Improving the Early Rumor Detection Performance of the Deep Learning Models By CGAN

Fangmin Dong[2], Yumin Zhu[2], Shuzhen Wan[2], Yichun Xu[1,2*]
[1]Hubei Province Engineering Technology Research Center for Construction Quality Testing Equipments
[2]College of Computer and Information Technology
China Three Gorges University, Yichang 443002, Hubei, China
*Corresponding author email: xuyichun@ctgu.edu.cn

***Abstract*—Deep learning models are recently applied to detect rumors on social media based on the information in the posts. However, at the early stage of rumor propagation, due to the lack of responses, the performance of these models often degrades. In this paper, we propose a method based on the conditional generative adversarial network, which can generate the responses like data and help the deep learning models in early detection. On two large-scale Sina Weibo datasets, the proposed method is applied on the existing convolution neural network model, the recurrent neural network model, and the recursive neural network model. The results show that the proposed method can significantly improve the performance of the models in the case of zero response, and has performance superiority in a certain early period.**

Keywords-Rumor Detection, Deep Learning, Conditional Generative Adversarial Network, Sina Weibo

I. INTRODUCTION1

The nowadays social media provided a place for the people to spread his statements rapidly on the internet; however, the rumors have also proliferated there. The proliferation of rumors has caused significant damage to individuals and society [1]. The social media operators and the government have established the platforms to deal with the rumors, such as WeiboPiyao, but these platforms rely on manual inspection, so they are inefficient. In order to improve the efficiency, machine learning technology has been applied to rumor detection, such as support vector machine, decision tree, and logistic regression [2]. These models work on the data features of the events, which are extracted from the texts and images within the posts, the user profiles, and the propagation structure. However, the data features are also manually extracted and the feature engineering is painstakingly detailed, biased, and labor-intensive. Recently, the deep learning techniques dispense with the complex manual feature extraction and perform the rumor detection by the neural networks, such as conventional neural networks (CNN) [3], recurrent neural networks (RNN) [4], tree recurrent neural networks (RvNN) [5, 6]. The reported results show that the deep learning models generally have better performance compared to traditional machine learning models.

Early detection is very important because it can reduce the damages by the rumor propagation. It should be ideal to make the detection when a user has just posted a source post and no any other user follows him (we later call this case as detection with zero response). On the other hand, since the aforementioned learning models are trained on the events with hundreds of responses, when they make the predictions with less or no response, their performance will certainly decline, because most important patterns may not appear at that time. In comparison, the reported results showed that the CNN or RNN models [3, 4, 7] had better early performance than the traditional methods, and the recent RvNN models [5, 6] gave the improved early detection results above the CNN and RNN models. However, the early detection performances of these models are still a little weak.

In this paper, we propose a method that can improve the early detection performance of some deep learning models. We believe that some conditional relationships exist between the source (first) post and the responses, so that we train a conditional generative adversarial network (CGAN) to generate the response data from source post. In early detection, when there is only a source post or with few responses, the generated data is added to assist the detection. The idea behind is that we adjust the test data and try to make it accord with the distribution of the training data.

On two public Sina Weibo datasets, we applied the proposed methods on the existing CNN model [3], LSTM model [4] and RvNN model [5] respectively. The results show that the improved CNN and RNN models are nearly exceed the original RvNN model in the early detection, and moreover, the RvNN model can also be improved.

II. RELATED WORKS

A. Deep Learning Models for Rumor Detection

In this section, we review the related deep learning models for rumor detection.

CNNs. Convolutional neural network (CNN) is a multi-layer network that uses local connectivity and shared weights to reduce the network complexity and has been applied very successfully in image classification [8]. CNN extracts features by convolutional and pooling computations, and they are suitable for the structured data, and were also used in text classification [9]. Liu et al. used word vector to represents the posts and responses and then applied CNN to perform rumor

This work was supported by the National Science Foundation of China (U1703261).
DOI reference number: 10.18293/SEKE2022-030

detection [10]. Yu et al. divided the posts into groups by time and used doc vector to represent them, finally used CNN to extract high level information and achieved rumor detection [3].

