A Graph Convolutional Encoder and Decoder Model for Rumor Detection

1st Hongbin Lin
Software Engineering
Shenzhen University
Shenzhen, China
1910273004@email.szu.edu.cn

2nd Xi Zhang
Faculty of Arts and Sciences
Shenzhen Technology University
Shenzhen, China
zhangxi@sztu.edu.cn

3rd Xianghua Fu
Faculty of Arts and Sciences
Shenzhen Technology University
Shenzhen, China
fuxianghua@sztu.edu.cn

***Abstract*—With the development of technology and the expansion of social media, rumors spread widely and the rumor detection has gradually caused widespread concern. The early method of using handcrafted features has been eliminated due to inefficiency, and deep learning methods have been gradually adopted in recent years. However, most of the methods only consider content information such as text, which is often not enough for the specific field, rumor detection. Some studies take propagation rule into consideration, such as Kernel-based, RvNN. In addition, the structure formed via propagation of rumors and non-rumors have different properties. Compared with dynamic propagation, structure here is the final result of propagation and it's static and global. In order to enhance the structure information, we proposes a model that obtains textual, propagation and structure information. The model contains three components: *Encoder*, *Decoder*, and *Detector*. The encoder uses the efficient Graph Convolutional Network to regard the initial text as input and update the representation through propagation to learn text and propagation information. Then the encoded representation would be used for subsequent decoder which uses AutoEncoder to learn the overall structure information. Simultaneously, the detector utilizes the output of encoder to classify events as fake or not. These three modules are jointly trained to improve the model effect. We verified our method on three real-world datasets, and the results show that our method outperforms other state-of-the-art methods.**

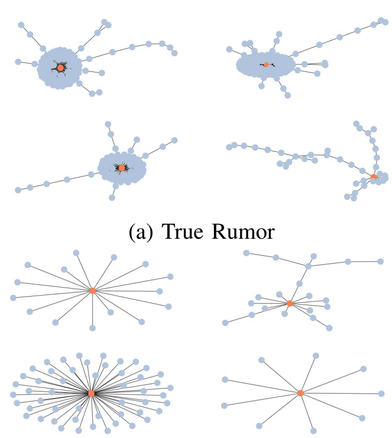
***Index Terms*—rumor detection; graph structure; autoencoder**

I. Introduction

In recent years, deep learning methods have been widely used in rumor detection, which are roughly divided into two categories. Many previous researches focused on text mining and learning semantic features of text. For example, Ma et al. [1] used recurrent neural networks (RNNs) including LSTM and GRU to learn representation of claims in rumor detection. Yu et al. [2] used convolutional kernels to extract features on text to train a convolutional neural network(CNNs).

But most of these methods ignore or oversimplify the message propagation. Analysis [3] has found that there is a "self-correct" effect among a certain circle like social media. Different tweets which response or retweet to the root claim, represent different opinions and correct the root claim. This process is dynamically formed and the latent representation of the tweets is influenced by each other. Therefore, Ma [4] used RvNN, a type of tree-structured neural network for syntactic and semantic parsing [5], to strengthen the high-level representation of tree nodes via propagation. Unlike standard RvNN, the input is a propagation tree rooted from a source post instead of parse tree and each node represents a post. They used GRU unit via recursion propagation to update nodes' representations. However, GRU unit is not a perfect way to learn representations and have some efficiency challenges since it's trained via sequential propagation. Compared with GRU, Graph Convolutional Network (GCN) can deal with the same issue which is based on nodes' relationships, and apart from this, GCN can update nodes' representations simultaneously with high efficiency and better accuracy. Since GCN successfully made progress on various fields in recent years, we can apply GCN to capture propagation information.

In addition, according to the research ([6]; [7]), structure of rumors and non-rumors formed differently through propagation. Different with dynamic propagation, structure is the final result of propagation and it represents static and global information. In order to explain uniformly, the item "structure" means the same meaning in the following article. In order to observe the structural difference between rumors and non-rumors more intuitively, we randomly visual structure of some data in the Twitter15 [8], a real world dataset.



