Applied Intelligence (2021) 51:4774–4787
https://doi.org/10.1007/s10489-020-02036-0

Detection of rumor conversations in Twitter using graph convolutional networks

Serveh Lotfi[1] · Mitra Mirzarezaee[1] · Mehdi Hosseinzadeh[2,3] · Vahid Seydi[4]

Accepted: 23 October 2020 / Published online: 6 January 2021

Abstract
With the increasing popularity of the social network Twitter and its use to propagate information, it is of vital importance to detect rumors prior to their dissemination on Twitter. In the present paper, a model to detect rumor conversations is proposed using graph convolutional networks. A reply tree and user graph were extracted for each conversation. The reply trees were created according to the source tweet and the reply tweets. By modeling this graph on graph convolutional networks, structural information of the graph and the contents of conversation tweets were obtained. The user graphs were created based on the users participating in the conversation and the tweets exchanged among them. Information regarding the users and how they interacted in the conversations were obtained through modeling this graph on the graph convolutional networks. The outputs of the two above-mentioned modules were combined to detect the rumor. Experimental results on the public dataset show that the proposed method has a better performance than baseline methods.

Keywords Rumor detection · Graph convolutional networks · Conversation · Twitter

1 Introduction

Social networks nowadays play an important role in people's lives. Information can easily be disseminated through these services due to the ease of sharing and high data transfer speed. Twitter is one of the most popular social media networks with a high volume of information and considered a prominent news source, meaning that information often spreads faster than conventional media [1]. In the past, in addition to promoting authentic and reliable information, Twitter has also been misused for spreading false information in the form of rumors. Propagation of rumors in social media has caused financial losses to infrastructures and even threatened human life in the real world. Due to increased concerns and emotional vulnerability, people are more likely to be trapped by rumors and false contents particularly under critical conditions and emergencies [2]. Thus, rumors endanger public safety and mislead people.

Different definitions have been provided for rumor by different sources. The term rumor in the Oxford Dictionary is defined as follows: "Information or a story that is passed from one person to another and may or may not be true''. However, Cai et al. [3] have defined rumor as false information. In the current paper, a rumor is considered a claim that may or may not be true at the time that it is posted on Twitter. Rumor detection on Twitter specifies whether the sets of incoming tweets are rumors or not. In the next step, the veracity of a rumor determines if the result of a rumor ends in true, false, or unproven. In rumor detection, one is faced with two types of rumors. The first type includes long-standing rumors that are discussed for long periods of time and their veracity is not confirmed. And the second type includes rumors which are

✉ Mitra Mirzarezaee
mirzarezaee@srbiau.ac.ir

Serveh Lotfi
serveh.lotfi@srbiau.ac.ir

Mehdi Hosseinzadeh
Hosseinzadeh.m@Iums.ac.ir

Vahid Seydi
v_seydi@azad.ac.ir

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Health Management and Economics Research Center, Iran University of Medical Sciences, Tehran, Iran

[3] Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

[4] Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

generated during breaking news and emergencies. As these rumors are unknown, they need to be automatically detected [4]. Conversations during breaking news play an important role in Twitter as approximately a quarter of Twitter users communicate with each other through conversations and thus a large proportion of tweets are related to conversations [5]. A conversation on Twitter consists of a source tweet and reply tweets in which the source tweet is the initiator of the conversation and the reply tweets are in reply to the source tweet and other reply tweets. If the source tweet in the conversation is a rumor, that conversation is called a rumor and if the source tweet is non-rumor, the conversation is called a non-rumor. Taking into consideration the importance of identifying Rumor Conversations which helps prevent the spread of rumors during breaking news, a method for detecting rumor conversations on Twitter is proposed in this paper.

Previous research has mostly focused on feature engineering in rumor detection including content features, user features and re-tweet propagation graphs. However, it is not easy to extract all suitable features in rumor detection because rumors are usually written across different topics and writing styles. Recently, due to the success of neural networks in other research, neural networks, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Convolutional Neural Networks (CNN) have been used in rumor detection [4, 6, 7]. We argue that in addition to the temporal and contextual features, the structural features are also essential in rumors detection. Thus, the proposed method focused on the characteristics of the conversation graphs and is based on Graph Convolutional Network (GCN) [8]. Hence a reply tree and user graph were extracted for each conversation [9]. By modeling the reply tree on GCN, the content features of the conversation tweets and the structural features are obtained. Using GCN to model the user graphs, the features of users and the interaction of users were obtained. The output of the above two models were combined to increase the efficiency of rumor detection.