RNNs. Recurrent neural network (RNN) takes sequence data as input, and performs recursive computation in the sequence direction, thus it can capture the historical information of the sequence. Ma et al. first applied RNN to rumor detection and discussed the performance of different RNN types, such as gate recurrent unit (GRU) and long short-term memory (LSTM) [4]. Chen et al. used LSTM network with soft attention mechanism [7]. Xu et al. used user information to perform data preprocessing to improve the performance of RNN [11].

Some other works combined CNN and RNN on the rumor detection. Nguyen et al. used CNN and RNN sequentially to score the reliability of a single tweet and then make the rumor detection by a time series model [12]. Liu et al. made the rumor detection only on the user data, that the data were processed and synthesized by both CNN and RNN [13].

RvNN. Unlike the RNN which performs chain recursion on sequence data, a recursive neural network (RvNN) which makes recursion on a tree structure was proposed by Ma et al. [5, 6]. They found that the propagation of information forms a tree, and the rumor tree and the non-rumor tree have different node relationships-the support and opposition structures are different. So they proposed a tree-structured RvNN to extract these relationships. Recently, the propagation of information were handled with the graph convolutional networks [14-16], where the tree structure were also used.

B. Generative Adversarial Network for Rumor Detection

Generative adversarial network (GAN) was proposed by Goodfellow et al. [17], consisting of a generator and a discriminator, where the generator generates fake data and the discriminator tries to distinguish between the real and fake data. During the training process, the generator and the discriminator compete with each other and the final generated data has the similar distribution as the real data. Currently GAN has been widely used in computer vision, natural language processing, and artificial intelligence [18]. When some conditional parameters are added to the input, we get a conditional generative adversarial network (CGAN), that the data can be generated according to the conditions [19, 20].

GANs were applied in some classification problems, where they were used to generate more data to help the training of the classifiers [21, 22]. But in this paper, we apply the GAN in a different way that the generated data is used in the prediction phase.

The applications of GAN have also been found in rumor detection field. Wang et al. used a GAN to remove the specificity in different types of events, thus their model can extract event-independent features and get good performance [23]. Considering that some rumor mongers tried to mislead the public with pseudo-responses, Ma et al. built a generator to insert pseudo-responses and thus strengthen the discriminator [24]. Song et al build adversary generator by encoder-decoder framework to produce a response for malicious attack [14]. However, the objectives of these works are not related to the early detection.

III. Problem Statement

In social media, most rumors are spreading only with textual information, and the techniques based on texts are the focus of rumor detection field. Other techniques based on multimedia information also need the texts. So in this paper we mainly conducted the rumor detection by the text of the posts. Some relevant concepts are listed below:

Post: The text message posted by a user in social media with a limited number of words. The first post about an event is called the source post;

Response: A comment post made by a user after he read a post. A response can be a comment to the source post, or to a response;

After a user posts a source post P_0 on social media, the post is then read and responded by several other users. Suppose there are N responses to the source post, which are P_1, P_2 ..., and P_N in time order. The sequence E=P_0, P_1, P_2 ..., P_N is called an event.

The machine learning algorithm uses the event data to train a model M, and then uses M to predict a new event being a rumor or not. The early rumor detection takes place at the early stage of propagation, that the number of responses N is very small, even zero. Original models trained with full-life event data will suffer performance degrade in early detection due to the lack of responses. In this paper, we integrate CGAN into the original model to improve its early detection performance.

IV. Early Rumor Detection Method

A. Overall Framework

The proposed method includes three main modules, which are illustrated in Fig. 1 with different colors.

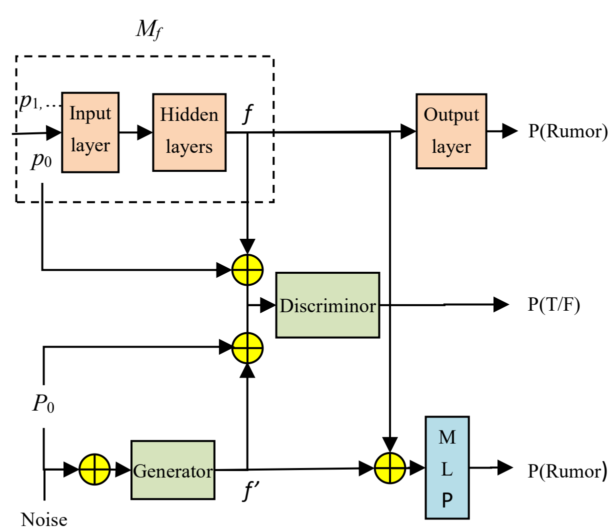


Figure 1. The framework of early rumor detection method

Original Deep learning Module: In the upper part of Fig. 1, from an original deep learning rumor detection model M, we decompose a sub model M_f , which consists of the input layer and all the hidden layers. With the inputs of source post P_0 and the responses P_1, P_2, ..., and P_N, the feature f is obtained through the hidden layers. The output layer is often a linear layer with a sigmoid function.

