Deep Structure Learning for Rumor Detection on Twitter

Conference Paper · July 2019

DOI: 10.1109/IJCNN.2019.8852468

5 authors, including:

Qi Huang
Chinese Academy of Sciences
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Chuan Zhou
Chinese Academy of Sciences
75 PUBLICATIONS **669** CITATIONS
SEE PROFILE

Jia Wu
University of Technology Sydney
94 PUBLICATIONS **1,234** CITATIONS
SEE PROFILE

Wang Mingwen
Jiangxi Normal University
79 PUBLICATIONS **180** CITATIONS
SEE PROFILE

IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

Deep Structure Learning for Rumor Detection on Twitter

Qi Huang*†, Chuan Zhou*†, Jia Wu‡, Mingwen Wang§, and Bin Wang¶
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
‡Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia
§School of Computer and Information Engineering, Jiangxi Normal University, Nanchang, China
¶Xiaomi AI Lab, Beijing, China
{huangqi, zhouchuan}@iie.ac.cn, jia.wu@mq.edu.au, mwwang@jxnu.edu.cn, wangbin11@xiaomi.com

***Abstract*—With the development of social media and the popularity of mobile devices, it becomes increasingly easy to post rumors and spread rumors on social media. Widespread rumors may cause public panic and negative impact on individuals, which makes the automatic detection of rumors become necessary. Most existing methods for automatic rumor detection focus on modeling features related to contents, users and propagation patterns based on feature engineering, but few work consider the existence of graph structural information in the user behavior. In this paper, we propose a model that leverages graph convolutional networks to capture user behavior effectively for rumor detection. Our model is composed of three modules: 1) a user encoder that models users attributes and behaviors based on graph convolutional networks to obtain user representation; 2) a propagation tree encoder, which encodes the structure of the rumor propagation tree as a vector with bridging the content semantics and propagation clues; 3) an integrator that integrates the output of the above modules to identify rumors. Experimental results on two public Twitter datasets show that our model achieves much better performance than the state-of-the-art methods.**

***Index Terms*—rumor detection, user embedding, graph convolutional networks**

I. Introduction

A rumor is commonly defined as a statement whose truth value is unverifiable or deliberately false [1]. These rumors spread on social media, carrying unreal or even malicious information, which will bring massive damage to individuals and society. For example, during the 2016 U.S. presidential election, there are 529 rumors about Donald Trump and Hillary Clinton on twitter [2], which has greatly damaged the candidate's reputation and influenced the judgment of the voters. Therefore, automatic rumor detection that is able to identify rumors accurately and immediately becomes significant.

Most of the existing rumor detection methods were based on statistical machine learning. These methods regarded rumor detection as a supervised classification problem, and used feature engineering methods to mining effective features from sequential microblogs streams [3]–[6], [10], where features based on content, user and propagation were integrated for rumor detection. Recently, some models for rumor detection exploited neural networks were inspired by the success of neural networks model in other tasks. Ma. et al. [7] proposed a model based on recurrent neural network to capture the variation of contextual information of relational tweet for rumor detection. In order to focus on the key factors of the rumor detection task, Chen et al. [8] and Guo et al. [9] introduced attention mechanism in the neural network model for rumor detection. Another branch of rumor detection work focused on the propagation structure patterns of rumors. Wu et al. [11] and Ma et al. [12] proposed a kernel-based model that identifies rumors and non-rumors by computing similarities of their propagation tree structure. To learn a high-level feature representation of rumor propagation structure, Ma et al. [13] presented a neural rumor detection approach based on recursive neural networks (RvNN) to represent the content semantics and propagation clues. On the similar task fake news detection, Ruchansky et al. [14] leveraged singular value decomposition method to deal with user behavior information.

Research [29] shown that rumors will deploy a group of loyal promoters to promote and disseminate false rumors in order to reach a large audience. Analysis on small-scale dataset found that there are often groups of users that heavily publicize false rumors, especially after the release of rumors. However, previous methods consider the user information insufficiently, and ignore the structure features of user relationship network, such as the relationship network formed by user behavior. Relationship network is generally represented by a graph. Due to the large number of users in social media, the edges of graph formed by user behavior are sparse. The method of modeling the user needs to overcome the sparsity of the edge to extract valid information. For graph structure data, graph convolutional networks enable to directly operate on the graph, which effectively learns the representation of each node through its neighborhoods.