(a) True Rumor

(b) False Rumor

Fig. 1: Structure visualization of sample data

978-1-7281-8206-3/20/$31.00 ©2020 IEEE
DOI 10.1109/DSAA49011.2020.00043

The red node in the Fig. 1 represents the root tweet, and other nodes have retweet or responsive relationships with each other. From the Fig. 1, we can roughly find some differences between these two categories and these differences were also explored in research [7] such as density. The density around the center is much greater and there is a long tail in data with label *True Rumor*, while information disperse regularly via nodes propagation in data with label *False Rumor*. It's consistent with the actual situation since fake news often contain opinionated language, crafted as "clickbait" (i.e., to entice users to click on the link to read the full article) or to raise up confusion and suspect, so that people will spread it widely and quickly. This phenomenon doesn't happen on true event so that it spread regularly. Zhou et al. [9] demonstrated that rumor's structure has some properties which is different with non-rumor's. Therefore, we want to enhance the structure information when training a model. Inspired by the success of AutoEncoder module in the field of learning latent information, we use GAE (Graph AutoEncoder) and its variant Variational GAE (VGAE) [10] to learn graph structure information on rumor detection. GAE and VGAE is the application of autoencoder and variational autoencoder on graph respectively. The difference between GAE and VGAE is whether to use gaussian distribution as the distrubution limitation of latent variable z, as described in more detail in Model Section. We will take these two modules alternatively as decoder in our experiment. Previous study [11] has shown the effective performance of autoencoder module in rumor detection and proposed a model, Multimodal Variational Autoencoder (MVAE). Compared with MVAE, we focus on the rumor structure instead of the multimodal (textual and visual) information.

To integrate all the useful clues mentioned above, in this paper, we proposed a model to capture the textual, propagation and structure information. Our model consists of three modules: the encoder, decoder and detector. The encoder handles a graph formed by tweets with GCN to learn representations, which bridges the content semantics and propagation rules. The decoder uses the output of encoder to reconstruct the structure so that obtain structure information. The detector incorporates representations by a fully connected layer for rumor detection. We train these modules jointly and evaluate it based on three real world datasets. The result shows that our method outperforms baselines on rumor detection with large margin.

The main contributions of this work are as follows:

- This is the first study that integrates text, propagation and structure information for detecting rumors from social media posts.
- To the best of our knowledge, this is the first study of employing GAE and Variational GAE in rumor detection.
- GCN is the core component of our method and can simultaneously update representations and encode the structure on graph data.
- Experimental results show that in the three real world datasets, our model performs better than other sota models. We make the source codes publicly accessible[1].

II. Related Work

Rumors are usually defined as a statement whose actual value has not been confirmed or that deliberately promotes concept contrary to fact. With the development of technology and Internet, tens of thousands of people post information on social media to share different views on various stuff. Among them, rumors spread widely and bring damage to society and people. Therefore, how to make the rumor detection accurately and efficiently becomes very important.

In recent years, with the gradual expansion of social networks, the spread and recognition of rumors have also attracted much attention. Supervised classification is widely used to identify rumors in social media posts. Early work focused on manually extracting features, using post content, author profiles, and propagation pattern. The survey [12] summarized the methodology involved in the rumor detection. Wu et al. [6] used kernel-based SVM classifier to detect rumor. These methods still depend on handcraft features so they are not effective and can't extract useful features.

With the increasing application of deep learning, there are many new methods to identify rumors. Wu et al. [13] uses the RNN network to classify and characterize the user profile of social media. Liu et al. [14] uses a combination of CNN and RNN to extract features for the task. Ma et al. [15] proposed a hierarchical attention network contains two attention layer, coherence-level and entailment-level. Similarly, Khoo et al. [16] employed self-attention and hierarchical token and post levels on two rumor detection datasets. Recently, the adversarial learning method is even used to improve the accuracy of rumor detection by training discriminators and generators based on Seq2Seq. Wechat Team [17] proposed a framework consisting of annotator, reinforced selector and fake news detector to obtain fresh and high-quality labeled samples.