CGAN module: In the middle part of Fig. 1, the connection between the source post P_0 and the feature f is established through a conditional generative adversarial network (CGAN). The generator outputs the fake feature f' based on the input of P_0 and a noise parameter, while the discriminator tries to judge whether f and f' are real. The symbol $\oplus$ indicates the concatenation of two vectors. After the adversarial training of CGAN, the generator is finally able to generate the features containing the information of responses.

MLP module: In the lower part of Fig. 1, M_f first extracts the feature f from the inputs of P_0, P_1, P_2, ..., and P_N, meanwhile the generator outputs a fake feature f' by the source post P_0, and finally f and f' are synthesized and fed to a MLP (multi-layer perception) to get the rumor detection results.

B. Design of CGAN

The CGAN module includes a generator and a discriminator. The generator concatenates the source post P_0 and the noise as the input, and outputs the generated feature by a three layered fully-connected network,

$$\begin{aligned} &x_0 = (z, P_0) \\ &x_i = relu(w_{i-1}x_{i-1} + b_{i-1}), i = 1,2 \\ &f = \tanh(w_2 x_2 + b_2) \end{aligned} \tag{1}$$

The discriminator is another three layered perceptron, which concatenates the source post and the feature as the input, and then outputs the probability of the feature being real or generated,

$$\begin{aligned} &x_3 = (f, P_0) \\ &x_i = relu(w_{i-1}x_{i-1} + b_{i-1}), i = 4,5 \\ &r = w_5 x_5 + b_5 \end{aligned} \tag{2}$$

Let θ be the parameters (w_0, w_1, w_2, b_0, b_1, b_2), w be the parameters (w_3, w_4, w_5, b_3, b_4, b_5), then the generator and discriminator are denoted by $f = G_\theta(z, P_0)$ and $r = D_w(x, P_0)$ respectively. For the convenience, we use superscript to indicate the id of the event, such as $E^i = (P_0^i, P_1^i, \cdots, P_N^i)$. Let M_f denote the sub model decomposed from the original deep learning model, we use Wasserstein Loss [25] in the training of the discriminator,

$$J_w = \frac{1}{m}\sum_{i=1}^{m}\left(D_w\left(M_f\left(E^i\right), P_0^i\right)\right) - \frac{1}{m}\sum_{i=1}^{m}\left(D_w\left(G_\theta(z^i, P_0^i), P_0^i\right)\right) \tag{3}$$

where m is the batch size of the data. The loss function of the generator is defined as (4),

$$J_\theta = -\frac{1}{m}\sum_{i=1}^{m}\left(D_w\left(G_\theta(z^i, P_0^i), P_0^i\right)\right) \tag{4}$$

C. Design of MLP

The final process of the early detection is finished by a two layered MLP, which concatenates the real feature $f = M_f(E)$ and the generated feature $f' = G_\theta(z, P_0)$ as the input, and then outputs the probability of the event being rumor,

$$\begin{aligned} &x_6 = (M_f(E), G_\theta(z, P_0)) \\ &x_7 = relu(w_6 x_6 + b_6) \\ &p = sigmoid(w_7 x_7 + b_7) \end{aligned} \tag{5}$$

We use the cross-entropy loss to train the MLP,

$$J = -\frac{1}{m}\sum_{i=1}^{m}\log(p_i)y_i + (1 - \log(p_i))(1 - y_i)) \tag{6}$$

where y_i is the label of the event E^i.

V. EXPERIMENTS

A. Three Deep Learning Models to Be Improved

We use the proposed method to improve three recent reported deep learning rumor detection models with different early detection ability. The sub model decompositions are illustrated in Fig 2, where the layers before "feature" make the sub model M_f.