In this paper, we propose a model that learning user representation by graph convolutional networks for rumor detection in social media. Our model consists of three modules: user encoder, propagation tree encoder and integrator. User encoder handles a graph formed by user behavioral information with graph convolutional networks to learn a representation of user. Meanwhile, propagation tree encoder learns the representation of propagation tree with recursive neural network, which

978-1-7281-2009-6/$31.00 paper N-20148.pdf

bridges the content semantics and propagation clues. Integrator incorporates the output of above module by a fully connected layer for rumor detection. The main contributions of our work are as follows:

- To the best of our knowledge, this is the first work that explores the deep structure of user behavior for rumor detection. The graph formed by user behavior information enable to effectively capture with graph convolutional networks.
- Our model covers influential factors of rumor detection: content, user, and propagation, and effectively captures information about those factors. User encoder models the user and propagation tree encoder learns the representation bridging the content semantics and propagation clues.
- Experiments on two real-word datasets show that our method is more superior to the state-of-the-art on rumor detection.

II. RELATED WORK

A. Rumor Detection

Most previous approaches for rumor detection were based on statistical machine learning. These work designed a series of effective features in order to identify rumors. The designed features included three aspects: content-based, user-based and propagation-based. Content-based features consisted of length, symbols, sentiment, URLs, hashtags [3], vocabulary and its part of speech et al [15]. User-based features were statistics on registration age, friendship, activity level and history actions of users [3]. Propagation-based features included the depth of the propagation tree, the number of comment, the number of retweet et al [3]. Subsequently, considered changes in features over time series, Kwon et al. [4] proposed a time-series-fitting model based on the number of tweets over time. Based on their research, Ma et al. [6] explored a model that considers more social contextual features changes in time series. These approaches are labor-intensive and oversimplify the features.

To alleviate the efforts of feature engineering, Zhao et al. [16] provided a set of regular expressions (such as "really?", "not true", etc.) for finding enquiry tweets which include verification questions and corrections. But the method has a low recall rate due to the limitations of the regular expressions. Ma et al. [7] used a recurrent neural networks (RNN) to learn the representation of contextual information changes in relevant tweets over time. Recently, they proposed a neural multi-task learning framework to improve the effectiveness of rumor detection and stance detection tasks [17], considering the mutual promotion between rumor detection and stance detection.

Another branch of rumor detection work focused on the specific propagation structure patterns of rumors. Wu et al. [11] first modeled the propagation structure using a kernel-based approach. They explored an SVM classifier with a hybrid kernel function which combines RBF kernel and random-walk-based graph kernel to detect rumors in Sina Weibo. Ma et al. [12] proposed a tree-based model to calculate the similarity of two propagation trees to identify different types of rumors in Twitter. And then Ma et al. [13] studied a recursive neural network bridging content semantics and propagation clues to learning a high-level and natural representation for detection.

However, regardless of modeling changes of statistic features based on content, user, and propagation over times series, or modeling propagation tree based on kernel, these methods do not fully take into account the user information. In addition to the statistical features of user such as number of fans, number of friends, age of registration, and whether the account is verified, there also has relationship networks between users. According to the characteristics of the social network platform, we build a user behavior relationship network based on the user participation in each claim on Twitter. And user behavioral information determines the authority and credibility of user sometimes. Therefore, we consider the integration of user behavior relationship network information into the rumor detection method.

B. Graph Neural Networks

Recently, the topic of graph neural networks has aroused widespread concern [30], [31], [36]–[41]. Some researchers had extended mature neural network models [32]–[34], [42], such as CNN, to apply to regular grid structures (two-dimensional grids or one-dimensional series) that can be used for graphics of arbitrary structures. Based on their ground-breaking work, Kipf and Welling [18] proposed a simple graph neural network model, called the Graph Neural Networks (GCN), which achieved better results than the benchmark method in the graph datasets. GCN has also been applied to various of applications, e.g., text classification [18], [19], relation extraction [20], image classification [21], [22], [35], molecular fingerprints [23], [24] and protein interface prediction [25]. For the unstructured data in the task, the researchers used the graph structure data of the external resource or assumed the relational structure in the task, and then used the graph convolution networks to directly operate on the graph. In this paper, we consider the integration of user behavior relationship network information into the rumor detection method, and a user behavior relationship network can be represented by a graph. Based on the excellent performance of graph convolutional networks in the representation of graph structure data, we propose to use graph convolution networks to model user for rumor detection.