Some methods were exploited to model the propagation information. Wu et al. [6] first using kernel-based approach to model propagation. They proposed a graph-kernel based hybrid SVM classifier to learn high-level propagation representation and also update semantic features. Ma et al. [8] then proposed a model called Propagation Tree Kernel to differentiate claims by comparing tree-based similarities between their propagation. And then [4] proposed a tree structure network based on RvNN to learn the feature representation of rumors and non-rumors.

Recently, the topic of graph neural network has aroused much concern and lots of researchers propose some approaches. Among them, GCN [18], proposed by Kipf and Welling, is the most effective to extract features based on convolutional network. GCN has good applications in many fields, e.g., protein interface prediction, text classification and community detection ([19]; [20]). When the issue is based on relationships of nodes, the researchers can use GCN to operate on the graph and obtain nodes' high-level features. In

[1] https://github.com/lhbrichard/rumor-detection

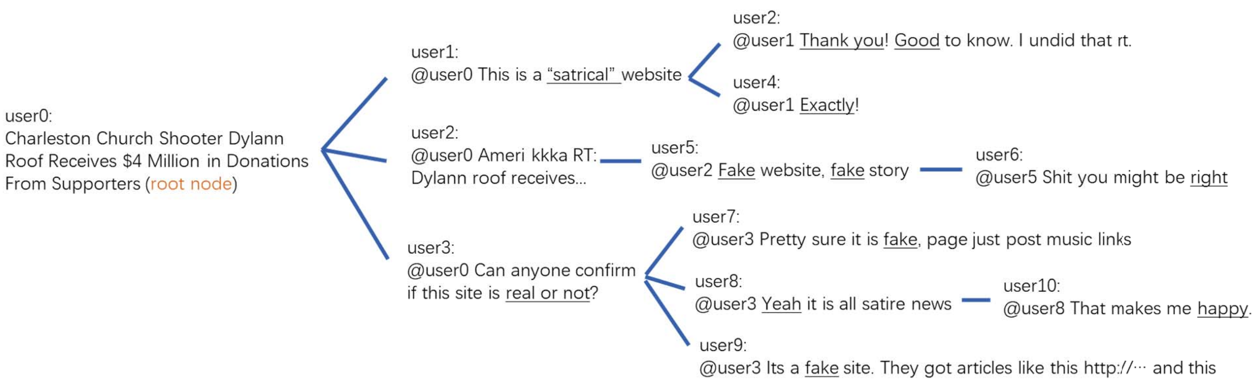


Fig. 2: An example of data

this paper, we use GCN as a core module to bridge semantics and propagation and encode graph structure.

The Autoencoder uses an unsupervised learning method to learn the latent representation which is be reconstructed to be similar to input by the decoder. Aiming at the special graph data, Kipf [10] proposed GAE to encode graphs to obtain latent representation of graph structure. In some fields, GAE has demonstrated great performance in some tasks, e.g., multimedia recommendation, link prediction ([21]; [22]). Variational GAE, the variant of GAE, can learn hidden variables with the limit probability distribution on the observed data. The common one is Gaussian distribution. Our experiment will take these models into account to encode graph structure and the parameters of the autoencoder can be learned through minimizing the reconstruction error.

III. Problem Statement

We define the rumor detection dataset format as $C = \{c_1, c_2, ..., c_{|C|}\}$ represent a set of statements and they correspond to label $Y = \{y_1, y_2, ..., y_{|Y|}\}$, where $|C|$ = $|Y|$. Each statement c_i consists of a set of tweets $V = \{v_0^i, v_1^i, ..., v_{n_i-1}^i\}$, v_0^i represents the root tweet of claim c_i, n_i represents the number of all tweets including the root tweet in claim c_i. There are some retweet or response relationships between each tweet. We use $E = \{e_{mn}^i | m, n = 0, 1, ..., n_i - 1\}$ to express the relationship, for example, if v_1^i responses to v_0^i, there is an edge $v_0^i \to v_1^i$, corresponding to $e_{01}^i = 1$, and the adjacency matrix is defined as A, since it is a graph structure, our proposed model can be suitable for the case. As for the text feature $x_i = \{x_0^i, x_1^i, ..., x_{n_i-1}^i\}$, each value is represented by TF-IDF. Fig. 2 shows an example of fake news from the dataset.

