deeper topological networks. Also, the parent-to-child node connectivity relationship is used to model the propagation pattern. Experimental results on two real datasets show that the GAT- and K-GNN-based methods outperform the existing baseline in terms of accuracy and efficiency. On the one hand, the model in this paper considers the causal features of top-down propagation pattern of rumors along the relationship chain. On the other hand, it considers the structural features of bottom-up aggregation and diffusion of rumors within the community. Comparing with existing social network rumor detection methods, the method in this paper has better performance.

In future, we will add a module of sentiment analysis to the detection model, which will be used to improve the interpretability of rumor detection and to give a corresponding confidence level to the detection results. Finally, we aim to design a rumor detection system to detect real-time rumor comments.

Data Availability

The detailed parameter data of this article has been listed in the paper; according to this data, everyone can get the results of this paper.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Key Program of the Joint Fund of National Natural Science Foundation of China (No. U1836212) and State Administration of Science, Technology and Industry for National Defence, PRC (No. JCKY2020602B008).

References

- [1] Q. Wu, F. C. Chen, R. Y. Huang, and C. Z. Chao, "A semantic path-based approach to heterogeneous network community discovery," *Journal of Electronics*, vol. 44, no. 6, pp. 1465–1471, 2016.
- [2] L. Wu, J. Li, X. Hu, and H. Liu, "Gleaning wisdom from the past: early detection of emerging rumors in social media," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 99–107, Houston, Texas, USA, 2017.
- [3] J. Ma, W. Gao, P. Mitra, S. Kwon, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *International Joint Conference on Artificial Intelligence*, New York, USA, 2016.
- [4] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, and M. Grohe, "Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4602–4609, 2019.

- [5] T. Bian, X. Xiao, T. Xu et al., "Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 549–556, 2020.
- [6] Y. Rong, W. Huang, T. Xu, and J. Huang, *DropEdge: Towards Deep Graph Convolutional Networks on Node Classification*, *8th International Conference on Learning Representations*, ICLR, Addis Ababa, Ethiopia, 2020.
- [7] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis can we trust what we RT?," in *Proceedings of the First Workshop on Social Media Analytics*, pp. 71–79, Washington, DC, USA, 2010.
- [8] 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 - MDS '12*, New York, USA, 2012.
- [9] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *2013 IEEE 13th International Conference on Data Mining*, Dallas, TX, USA, December 2013.
- [10] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, New York, USA, October 2015.
- [11] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*, Florida, Italy, May 2015.
- [12] M. Jing, G. Wei, Z. Wei, Y. Lu, and K.-F. 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*, New York, USA, October 2015.
- [13] L. Yin, X. Luo, C. Zhu, L. Wang, Z. Xu, and H. Lu, "ConnSpooiler: disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1373–1384, 2020.
- [14] 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 (Volume 1: Long Papers)*, pp. 708–717, Vancouver, Canada, 2017.
- [15] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3901–3907, AAAI Press, 2017.
- [16] C. Tong, L. Xue, H. Yin, and J. Zhang, "Call attention to rumors: deep attention based recurrent neural networks for early rumor detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, 2018.
- [17] S. Su, Z. Tian, S. Li et al., "IoT root union: a decentralized name resolving system for IoT based on blockchain," *Information Processing & Management*, vol. 58, no. 3, article 102553, 2021.
- [18] W. Ke, Y. Song, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," in *2015 IEEE 31st International Conference on Data Engineering*, Seoul, Korea (South), April 2015.
- [19] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The World Wide Web Conference on - WWW '19*, pp. 3049–3055, San Francisco, USA, 2019.
- [20] 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 (Volume 1: Long Papers)*, pp. 1980–1989, Melbourne, Australia, July 2018.
- [21] S. Su, Z. Tian, S. Liang, S. Li, S. du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, 2020.
- [22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.
- [23] N. T. Kipf and M. Welling, "Semi supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, 2017.
- [24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, <https://arxiv.org/abs/1710.10903>.
- [25] J. Qiu, Y. Chai, Z. Tian, X. du, and M. Guizani, "Automatic concept extraction based on semantic graphs from big data in smart city," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 225–233, 2020.
- [26] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, Sydney, Australia, 2017.
- [27] Y. Rong, W. Huang, T. Xu, and H. Junzhou, "The truly deep graph convolutional networks for node classification," 2019, <https://arxiv.org/abs/1907.10903>.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [29] "thunlp/Chinese_Rumor_Dataset [EB/OL]," 2019, https://github.com/thunlp/Chinese_Rumor_Dataset.
- [30] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, <https://arxiv.org/abs/1412.6980>.
- [31] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, 2020.
- [32] D. Mudali, L. K. Teune, R. J. Renken, K. L. Leenders, and J. B. T. M. Roerdink, "Classification of Parkinsonian syndromes from FDG-PET brain data using decision trees with SSM/PCA features," *Computational and Mathematical Methods in Medicine*, vol. 2015, Article ID 136921, 2015.
- [33] H. Bisgin, O. U. Kilinc, A. Ugur, X. Xu, and V. Tuzcu, "Diagnosis of long QT syndrome via support vector machines classification," *Journal of Biomedical Science and Engineering*, vol. 4, no. 4, pp. 264–271, 2011.
- [34] H. Ming, Y. Lu, Z. Zhang, and M. Dong, "A light-weight method of building an LSTM-RNN-based bilingual tts system," in *2017 International Conference on Asian Language Processing (IALP)*, Singapore, December 2017.
- [35] A. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Systems with Applications*, vol. 158, article 113595, 2020.

- [36] L. Yang and Y. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018.
- [37] F. Xing and C. Guo, "Mining semantic information in rumor detection via a deep visual perception based recurrent neural networks," in *2019 IEEE International Congress on Big Data (BigDataCongress)*, Milan, Italy, July 2019.
- [38] S. Y. Louis, Y. Zhao, A. Nasiri et al., "Graph convolutional neural networks with global attention for improved materials property prediction," *Physical Chemistry Chemical Physics*, vol. 22, no. 32, pp. 18141–18148, 2020.
- [39] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral Networks and Locally Connected Networks on Graphs, Computer Science," 2013, <http://arxiv.org/abs/1312.6203>.