III. PROBLEM STATEMENT

We define a rumor detection dataset as a set engagements $Eg = \{Eg_1, Eg_2, \cdots, Eg_{|Eg|}\}$, where each engagement contains a claims set and a corresponding set of users, i.e., $Eg_i = \{C_i, U_i\}$. C_i represents all tweets that are ideally chronologically ordered by the reply source tweet r_i, i.e., $C_i = \{r_i, x_{i1}, x_{i2}, \cdots, x_{im}\}$ where each x_{i*} is a responsive tweet of the root r_i or the retweet of r_i. U_i is the poster corresponding to each tweet in C_i, i.e., $U_i = \{u_{r_i}, u_{x_{i1}}, u_{x_{i2}}, \cdots, u_{x_{im}}\}$. According to the reply or repost relationship between the

Fig. 1: An example of propagation tree structure

tweets, we can form C_i as a propagation tree structure [11]–[13] with r_i being the root node (an instance sees Fig. 1). The co-occurrence relationship of users in different C_i can form a user graph *G = (V, E)*, where *V* represents the node of the graph and *E* represents the edge of the graph. All users in the engagements *Eg* form the node *V*, and their co-occurrence relationship derived from behavior information forms the edge *E* of the graph.

The goal of rumor detection task is to construct a classifier that can determine whether the claim in engagement is a rumor. The classifier can be formalized as a function $f : C_i \rightarrow Y_i$, where Y_i is one of four categories: non-rumor, false rumor, true rumor, and unverified rumor that has been introduced in the previous work [12], [13], [26]. To get an effective classifier, we used different neural networks to capture the characteristics of the propagation tree and user graph.

IV. PROPOSED MODEL

In this section, we give the details of the proposed model. Our model consists of three parts, namely user encoder, propagation tree structure encoder with content semantics and integrator (see Fig. 2). User encoder obtains the user representation using graph convolutional networks to model the user graph. Propagation tree structure encoder encodes the propagation tree by a tree-based recursive neural network. And then integrator combines these features to a fully connected layer for rumor detection. In the sections that follow, we introduce each module in detail.

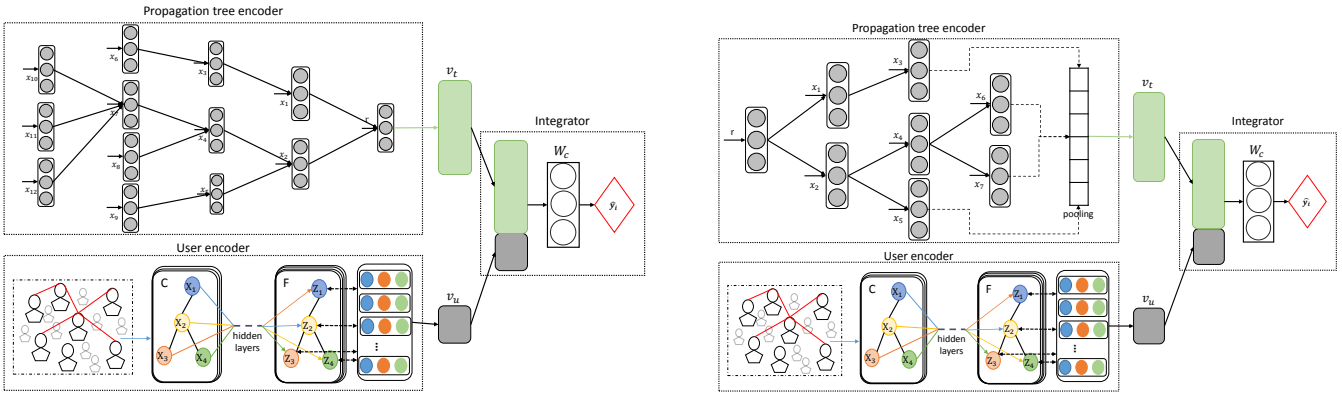
A. User Encoder

In the user encoder module, we try to consider more comprehensive and effective user information including user features and their behavioral information. In other words, In addition to the statistical-based user features used by most researchers, such as the number of followers, the number of fans and the age of registration, we want to capture the behavior information. Further, we incorporate statistical-based user features and user behavior information to get a high-level user representation in a more efficient and automated way.

As the core of the user encoder module, we use a graph convolutional networks, for the reason that graph convolutional networks have been proven to capture node features and graph structure features in graph structure data efficiently. A GCN is a multi-layered neural network that operates directly on the graph, which updates the representation of nodes based on the properties of their neighborhoods. In the work of Kipf et al. [18], graph convolutional networks have demonstrated its effectiveness in the node classification task: classifier with a GCN can learn the neighborhoods feature of nodes to provide information for node classification problems. Whether the GCN captures the information of the immediate neighbors (with one layers of convolution) or the neighbors information of the k-level hops (if K layers are stacked on top of each other) depends on how many layers of the convolution are used.