We formulate the task as a supervised classification which learns a classifier $g : c_i \to y_i$, where y_i takes fine-grained labels(*True Rumor*, *False Rumor* for Weibo; *True Rumor*, *False Rumor*, *Non Rumor*, *Unverified Rumor* for Twitter).

IV. Model

The core idea of our method is to learn the representation of nodes via propagation and structure reconstruction. As "self-connect" concept was mentioned above, for example, the retweeted or responsive nodes supporting a node would strengthen the stance of the node while deny or suspect (e.g. really?) would weaken its stance. Therefore, through propagation, nodes affect others and are influenced by others at the same time. GCN regard retweet or responsive nodes as neighbors and update all the nodes' representations simultaneously.

As what we see in Fig. 1, structure of rumors and non-rumors formed differently through propagation. Rumors always are described used eye-catching and clickbait headlines to arouse curiosity so that lots of people response or retweet it. When someone stands out and correct it, the rumor would never spread further. These fact reasons illustrate the differences in Fig. 1 between rumor and non-rumor so that the structure information can provide more important rules on rumor detection.

The overall architecture of the proposed model is shown in Fig. 3. For a brief description, we only use one event c_i to discuss how our model works and omit the subscript $_i$.

A. Encoder Module

Let the initial input x_i denotes a post represented as a vector of words in the fixed vocabulary in terms of TF-IDF. A is the adjacency matrix which indicates the relationship between posts. We use the 1stChebNet, employed in the original GCN paper [18], to update each node feature by aggregating its neighbors. Single layer GCN generally works less effective, thus we use two layers to enhance the ability of learning, the equations are written below:

$$H_1 = GCN(X, A) = \tilde{A}\sigma(\tilde{A}XW_0)W_1 \tag{1}$$

$$H_2 = GCN(H_1, A) = \tilde{A}\sigma(\tilde{A}H_1W_2)W_3 \tag{2}$$

where H_1 and H_2 represent the hidden features of two layer GCN; $\tilde{A}$ is the adjacency matrix after regularization $\tilde{A} = \tilde{D}^{-\frac{1}{2}}A\tilde{D}^{-\frac{1}{2}}$, D is the degree matrix; $X \in \mathbb{R}^{n \times d}$ is feature matrix of c; W_0, W_1, W_2 and W_3 are the trainable parameter matrices of GCN; the activation function σ is ReLU function.