The proposed method was evaluated based on the PHEME dataset from Twitter. The experimental results indicated that the efficiency of the proposed model is greater than state-of-the-art methods. In addition, this model demonstrates good performance in early detection of rumors. The main contributions of the present research are as follows:

- A novel method for constructing weighted user graph by following the interaction of users in conversations is proposed. By modeling this graph on GCN, useful information about users participating in the rumors is obtained.
- Simultaneous modeling of user graphs and reply trees with GCN provides key features in rumor detection. These features include content-base, propagation time, structural information of the reply tree, user-base, and structural information of the user graph.
- Experimental results on the public dataset show that our method is better than state-of-the-art methods.

The present paper is organized as follows: Section 2 reviews the research background on rumors in Twitter. The problem statement is addressed in Section 3. Section 4 explains in detail the proposed method and background studies. Section 5 discusses the experimental results and in Section 6, suggestions for further studies are provided.

2 Related works

This paper proposes a model for detecting rumor conversations in twitter events. Related research in this field includes classification approaches which are based on traditional machine learning and deep learning in detecting and verifying rumors and the study of GCN applications in different scientific fields.

2.1 Approaches based on traditional machine learning

Various attempts have been made to detect and verify rumors in on-line social networks. The early work performed to identify rumors mostly considered feature engineering by means of machine learning techniques. These features included content-based, user-based, temporal-based and propagation-based features [4].

Among the most prominent work in this field is a study by Castillo et al.[10] which provide an automated method to assess information credibility on Twitter. They used four categories of features consisting of message-based (including Twitter-independent (such as the length of a tweet, exclamations, number of urls, and question marks) and Twitter-dependent (such as hashtags, emotion smiles)), user-based, topic-based, and propagation-based features. Finally, 15 features were selected using the feature selection method which achieved the maximum accuracy 89% on a handcrafted dataset in the J48 learning algorithm. The above-mentioned study was used as the main reference for most related work that provide an automatic method for evaluating the credibility of news topics. Qazvinian et al. [11] provide a system to detect rumors previously identified. Their experiment on five different topics achieved 94% accuracy. This kind of rumor detection is more effective for exploring the beliefs of people who talk about those rumors over a long period of time. Kwon et al. [12] offer a new set of linguistic, structural and temporal features including 10 temporal features related to the propagation of rumors in the intended time period. 15 structural features were extracted based on the connection between users who talked about rumors and 65 linguistic features based on the language used in the content. Their findings demonstrate that temporal features are effective in rumor detection. Finally, 11

features were selected using the feature selection method. The selected features in the Random Forest classifier achieved a maximum accuracy of 90%. Zhao et al. [13] state that in rumors, most of the users are skeptics and have greater doubt. They used the enquiring behavior of users on social media. Thus, they created a manually curated list of five regular expressions that were used to identify enquiring tweets. The enquiring tweets together formed a cluster and each cluster was identified as a candidate for a rumor. Zubiaga et al. [14] use the source tweet in each conversation on the PHEME dataset to detect rumors and hypothesize that due to a lack of textual content, it is impossible to determine whether a tweet is a rumor or not with just a single tweet in a story. To test their hypothesis, they used a sequential classifier based on Conditional Randomized Field (CRF). They used the sets of content-based (including number of question marks, number of exclamation marks, number of periods, ratio of capital letters, word vectors) and social-based features (including number of tweets written by the author, number of lists, number of followers, verification of author's account age). The evaluation results of the combined content features and users on the CRF classifier and non-sequential classifiers indicated that the CRF classifier was more efficient than other non-sequential classifiers. The CRFclassifier achieved 60.7% F-score for the rumor class. Vosoughi et al. [15] create a human-machine collaborative system on Twitter for detecting rumors of real-world events. They entered the event-related raw tweets in the assertion detection system; tweets with similar assertions were classified in a single cluster. In the next step, the user could detect if the clusters were rumors or not. Vosoughi et al. [16] assess the veracity of rumors using three categories of features: linguistic, user, and temporal propagation. They selected 17 features. Using the Hidden Markov Model, the veracity of rumors achieved 75% accuracy. Their results indicated that the temporal propagation feature was more efficient than linguistic and user features. Jahanbakhsh et al. [17] focus on speech act identification of Persian texts developing a dataset of Persian Telegram rumors to determine the common speech acts in these rumors. They showed the positive effect of speech act on rumor detection by combining the common content features and four speech act classes. Giasemidis et al. [18]provided the results of their experiment on 72 different rumors that had one hundred million tweets. They used the classifiers and features of previous work at various time windows from the outbreak of the rumor and obtained good results using a decision tree.