More formally, GCNs process an undirected graph *G = (V, E)*, where $V(|V| = n)$ and *E* are sets of nodes and edges, respectively. Each node in the graph assumes self-loop, i.e., $(v, v) \in E$. The features of all nodes in the graph can be defined as matrix $X \rightarrow \mathbb{R}^{n \times m}$, where m is the dimension of the feature vectors. Each row of matrix $x_v \rightarrow \mathbb{R}^m (v \in V)$ represents the feature vector of a node v in the graph. To formalize the edges of the graph, we introduce the adjacency matrix A and its degree matrix $\widetilde{D}$, where $\widetilde{D}_{ii} = \sum_j A_{ij}$. The diagonal elements of A are set to 1 because of self-connection. The GCN can choose to convolve only one layer of immediate neighbors to capture information, and also can convolve multiple layers to combine more neighborhoods information. For a one-layer GCN, the calculation formula of the new k-dimensional node feature matrix $L^{(1)} \rightarrow \mathbb{R}^{n \times k}$ is as follow:

$$L^{(1)} = \delta(\widetilde{A}XW_0). \tag{1}$$



(a) hybrid model with bottom-up RvNN

(b) hybrid model with top-down RvNN

Fig. 2: An illustration of the proposed hybrid neural model

Here, $\widetilde{A} = \widetilde{D}^{-\frac{1}{2}} A \widetilde{D}^{-\frac{1}{2}}$, and $W_0 \in \mathbb{R}^{m \times k}$ is a trainable weight matrix. $\delta(\cdot)$ represents an activation function, e.g. a ReLU function $\delta(x) = max(0, x)$. When GCN convolves multiple layers to combine more neighborhoods information, the update of node matrix follows:

$$L^{(j+1)} = \delta(\widetilde{A} L^{(j)} W_j). \tag{2}$$

Where *j* represents the number of convolution layer and $L^{(0)} = X$.

As mentioned above, the user behavior information, participating in discussing different claims, can form a co-occurrence relationship graph $G = (V, E)$. The nodes in the co-occurrence graph are the user who posts, reposts or replies the claim. The feature matrix X of these users is composed of user profile information, such as the number of fans, number of friends, whether it is officially verified, etc. The values of the adjacency matrix A are defined as follows:

$$A_{i,j} = \begin{cases} 1, & u_i, u_j \, discuss \, the \, same \, claim \\ 1, & i = j \\ 0, & otherwise \end{cases} \tag{3}$$

B. Propagation Tree Structure Encoder

In the propagation tree structure encoder module, we aim to capture the structural and semantic features of propagation tree. We use a recursive neural network based on tree structure proposed by Ma et al. [13] to capture propagation clues and semantic features. Since the traversal of tree structure data is directional, we adopt two different structures of recursive neural network: bottom-up RvNN encoder and top-down RvNN encoder (see Fig. 3).

The bottom-up RvNN model uses a bottom-up approach to traversing the entire tree to obtain the root node representation, where the representation of parent node is calculated by itself features and representations of its child nodes. Then, we use the representation of root node as the embedding v_t of propagation tree. To model long-distance interactions in tree nodes, the model uses an extended gate recurrent unit (GRU) [27]. The calculation formula of the parent node is as follows:

$$\begin{aligned} h_C &= \sum_{c \in C(j)} h_c \\ r_j &= \delta(W_r x_j + U_r h_C + b_z) \\ z_j &= \delta(W_z x_j + U_z h_C + b_z) \\ \widetilde{h}_j &= tanh(W_h x_j + U_h(h_C \odot r_j)) \\ h_j &= (1 - z_j) \odot h_C + z_j \odot \widetilde{h}_j \end{aligned} \tag{4}$$

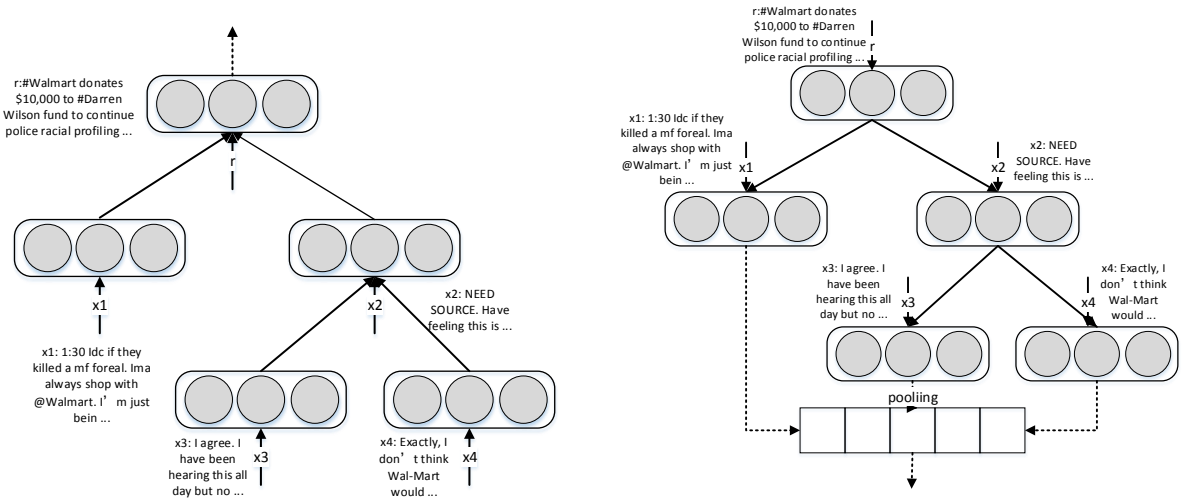
Where $C(j)$ is the set of all child nodes of parent *j*, and x_j is the vector representation of the semantic text of parent *j*. $\delta(\cdot)$ representes a sigmod function, and $\odot$ denotes element-wise multiplication. $[W_*, U_*]$ are the weight matrix inside GRU, and b_* is the bias vector. h_j and h_c refer to the hidden state of node *j* and its *c-th* child.