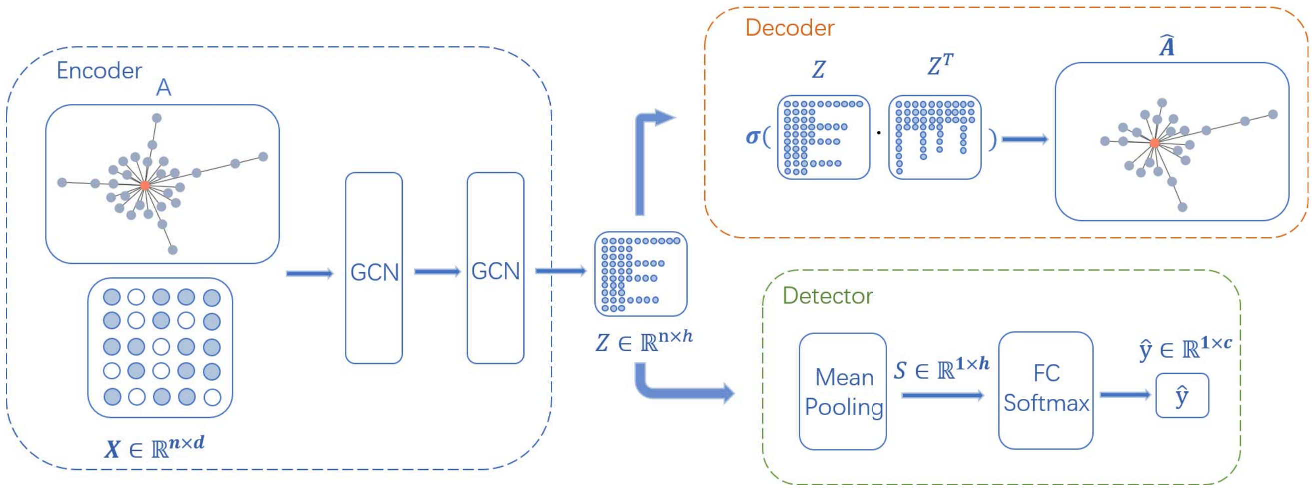


Fig. 3: Overall architecture of the proposed model.

GCN is the core component which accomplish the following three things:

- Since the initial input x is a vector of words using TF-IDF, it represents the sentence semantics of the post. Thus, the update procedure of node features is based on semantics and it contains textual information.
- GCN updates node features via aggregating neighbors' features. It models how the information flows from source post to the current node via propagation. The idea is to generate reinforced features for each post considering its propagation path.
- For subsequent decoding work, the latent representation of structure is also encoded here. The entire structure information can be learned at the same time.

As for Variational GAE, we use GCN to learn a Gaussian Distribution, and then sample z from this distribution. The Gaussian Distribution can be uniquely determined by the mean μ and standard deviation σ which can be learned using GCN respectively. Since sampling can't provide gradient information, the paper used a reparametrization [23] method to construct z and update the gradient. The formula is as follows:

$$\mu = GCN(H_1, A) \tag{3}$$

$$log\sigma = GCN(H_1, A) \tag{4}$$

$$z = \mu + \epsilon\sigma \tag{5}$$

where ϵ is sampled from a standard Guassian Distribution.

B. Decoder Module

The decoder has various approaches to decode latent representation [24]. We use inner product and a sigmoid function to reconstruct the original graph, and the reconstructed adjacency matrix is obtained through the formula.

$$\hat{A} = \sigma(ZZ^T) \tag{6}$$

where $Z \in \mathbb{R}^{n \times h}$ stands for the matrix form of z; σ here is the inner-product operation.

A good z should make the reconstructed adjacency matrix $\hat{A}$ as similar as the original adjacency matrix A. GAE models are trained by optimizing the sum of the reconstruction loss. We apply categorical cross-entropy loss for reconstruction of adjacency matrix. These can be calculated as follows:

$$l_{rec} = \frac{1}{A_{row}A_{col}} \sum m log\hat{m} + (1-m) log(1-\hat{m}) \tag{7}$$

where m and $\hat{m}$ are the elements of A and $\hat{A}$ respectively.

As for Variational GAE, except the reconstruction loss, there is a KL divergence loss need to be added. The KL divergence between two probability distribution measures how much they diverge from each other. Minimizing it means optimizing the probability distribution parameters (μ and σ) as similar as possible to the target distribution (Gaussian Distribution). This can be represented as follows:

$$l_{kl} = -\frac{1}{2n_i} \sum_{i=1}^{n_i} \sum_{j=1}^{n_d} (\mu_{ij}^2 + \sigma_{ij}^2 - log\mu_{ij} - 1) \tag{8}$$

where n_d is the dimensionality of latent features. n_i represents the number of all nodes.

C. Detector Module

Detector Module takes the latent representation as input and aims to classify the event to fine-grained labels. Here we employ mean-pooling operator to aggregate all the node information to event representation. It is formulated as:

$$S = Mean - Pooling(Z) \tag{9}$$

where $S \in \mathbb{R}^{1 \times h}$.

Then we calculate the event label via fully connected layer with softmax function. $\hat{y}$ is calculated as:

$$\hat{y} = Softmax(VS + b) \tag{10}$$

We use cross-entropy loss of the predictions $\hat{y}$ and ground truth y to train the detector component. The loss can be written as follows:

$$l_{dec} = -\sum_{k=1}^{K} y_k log \hat{y}_k \tag{11}$$

where y_k is the ground truth and $\hat{y}_k$ is the prediction probability of a class, K is the number of classes.