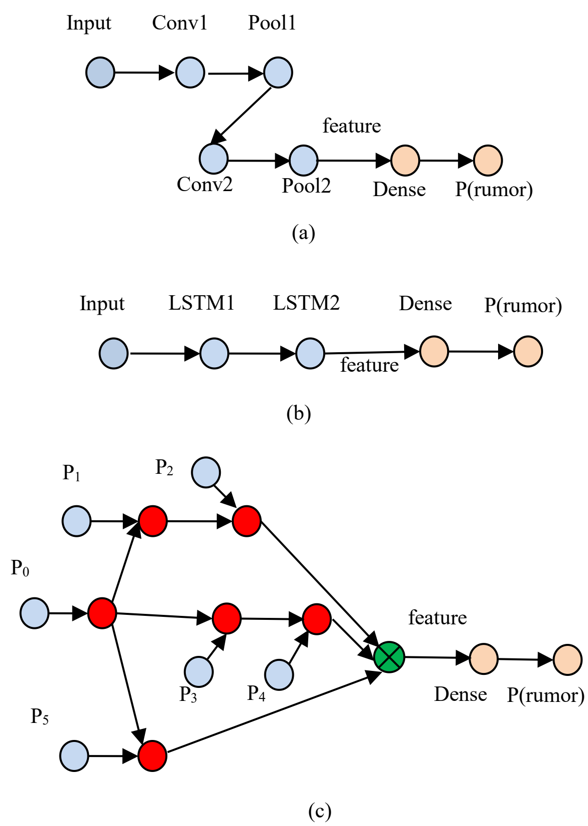


Figure 2. The models of CNN(a), LSTM(b), and RvNN(c)

CNN model [3]. As in Fig 2-a, in the input layer, the authors divided all the posts of an event into groups by time and then used doc vector to represent each group. They used two convolutional lays with two pooling layers as the hidden layers to extract the feature, and a fully-connected layer as the output layer to make the classification.

LSTM model [4]. As in Fig 2-b, the authors also divided the input posts into groups by time and then got a data sequence. After converting the data into vectors, they extracted the feature by two LSTM layers. Finally, they used a fully-connected layer with sigmoid activation to finish the rumor detection.

RvNN model [5]. As in Fig 2-c, the authors found that the propagation of the posts in an event is tree-structured, so they configured an RvNN layer at each node (post) of the tree. Taking the post in a node and the output of its parent node as the input, RvNN layer makes a computation and the results flow to the child nodes. The computation is recursively continued till to each leaf node, and then the outputs of all leaf nodes are pooled to get the feature. The classification is performed by a fully-connected layer finally.

In comparison, the CNN and LSTM models [3, 4] group all the posts and then extract the features on the whole data, while the RvNN model [5] processes each post one by one, so the RvNN model depends more on the pattern of single post and thus it has better early detection ability than the CNN and RNN models.

B. Dataset

The experiments were on two large Sina Weibo datasets, and their statistical information is shown in TABLE I. .

TABLE I. WEIBO AND CHECKED DATASETS.

Name	Weibo	CHECKED
Year	2016	2021
Number of rumor	2313	344
Number of non-rumor	2351	1760
Average # of responses in a rumor	741	48
Average # of responses in a non-rumor	889	664

The first dataset Weibo is a large Sina Weibo dataset reported in the paper [4], with events covering various aspects of politics, economics, entertainment, etc. It is a balanced dataset and each event in it is with a large number of responses, the complete propagation information is also provided.

The second dataset CHECKED is a Sina Weibo dataset recently published in the paper [26], which is special about the COVID-19 events. It is unbalanced dataset with a ratio of positive and negative samples of 1:5. In addition, the average numbers of responses of rumor and non-rumor are very different, that the former is only 48. It should be because that the society is sensitive to COVID-19 rumors that they were ended in short time.

The improvements of the CNN, LSTM, and RvNN were tested on the Weibo dataset. Because the CHECKED dataset does not provide the tree structure information among the responses, so the RvNN was not test on it.

C. Parameter Setting

In the experiments, the same hyper-parameters are used for the CGAN which can be modified according to the models to be improved in practice. For the generator, P_0 needs to be normalized and the noise is standard normally distributed, and they are both 100-dimensional vectors. After the concatenation, a 200-dimensional vector is obtained and then fed to three fully connected layers and the dimensions become 160, 120 and 100 in turns. Because there are positive and negative elements in the real feature vector, we use *tanh* as the active function in the last layer. For the discriminator, the input P_0 and features are both 100-dimensional vectors and concatenated into a 200-dimensional vector, and then it is fed to a three layer perceptron, the dimension is convert to 100, 20, and 1 in turns. We choose RMSprop [24] as the optimizer. The learning rate α is set to 5×10^{-5}, the truncation amplitude c is set to 0.01, the batch size m is set to 128, and the discriminator-generator training ratio n is set to 5.