The top-down RvNN model traverses the entire tree in a top-down manner. So when calculating the hidden state of a node, the model considers the features of the current node and the representation of the node's parent node. Therefore, the hidden state $h_j$calculation formula does not have the formula for calculating the sum of all child nodes in Equation 4, and replaces the symbol h_C with the representation h_P of the parent node (see Equation 5). Finally, all leaf nodes of the tree pass through a pooling layer as a embedding v_t of the propagation tree.

$$\begin{aligned} r_j &= \delta(W_r x_j + U_r h_{P(j)} + b_z) \\ z_j &= \delta(W_z x_j + U_z h_{P(j)} + b_z) \\ \widetilde{h}_j &= tanh(W_h x_j + U_h(h_{P(j)} \odot r_j)) \\ h_j &= (1 - z_j) \odot h_{P(j)} + z_j \odot \widetilde{h}_j \end{aligned} \tag{5}$$

C. Integrator

The user encoder module obtains a user matrix $U \in \mathbb{R}^{n \times m}$ that fuses user statistics features and behavior information. And the propagation tree encoder module obtains a tree representation $v \in \mathbb{R}^d$ that combines the propagation tree



(a) Bottom-up RvNN encoder

(b) Top-down RvNN encoder

Fig. 3: An illustration of propagation tree structure encoder

structure and text semantic features. In order to fuse the output information of the two modules together, we propose an integration module in which user embedding v_{u_i} combined with the propagation tree representations v_{t_i} to produce a category prediction $\widetilde{y}_i$ for each claim *i*.

In order to integrate the output of two modules, we find a representation v_u corresponding to a user in the user matrix U, which is the poster of root node tweet in the propagation tree. Then we concatenate the user representation v_u with the representation v_t. The cascaded result v_c is fed into a fully connected layer to predict the category of each claim:

$$\widetilde{y}_i = \delta(w_c v_c + b_c) \tag{6}$$

The integration module integrates two modules together to get a more accurate prediction. By combining the user encoder module and the propagation tree encoder module, the model simultaneously learns the propagation structure information and user information. At the same time, the model contains almost all the information involved in rumor detection: contents, users, and propagation.

D. Model Training

The goal of the training is to minimize the cross entropy of the probability distribution of the prediction and the ground truth:

$$L(Y,P) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} y_{i,k} log(p_{i,k}) + \lambda\|\theta\|_2^2, \tag{7}$$

where $y_{i,k}$ is the ground truth of the *i-th* sample in the *k-th* class (1 if $y_{i,k}$ belongs to the *k-th* class, otherwise 0), and $p_{i,k}$ is the probability of prediction that the *i-th* sample belongs to the *k-th* class. *M* is the number of training data, *K* is the number of classes, $\|\cdot\|_2$ is the L_2 regularization term for all parameters θ in the model, and λ is the trade-off coefficient.

During training, we use an efficient backpropagation algorithm to update the parameters of the model. The optimizer uses a gradient-based approach following the Ada-grad rules [28] to accelerate the convergence of the model. We empirically set the number of layers of the graph convolution networks to 2, the number of convolution neurons in the hidden layer is 64, and the output dimension of user is 4. The parameter settings of the recurrent neural network are consistent with the settings of ma et al. [13] and the dimension of propagation tree representation is 100. We iterate through all the training data every epoch until the loss value converges or reaches the maximum number of iterations.

E. Rumor Detection

For the sake of structural integrity, in this section we review the entire rumor detection process. Given a rumor propagation tree structure and the poster of root node, the user encoder gets the user vector representation, the propagation tree encoder gets the propagation tree vector, and the integrator cascades above two vectors to identify rumors.