D. Training

The complete architecture of our proposed model is illustrated in Fig. 3. The encoder module encodes the textual and propagation information, which is the input of decoder and detector. The decoder aims to reconstruct the data to learn the structure information while the detector aims to classify the event. We jointly train these modules by minimizing the loss over all events and the final loss is computed as:

$$loss = l_{dec} + l_{rec} + Il_{kl} \tag{12}$$

where I is an indicator function that takes value 1 when using Variational GAE model while 0 when using GAE.

V. Experiment

In this section, we first introduce datasets used in the experiment and then we will evaluate our proposed model on the datasets compared with other baseline models.

A. Datasets

We verified our proposed model on three real world datasets including Twitter15 [8], Twitter16[2] [8] and Weibo [1]. They are most famous social sites all over the world. In the datasets, nodes represent the users' tweet while edges represent retweet or responsive relationship, and features are indicated using TF-IDF values. These three datasets have different types of labels respectively. Each root tweet was labeled as True Rumor(T) or False Rumor(F) in Weibo while each root tweet was assigned one of four labels: True Rumor(T), False Rumor(F), Unverified Rumor(U) and None Rumor(N). The datasets were designed by Ma [8] balancedly and the number of each kind of rumors are same, so it helps us evaluate our model properly.

B. Setting

We implement our models using the same set of hyper parameters in our experiments. The batch size is 128. The hidden dim is 64. The total process is iterated upon 50 epochs. The learning rate is 5e-4. We randomly split the datasets and conduct a 5-fold cross-validation. As for validation metrics, we use Accuracy and F1 on Twitter15 and Twitter16 while using Accuracy, Precise, Recall and F1 on Weibo.

[2]https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0

TABLE I: Rumor detection results on Twitter15 and Twitter16 datasets (N: *Non-Rumor*; F: *False Rumor*; T: *True Rumor*; U: *Unverified Rumor*)

Twitter15					
Method	Acc.	N	F	T	U
		F	F	F	F
DTC	0.460	0.481	0.423	0.538	0.392
SVM-RBF	0.234	0.146	0.075	0.220	0.216
SVM-TS	0.544	0.796	0.472	0.404	0.483
MVAE	0.612	0.523	0.656	0.701	0.445
RvNN	0.723	0.682	0.758	0.821	0.654
Only-GCN	0.840	**0.837**	0.851	0.884	0.781
AE-GCN	0.851	0.773	**0.883**	0.857	0.740
VAE-GCN	**0.856**	0.749	0.795	**0.905**	**0.809**
Twitter16					
Method	Acc.	N	F	T	U
		F	F	F	F
DTC	0.548	0.439	0.483	0.743	0.517
SVM-RBF	0.553	0.67	0.085	0.117	0.361
SVM-TS	0.574	0.755	0.42	0.671	0.526
MVAE	0.631	0.540	0.687	0.721	0.578
RvNN	0.737	0.662	0.743	0.835	0.708
Only-GCN	0.852	0.724	**0.892**	0.936	0.7787
AE-GCN	**0.881**	**0.803**	0.844	0.905	0.828
VAE-GCN	0.868	0.795	0.809	**0.947**	**0.885**

C. Baselines

We compare our model with some state-of-the-art baselines on rumor detection.

- DTC [25]: A method using Decision Tree Classifier to detect rumor based on handcrafted features.
- SVM-RBF [26]: A SVM-based model with RBF kernel, using handcrafted features based on the over- all statistics of the posts.
- SVM-TS [27]: 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.
- MVAE [11]: A multimodal variational autoencoder combined with a classifier for the task of rumor detection.
- RvNN [4]: An approach using GRU units to learn rumor representations via tree structure.
- Only-GCN: A model using GCN to learn textual and propagation information without structure reconstruction, which is evaluated in comparative experiment.
- AE-GCN: Our model using GCN as encoder and GAE as decoder.
- VAE-GCN: Our model using GCN as encoder and Variational GAE as decoder.