The input layer of the MLP module concatenates two 100-dimension feature vectors and gets a 200-dimensioal vector. Passing through two fully-connected layers, the dimensions of the vector become 100 and 1 in turn, and with a sigmoid activation, we get the probability of being a rumor. The optimizer of MLP is Adagrad, and the learning rate is set to 0.01。

D. Results and Analysis

The 5-fold cross-validation was performed for each model. In the prediction step, we chose k responses of each event in the time order to test early detection performance of the original models and the proposed model. The accuracy, precision, recall, and F1 are used as the metrics of evaluation. In addition, due to the unbalance in CHECKED, Macro F1 is also used.

1) Performance of the original models on all responses

We first trained the three original models with all the responses. Because CGAN is trained on the features extracted by the original models, their performances are vital to the proposed method. The results in TABLE II show that the trained original models all have good performance, and provide a good basis for the subsequent steps.

TABLE II. MACRO F1 VALUE OF ORIGINAL MODELS.

Dataset	CNN	LSTM	RvNN
Weibo	0.930	0.933	0.912
CHECKED	0.985	0.982	\

2) Results on Weibo

a) Early detection with zero response

TABLE III shows the performance of the proposed method in early detections when there is zero response, where CNN, LSTM and RvNN are the original models and the iCNN, iLSTM and iRvNN are the corresponding improved models.

We find that if only the source post is provided, the performances of three original models are seriously degraded compared to the results inTABLE II. It shows that the responses play an important role in the detection. The accuracies of CNN and LSTM are just 0.504 and 0.582. We noted that the recall of class rumor is quite small which means a large number of rumors are misclassified to be non-rumors. By the help of the generated features, the accuracies of iCNN and iLSTM are increased about 23% and 13% respectively. The macro F1 of them are also improved about 37% and 20% respectively.

Among the three models, the original RvNN has better performance when there is no response that it gets an accuracy of 0.734. The improved model iCNN has already outperformed the RvNN model in accuracy, while iLSTM is closed to it. Moreover, the RvNN can also be improved that the accuracy and Macro F1 of iRvNN are about 1% above the original RvNN.

TABLE III. RESULTS ON WEIBO WITH ZERO RESPONSE

Models	Acc	Macro F_1	Rumor			Non-rumor		
			Prec	Recall	F_1	Prec	Recall	F_1
CNN	0.504	0.335	0	0	0	0.504	1.000	0.670
iCNN	0.737	0.718	0.921	0.494	0.643	0.673	0.961	0.792
LSTM	0.582	0.505	0.862	0.19	0.309	0.549	0.968	0.700
iLSTM	0.716	0.702	0.823	0.519	0.637	0.669	0.897	0.767
RvNN	0.734	0.729	0.689	0.860	0.765	0.810	0.606	0.693
iRvNN	0.743	0.743	0.688	0.926	0.789	0.871	0.544	0.699

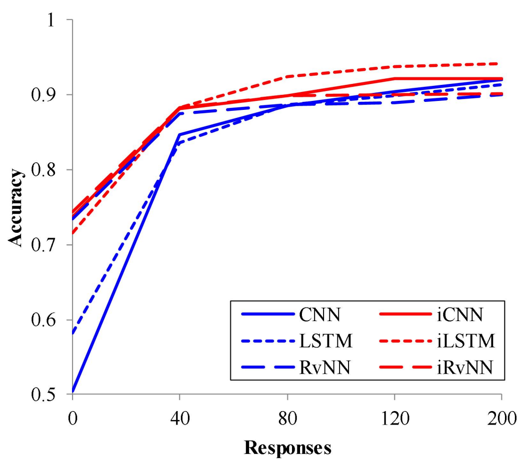


Figure 3. Results on Weibo with some responses

b) Early Detection in different time stages

With the increase of the responses, the accuracies of the three original models increase. However, the proposed method can still improve their performance in certain time stages (Fig. 3). The iCNN model improves the accuracy about 3.4% when there are 40 responses, and it keeps the superiority until there are 200 responses. The iLSTM model improves the accuracy about 4.7% in the first 40 responses, and the improvement is kept above 3% till 200 responses. For the iRvNN model, its improvement is about 1% over the RvNN until there are 200 responses. Among the six models, iLSTM gains the best early detection performance when there are some responses.