V. EXPERIMENTS

A. Dataset

For experiment evaluation, we use two publicly available twitter datasets collected in [12], namely Twitter15 and Twitter16. Twitter15 dataset contains 1,381 propagation trees and 276,663 users. Twitter16 contains 1,181 propagation trees and 173,487 users. Each propagation tree in the dataset is labeled as one of four categories, i.e., non-rumor, false rumors, true rumors, and uncertain rumors. The more detailed statistics are shown in Table I.

TABLE I: Statistics of the datasets

Statistic	Twitter15	Twitter16
# of users	276,663	173,487
# of source tweets	1,490	818
# of threads	331,612	204,820
# of non-rumors	374	205
# of false-rumors	370	205
# of true-rumors	372	203
# of unverified rumors	374	203
Avg. time length / tree	1,337 Hours	848 Hours
Avg. # of posts / tree	223	251
Max # of posts / tree	1,768	2,765
Min # of posts / tree	55	81

B. Baseline Methods

On the rumor detection task, we compare our model with some state-of-the-art baselines, including some feature based methods (i.e. DTR, DTC, RFC, and SVM-TS), two kernel based methods (i.e. SVM-TK and SVM-HK) and three neural network based methods (i.e. GRU-RNN, BU-RvNN and TD-RvNN).

1) **DTR**: Decision-Tree-based method proposed by Zhao et al. [16], which ranks the clusters obtained by searching regular expression queries to identify trending rumors.
2) **DTC**: Castillo et al. [3] proposed an information credibility model using a decision-tree classifier. The model considers a variety of handcrafted statistical features to discriminate the credibility of trending topics.
3) **RFC**: Kwon et al. [4] explored a random forest classifier that utilizes temporal properties and a set of manually extracted features from user, language, and structural attributes.
4) **SVM-TS**: A linear SVM classifier model [6], which takes into account the time-series structure to model the variation of social context features from contents, users, and diffusion patterns.
5) **SVM-HK**: An SVM classifier which uses a Hybrid Kernel [11] consisting of a random-walk-based graph kernel and an RBF kernel.
6) **SVM-TK**: An SVM classifier uses a Tree Kernel [12], which identifies different types of rumors by computing the similarity of the propagation tree structure.
7) **GRU-RNN**: A detection model based on recurrent neural networks [7] with GRU units to represent rumors by modeling sequential structure of relevant posts.
8) **BU-RvNN And TD-RvNN**: A recursive neural networks with bottom-up tree structure and top-down tree structure [13], respectively.
9) **BU-Hybrid and TD-Hybrid**: our model adding a graph convolution networks to model user based on BU-RvNN and TD-RvNN, respectively.

We used pytorch to implement the recursive neural network and graph convolutional neural network in the model. We performed 5-fold cross-validation on the datasets and used accuracy for all four categories and F1 measurements for each category to assess the performance of the model.

TABLE II: Results of rumor detection.(NR: non-rumor; FR: false rumor; TR: true rumor; UR: unverified rumor)

(a) Twitter15 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
DTR	0.409	0.501	0.311	0.364	0.473
DTC	0.454	0.733	0.355	0.317	0.415
RFC	0.565	**0.810**	0.422	0.401	0.543
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-HK	0.493	0.650	0.439	0.342	0.336
SVM-TK	0.667	0.619	0.669	0.772	0.645
GRU-RNN	0.641	0.684	0.634	0.688	0.571
BU-RvNN	0.708	0.695	0.728	0.759	0.653
TD-RvNN	0.723	0.682	0.758	0.821	0.654
BU-Hybrid	0.738	0.796	0.713	0.773	0.663
TD-Hybrid	**0.752**	0.699	**0.773**	**0.831**	**0.709**

(b) Twitter16 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
DTR	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	**0.755**	0.420	0.571	0.526
SVM-HK	0.511	0.648	0.434	0.473	0.451
SVM-TK	0.662	0.643	0.623	0.783	0.655
GRU-RNN	0.633	0.617	0.715	0.577	0.527
BU-RvNN	0.718	0.723	0.712	0.779	0.659
TD-RvNN	0.737	0.662	0.743	0.835	0.708
BU-Hybrid	0.735	0.706	0.735	0.860	0.636
TD-Hybrid	**0.773**	0.716	**0.756**	**0.870**	**0.756**

C. Rumor Classification Performance

As shown in Table II, our proposed model basically achieved better performance than the other methods on the two datasets via capturing the user embedding with graph convolutional networks.