Note MVAE is a multimodal model for text and pictures, and the datasets we use doesn't contain image information, we modify the model structure of MVAE and only perform

TABLE II: Rumor detection results on Weibo dataset (F: *False Rumor*; T: *True Rumor*)

Method	Class	Acc.	Prec.	Rec.	F1
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.824	0.819
SVM-RBF	F	0.879	0.777	0.656	0.708
	T		0.579	0.708	0.615
SVM-TS	F	0.885	0.950	0.932	0.938
	T		0.124	0.047	0.059
MVAE	F	0.873	0.789	0.777	0.730
	T		0.801	0.719	0.758
RvNN	F	0.908	0.912	0.897	0.905
	T		0.904	0.918	0.911
Only-GCN	F	0.935	0.942	**0.925**	0.929
	T		**0.935**	0.940	0.933
AE-GCN	F	0.942	0.966	0.920	0.939
	T		0.920	**0.969**	**0.940**
VAE-GCN	F	**0.944**	**0.968**	0.921	**0.940**
	T		0.917	0.964	0.936

AutoEncoder on text.

D. *Overal Performance*

Table I and Table II show the model performance on the Twitter and Weibo datasets, respectively.

E. *Analysis*

Table I and Table II show the models' performance on the Twitter and Weibo datasets, respectively. Our proposed model basically achieved better performance than the other methods via capturing the textual, propagation and structure information with GCN.

First, comparing with the previous machine learning methods, it can be seen that the overall effect of the deep learning model is better, which also demonstrates methods based hand-crafted features have low accuracy and inefficiency.

In our experiment, we compare our models with MVAE, a model also using AutoEncoder ideas but based on text information. Like other models based on hand-crafted features, it's indicated that only textual semantics is not enough for rumor detection.

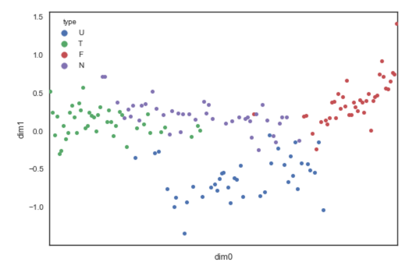
We compare our model with RvNN, a model also learning propagation information but based on GRU. Since RvNN only takes leaf nodes into account and use max-pooling operator so that the classification result heavily depends on the latest posts which are lack of enough information. In addition, GRU units have limitation efficiency via sequential propagation so that GCN module performs better than it. Nevertheless, RvNN still gets decent performance because it captures both textual and propagation information. This demonstrates the propagation plays an important role in rumor detection.

Our model uses GCN to update nodes' representations via propagation. Since the high accuracy of spectrum convolutional methodology on graph data, the experimental result shows that we can learn high-level and better representations. Incorporating with structure representation, our model pays more attention to the final global result of propagation, which helps improve our models much more.

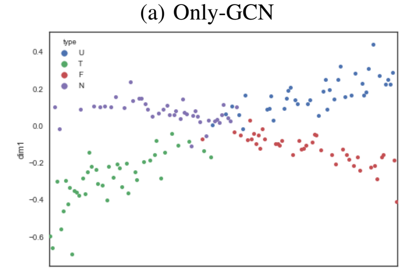
F. *Structure Performance*

With the powerful effect of GCN to capture global features and considering to jointly learn text, propagation and structure information, our models outperform other state-of-the-art models. In order to observe the structure effect on rumor detection, we compare our models with the model based on only GCN, named Only-GCN. We use two layers GCN to classify events without reconstruction of the event structure while other experimental settings are kept the same. In addition, we extract the encoder output latent representation z on Twitter16 (since the results of these three models vary with bigger margin than other datasets') to intuitively see the difference between them. We visualize z in a two-dimensional space by applying the t-SNE algorithm [28]. The result is shown in Fig. 4.