According to TABLE I, the average response in each event is less than 900, which indicates that the proposed method can be applied in a long early time stage.

3) Results on CHECKED

The results on the CHECKED dataset are shown in TABLE IV and Fig. 4. Because the responses in rumor class are relatively few, we chose 80 responses at most to test the early detection performance. The proposed method also shows the performance superiority.

In the cases of zero responses, CNN and LSTM have bad accuracy. The precision and recall show that the models tend to wrongly classify most events as rumors. The accuracies of the proposed iCNN and iLSTM are improved above 60% in the case of zero response.

When there are some responses, the accuracies of CNN and LSTM are both very high. However iCNN and iLSTM can still improve the accuracy about 1% and the superiorities are kept in the early 80 responses.

TABLE IV. RESULTS ON CHECKED WITH ZERO RESPONSE

Models	Acc	Macro F_1	Rumor			Non-rumor		
			Prec	Recall	F_1	Prec	Recall	F_1
CNN	0.164	0.142	0.164	1.000	0.281	0.400	0.001	0.002
iCNN	0.905	0.840	0.622	0.903	0.737	0.982	0.905	0.942
LSTM	0.163	0.141	0.163	1.000	0.281	0	0	0
iLSTM	0.845	0.775	0.488	0.967	0.649	0.993	0.824	0.901

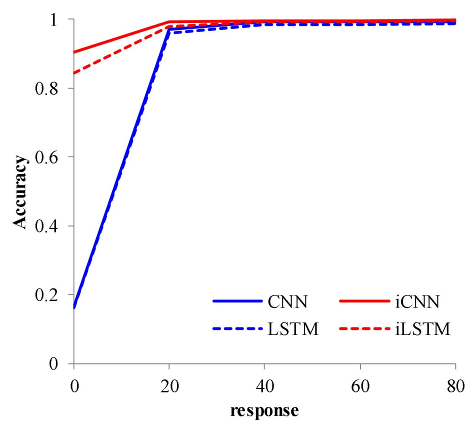


Figure 4. Results on CHECKED with some responses

4) Cases Study

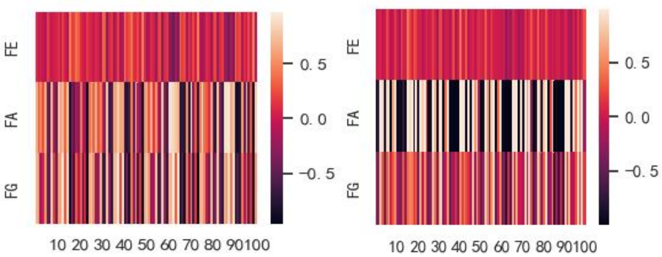
Because the generated features are just vectors of real number, we cannot observe the semantic information directly from them. However, the relationship between the feature generated from the source post (FG), the feature extracted by the original model from some responses in the early stage (FE), and the features extracted by the original model from all the responses (FA) can be investigated through data visualization.

Fig. 5-a is from the event with ID 3514388935498432 in Weibo dataset, which is a rumor correctly predicted by the iCNN model with zero response, but the original CNN model makes wrong prediction. In the figure, the x-axis represents the elements of the 100-dimensinal feature vector, and the color represents the value of the each element. We can see that the magnitude and variation of the elements in the generated feature (FG) is more similar to the feature from all the responses (FA) compared to the early feature (FE) extracted by CNN.

Fig. 5-b shows the different situation of another event with ID 3912024620676243, which is a non-rumor, and the

prediction is on 40 responses. The CNN model classifies it as a rumor incorrectly, but the iCNN model makes a correct prediction. It can be seen that the variation of the elements in FG is still more similar to the FA compared to the FE of the original CNN model.

The case study shows that the CGAN model is able to simulate the distribution of the responses.



(b)

Figure 5. Comparison of the features of FG, FE, and FA from two sample events in Weibo.