It is observed that the performance of four baseline methods in the first group based on manual features are very poor, varying between 0.409 and 0.585 in accuracy. DTR ranks the candidate rumor cluster which collected related tweets matching a set of regular expressions to identify rumors. While only a few posts match these regular expressions, it preforms the worst. Compared to the DTC, the SVM-TS and RFC take into account the time-series structure to model variation of handcrafted features, so their effect is better than DTC.

When observing the performance of the method considering the propagation of structural features, except for SVM-HK, other methods are superior to methods based on handcrafted

features. And TD-RvNN achieves the best performance among all the baselines. This is because posters will correct some inaccurate information during the propagation of information in social media and TD-RvNN captures this self-correcting information effectively.

However, all baseline do not adequately extract user information, especially the behavior information of user. Our model uses a graph convolutional network for learning user representations to model user features and behaviors. The experimental results show that we have learned a valid user representation for rumor detection, and graph convolution networks captures the node attributes and structural attributes in the graph well.

For only the non-rumor class, our method is worse than the RFC and SVM-TS on the datasets. Compared to RFC and SVM-Ts, we not only consider the statistical features of user, but also consider the user behavioral network information. Since the users group behavior is more likely to spread rumors, so when we consider the user behavior information, it can improve the accuracy of rumor detection. Meanwhile, it bring some interference to the detection of non-rumor.

D. User Encoder Performance

To demonstrate the validity of the user encoder in the model, we designed two benchmark methods that replace the user encoder module: 1) a method that directly uses the user's statistical features ("BU-Features" and "TD-Features"). 2) An approach exploit the output of a fully connected layer that encodes user statistical features and the low dimensional representation of adjacency matrix formed by user behaviors ("BU-SVD" and "TD-SVD").

As shown in Table III, user encoder effectively improve the effect of rumor detection. A closer look reveals that the performance on twitter15 is better; we suspect that this due to the user scale on Twitter15 is larger and the user encoder can learn a perfect user representation.

It observes that the performance of our model with TD-RvNN is inferior to the benchmark method from Table IV. We analyze that the reason is the propagation tree encoder with top-down tree structure effectively integrates the information of the whole tree, and we only use the representation of the user who first posts the rumors. Therefore, relying on the cascading method to integrate these two vectors means that the effect is not good.

TABLE III: the performance of methods with bottom-up tree structure

(a) Twitter15 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
BU-Features	0.660	0.650	0.666	0.716	0.600
BU-SVD	0.675	0.663	0.673	0.759	0.607
BU-Hybrid	**0.738**	**0.796**	**0.713**	**0.773**	**0.663**

(b) Twitter16 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
BU-features	0.663	0.565	0.705	0.766	0.587
BU-SVD	0.686	0.610	0.729	0.792	0.582
BU-Hybrid	**0.735**	**0.706**	**0.735**	**0.860**	**0.636**

TABLE IV: the performance of methods with top-down tree structure

(a) Twitter15 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
TD-Features	**0.789**	0.732	**0.817**	**0.872**	**0.740**
TD-SVD	0.785	**0.743**	0.805	0.862	0.734
TD-Hybrid	0.752	0.699	0.773	0.831	0.709

(b) Twitter16 dataset

Method	Acc.	NR	FR	TR	UR
		F_1	F_1	F_1	F_1
TD-Features	**0.804**	**0.738**	0.781	0.924	**0.782**
TD-SVD	0.793	0.703	**0.786**	**0.927**	0.759
TD-Hybrid	0.773	0.716	0.756	0.87	0.756

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present a hybrid neural network model for rumor detection on twitter. This is the first work that model user with graph convolutional networks for rumor detection. The model consists of three modules: a user encoder module modeling the graph formed by user behaviors based on graph convolutional networks, a propagation tree structure encoder represents the propagation tree using a recursive neural network, and an integrator to integrate feature representations. Our model considers three aspects of rumor detection: contents, users, and propagation. Experiments on real-world datasets show that our method is more efficient than previous methods.

In future work, we plan to give weights to the edges in the user graph based on the stance of interaction between users. Moreover, we will further consider converting the entire rumor data set into a graph structure data.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No.2016YFB0801301), the NSFC (No. 61872360), the MQEPS (No. 9201701455 and No. 96158905), the MQRSG (No. 95109718), and the Youth Innovation Promotion Association CAS (No. 2017210). We would like to thank anonymous reviewers for the insightful comments. C. Zhou is the corresponding author.

REFERENCES

[1] D. Nicholas, and P. Bordia, "Rumor psychology: Social and organizational approaches," American Psychological Association. Washington, vol. 1, pp. 1-34, 2007.

[2] Z. Jin, J. Cao, H. Guo, Y. Zhang, Y. Wang, and J. Luo, "Detection and Analysis of 2016 US Presidential Election Related Rumors on Twitter," in Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, 2017, pp. 14-24.