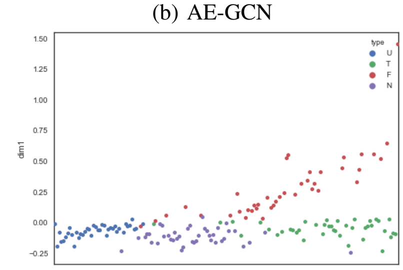
From Table I and Table II, it's seen that AE-GCN and VAE-GCN are better than base model Only-GCN by averagely two percentage points which can be shown that among overall accuracy scores. As for F1 scores, Only-GCN outperforms



(a) Only-GCN



(b) AE-GCN



(c) VAE-GCN

Fig. 4: Visualization on three models: Only-GCN, AE-GCN, VAE-GCN

in one or two specific classes, nevertheless, from an overall perspective our model with structure information performs better.

As can be seen in Fig. 4, different types of events (T,N,U,F) are well separated into four clusters since the decent performance of these three models, especially the model AE-GCN. In detail, points in "Only-GCN" scatter irregularly and there are several events overlapped with each other while points in "AE-GCN" spread around regularly. The two dimensions of VAE-GCN roughly represent the features of mean and standard deviation of its distribution. Therefore, with the limit of Gaussian Distribution, points are nearly aligned horizontally. The results in Fig. 4 validate that by applying autoencoder to learn structure information, we can obtain a better result.

VI. Conclusion

In this paper, we proposed a model which consists of three components: Encoder, Decoder and Detector. We use the core module GCN to update node representation and encode the graph structure. The model is trained by jointly learning the encoder, decoder and the detector to capture text, propagation and structure information simultaneously. Results on three real world datasets Twitter15, Twitter16 and Weibo show that our method improves rumor detection performance and outperforms other state-of-the-art baselines. In our future work, we plan to do some fault analysis and explore result in more detail like why our model has a lower non-rumor performance shown in Table I. We will compare with more sota methods and make a more comprehensive analysis of rumor detection.

VII. Acknowledgment

This research is supported by the Scientific Research Platforms and Projects in Universities in Guangdong Province under Grants 2019KTSCX204.

References

[1] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," 2016.

[2] F. Yu, Q. Liu, S. Wu, L. Wang, T. Tan *et al.*, "A convolutional approach for misinformation identification," 2017.

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

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

[5] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 1201–1211.

[6] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 651–662.

[7] Q. Huang, C. Zhou, J. Wu, M. Wang, and B. Wang, "Deep structure learning for rumor detection on twitter," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

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

[9] J. Zhou, Z. Liu, and B. Li, "Influence of network structure on rumor propagation," *Physics Letters A*, vol. 368, no. 6, pp. 458–463, 2007.

[10] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[11] D. Khattar, J. S. Goud, M. Gupta, and V. Varma, "Mvae: Multimodal variational autoencoder for fake news detection," in *The World Wide Web Conference*, 2019, pp. 2915–2921.

[12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[13] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, 2018, pp. 637–645.

[14] 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 *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] J. Ma, W. Gao, S. Joty, and K.-F. Wong, "Sentence-level evidence embedding for claim verification with hierarchical attention networks." Association for Computational Linguistics, 2019.

[16] L. M. S. Khoo, H. L. Chieu, Z. Qian, and J. Jiang, "Interpretable rumor detection in microblogs by attending to user interactions," *arXiv preprint arXiv:2001.10667*, 2020.

[17] Y. Wang, W. Yang, F. Ma, J. Xu, B. Zhong, Q. Deng, and J. Gao, "Weak supervision for fake news detection via reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 516–523.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[19] Z. Chen, X. Li, and J. Bruna, "Supervised community detection with line graph neural networks," *arXiv preprint arXiv:1705.08415*, 2017.

[20] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Q. Xu, F. Shen, L. Liu, and H. T. Shen, "Graphcar: Content-aware multimedia recommendation with graph autoencoder," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 981–984.

[22] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[25] 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.

[26] 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, pp. 1–7.

[27] 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.

[28] P. E. Rauber, A. X. Falcão, and A. C. Telea, "Visualizing time-dependent data using dynamic t-sne." in *EuroVis (Short Papers)*, 2016, pp. 73–77.