VI. CONCLUSIONS

In the early stage of rumor propagation, the performance of existing deep learning models is not high due to the small amount of response data. In this paper, we generate the feature data containing the response information based on the source post by the conditional adversarial generative network, and the generated feature is combined with the real feature to improve the early detection performance. The effectiveness and generality of this method are verified on three deep learning models.

Future work is to apply this method to more deep learning models, and investigate how to make the improvements more effective.

REFERENCES

[1] Y. Liu, "Two Defendants Were Sentenced in Yili Rumor Case ", ed: China News Agency 2018.

[2] G. Liang, W. He, C. Xu, and L. Chen, "Rumor Identification in Microblogging Systems Based on Users Behavior," *IEEE Transactions on Computational Social Systems,* vol. 2, no. 3, p. 10, 2015.

[3] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A Convolutional Approach for Misinformation Identification," presented at the Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017.

[4] J. Ma *et al.*, "Detecting Rumors from Microblogs with Recurrent Neural Networks," presented at the IJCAI2016 New York, USA, July 9–15, 2016, 2016.

[5] J. Ma, W. Gao, and K.-F. Wong, "Rumor Detection on Twitter with Tree-structured Recursive Neural Networks," presented at the Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018.

[6] J. Ma, W. Gao, S. Joty, and K. F. Wong, "An Attention-based Rumor Detection Model with Tree-structured Recursive Neural Networks," *ACM Transactions on Intelligent Systems and Technology,* vol. 11, no. 4, pp. 1-28, 2020.

[7] T. Chen, X. Li, H. Yin, and J. Zhang, "Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection," presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[9] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Eprint Arxiv,* 2014.

[10] Z. Liu, Z. Wei, and R. Zhang, "Rumor Detection Based on Convolutional Neural Network," *Journal of Computer Applications,* vol. 37, no. 11, pp. 3053-3056, 2017.

[11] Y. Xu, C. Wang, Z. Dan, S. Sun, and F. Dong, "Deep Recurrent Neural Network and Data Filtering for Rumor Detection on Sina Weibo," *Symmetry,* vol. 11, no. 11, p. 1408, 2019.

[12] T. N. Nguyen, C. Li, and C. Nieder´ee, "On Early-stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners," presented at the The 9th International Conference on Social Informatics, 2017.

[13] Y. Liu and Y.-F. Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks," presented at the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[14] Y.-Z. Song, Y.-S. Chen, Y.-T. Chang, S.-Y. Weng, and H.-H. Shuai, "Adversary-Aware Rumor Detection," presented at the ACL-IJCNLP 2021, 2021.

[15] Tian Bian *et al.*, "Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks," presented at the AAAI 2020, 2020.

[16] L. Wei, D. Hu, W. Zhou, Z. Yue, and S. Hu, "Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection," presented at the ACL2021, 2021.

[17] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," *Advances in Neural Information Processing Systems,* vol. 3, pp. 2672-2680, 2014.

[18] D. Saxena and J. Cao, "Generative Adversarial Networks (GANs)," *ACM Computing Surveys (CSUR),* vol. 54, pp. 1 - 42, 2021.

[19] D. Chang, W. Yang, X. Yong, G. Zhang, and Y. Wang, "Seismic Data Interpolation Using Dual-Domain Conditional Generative Adversarial Networks," *IEEE Geoscience and Remote Sensing Letters,* vol. PP, no. 99, pp. 1-5, 2020.

[20] H Zhang *et al.*, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV),* 2017.

[21] S. W. Huang, C. T. Lin, S. P. Chen, Y. Y. Wu, and S. H. Lai, "AugGAN: Cross Domain Adaptation with GAN-based Data Augmentation," presented at the ECCV, 2018.

[22] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera Style Adaptation for Person Re-identification," presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-23 June 2018, 2018.

[23] Y. Wang *et al.*, "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection," presented at the 24 th Acm Sigkdd International Conference, 2018.

[24] J. Ma, W. Gao, and K.-F. Wong, "Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning," presented at the The World Wide Web Conference, San Francisco, CA, USA, 2019.

[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN." doi: arXiv:1701.07875 [stat.ML]

[26] C. Yang, X. Zhou, and R. Zafarani, "CHECKED: Chinese COVID-19 Fake News Dataset," *Social Network Analysis and Mining,* vol. 11, no. 1, p. 58, 2021.