[3] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 675-684.

[4] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in Proceedings of the 13th International Conference on Data Mining, 2013, pp. 1103-1108.

[5] 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, 2015, pp. 1867-1870.

[6] J. Ma, W. Gao, 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, 2015, pp. 1751-1754.

[7] J. Ma, W. Gao, P. Mitra, S. Kwon, B. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 3818-3824.

[8] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang, "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection," arXiv preprint arXiv 1704.05973, 2017.

[9] H. Guo, J. Cao, Y. Zhang, J. Guo, and J. Li, "Rumor Detection with Hierarchical Social Attention Network," in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 943-951.

[10] 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, 2012, p. 13.

[11] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in Proceedings of the 31st IEEE International Conference on Data Engineering, 2015, pp. 651-662.

[12] 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), 2017, vol. 1, pp. 708-717.

[13] 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 (Long Papers), 2018, pp. 1980-1989.

[14] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 797-806.

[15] V. Qazvinian, E. Rosengren, D.-R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, p. 1589-1599.

[16] 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, 2015, pp. 1395-1405.

[17] J. Ma, W. Gao, and K.-F. Wong, "Detect Rumor and Stance Jointly by Neural Multi-task Learning," in Companion of the The Web Conference 2018 on The Web Conference 2018, 2018, pp. 585-593.

[18] T.-N. Kipf, and M. Welling, " Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2017.

[19] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," arXiv preprint arXiv:1809.05679, 2018.

[20] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," arXiv preprint arXiv:1809.10185, 2018.

[23] D. K. Duvenaud, D. Maclaurin, J. Aguileraiparraguirre, R. Gomezbombarelli, T. D. Hirzel, A. Aspuruguzik, and R. P. Adams, ".Convolutional networks on graphs for learning molecular fingerprints," in Advances in neural information processing systems, 2015, pp. 2224-2232.

[21] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," arXiv preprint arXiv:1711.04043, 2017.

[22] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6857-6866.

[24] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," Journal of computer-aided molecular design. vol. 30, pp. 595-608, 2016.

[25] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in Advances in Neural Information Processing Systems, 2017, pp. 6530-6539.

[26] A. Zubiaga, M. Liakata, R. Procter, G.-W.-S. Hoi, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," PloS one. vol. 11, e0150989, 2016.

[27] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

[28] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research. vol. 12, pp. 2121-2159, 2011.

[29] "Social Network Analysis Reveals Full Scale of Kremlins Twitter Bot Campaign," (April 2015). globalvoices.org/2015/04/02/analyzing-kremlin-twitter-bots/. 2015.

[30] H. Cai, V. W. Zheng, and K. Chang, " A comprehensive survey of graph embedding: problems, techniques and applications," IEEE Transactions on Knowledge and Data Engineering, vol. 30, pp. 1616-1637, 2018.

[31] P. W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti,D. Raposo, A. Santoro, R. Faulkner, et al. "Relational inductive biases, deep learning, and graph networks," arXiv preprint arXiv:1806.01261, 2018.

[32] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[33] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graphstructured data" arXiv preprint arXiv:1506.05163, 2015.

[34] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral ltering," in Advances in Neural Information Processing Systems, 2016, pp. 3844-3852.

[35] S. Fang, H. Xie, Z.J. Zha, N. Sun, J. Tan, and Y. Zhang, "Attention and Language Ensemble for Scene Text Recognition with Convolutional Sequence Modeling," in Proceedings of 2018 ACM Multimedia Conference on Multimedia Conference, 2018, pp. 248–256.

[36] C.Y. Liu, C. Zhou, J. Wu, Y. Hu, and L. Guo, "Social Recommendation with an Essential Preference Space," in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 346-353.

[37] L. Gao, H. Yang, C. Zhou, J. Wu, S. Pan, and Y. hu, "Active Discriminative Network Representation Learning," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2142-2148.

[38] H. Wang, C. Zhou, J. Wu, W. Dang, X. Zhu, and J. Wang, "Deep Structure Learning for Fraud Detection," in Proceedings of the 18th IEEE International Conference on Data Mining, 2018, pp. 567-576.

[39] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-Party Deep Network Representation," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 1895-1901.

[40] J. Wu, S. Pan, X. Zhu, C. Zhang, and S.Y. Philip, "Multiple Structure-View Learning for Graph Classification," EEE transactions on neural networks and learning systems. vol. 29, pp. 3236-3251, 2018.

[41] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially Regularized Graph Autoencoder for Graph Embedding," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2609-2615.

[42] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "DiSAN: Directional self-attention network for rnn/cnn-free language understanding," in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 5446-5455.