2.2 Approaches based on deep learning

In recent years, deep learning has been successfully applied in many areas of artificial intelligence such as Natural Language Processing (NLP) and traffic flow prediction [19, 20]. The two most widely implemented paradigms in neural networks are RNN and CNN. CNNs have been widely studied for image recognition and processing [21], crowd density estimation [22] and character verification on integrated circuit components of printed circuit boards [23]. RNN and LSTM are particularly effective for modeling sequential data such as language modeling [24] and text classification [25].In recent years, deep learning models such as RNN, LSTM, GRU, and CNN have been used to detect rumors on Twitter. Ma et al. [26] apply the Recurrent Neural Network Model using deep learning in their early work to identify and verify rumors. They observed that a rumor event consists of an original post and a set of related posts including re-tweets and replies which create a continuous stream of posts. Thus, they modeled the rumor data as a variable length time series. They put a set of tweets with equal time intervals as a unit in a time series that was then modeled using the recurrent neural network. Their model was more efficient than other models with used feature engineering. Their experiment on Weibo and Twitter datasets achieved 90% and 88%accuracy. Ajao et al. [27] provide a model to detect fake news messages using a combination of CNNs and LSTM. They extracted the features of tweets and images from the conversations and argued that their utilized method could identify fake news stories without the need of previous knowledge of the domain. Their experiment using the LSTM-CNN model on the PHEME dataset achieved 40% F-score. Xu et al. [28] present a Merged Neural Rumor Detection model to detect rumors based on the content of the source tweet, re-tweets and information of users. They used the attention mechanism to focus on keywords of the source tweet and important re-tweets in the propagation process. Chen et al. [29] provide an RNN-based deep attention model called CallAtRumor. This model learns the temporal hidden representation of sequential tweets. In rumor detection, CallAtRumor determined different features by learning latent representation from sequential tweets. Their experiment on Weibo and Twitter datasets achieved 87% and 87% F-score. Ma et al. [30] use a Generative Adversarial Networks (GAN)-style model to detect rumors where the discriminator is used as a classifier and the corresponding generator attempts to improve the discriminator by removing inconsistent noises. In this model, the generator attempts to produce controversial opinions using a seq2seq model based on GRU, making the distribution of tweets' viewpoints more complex, and tries to identify stronger features of rumors from augmented samples in the discriminator using RNN. The results of experiments on public PHEME and TWITTER datasets showed that the GAN-style model is very robust and effective. They achieved 78% accuracy on the PHEME dataset and 86% accuracy on the Twitter dataset. Alsaeedi et al. [31] propose a deep learning model based on a CNN layer and dense layers to build a rumor identification model. They performed different experiments to find the best hyper parameter settings to improve the model's performance. The PHEME dataset was used to train their model and achieved 87% accuracy on the dataset.

Liu et al. [32] use LSTM coupled with pooling operation of convolutional neural networks to detect rumors based on forwarding contents, spreaders and diffusion structures Experiments showed the forwarding content as the input of the combination of LSTM and max pooling achieved 95% accuracy on the Weibo dataset and the best performance. Santhoshkumar et al. [33] propose a CNN-based method implemented with two CNN networks; in the input to the first neural network all posts related to an event are collected and converted into a variable length vector and then the input to the second CNN network is vectorized by leveraging the respective temporal and structural information of posts of the first input. They combine and use the outputs of CNNs to generate the final classification results via a decision tree. Zubair Asghar et al. [34] provide a deep learning model called BiLSTM-CNN. By using a BiLSTM layer, the long term dependency in a tweet based on both past and future context information is obtained and the CNN layer is used to extract features of the tweet by processing the information in a hierarchical manner for efficient classifications as rumor and non-rumor.

The use of deep learning models on graphs has recently been shown to have high efficiency in application domains [35]. Compared to other deep learning models, GCN covers the overall structural features of graphs. Today, GCN is widely applied in different applications such as computer vision, natural language processing, physics, chemistry, biology, social network analysis [36] and bot detection [37]. In social network analysis, different issues such as the identification of community [38], link prediction [39] and re-tweet counts anticipation are dealt with [40]. Dong et al. [41] proposed a GCN -based model that could identify several rumor sources without prior knowledge of the underlying propagation model.

Despite the use of different models in rumor detection, these models do not cover the structural information of two user graphs and reply trees simultaneously with deep learning models on graphs. In the current study, how users interact is obtained by modeling the user graphs on GCN in addition to the features of users participating in the conversation. Moreover, content, propagation time and structural information of the conversation are also covered by modeling the reply tree on GCN. Finally, in order to comprehensively cover the features, outputs of the two modules are integrated to detect rumor conversations.

3 Problem statement

In this paper, the dataset D is defined as D= {(C_1, y), (C_2, y), (C_3, y)… (C_n,y)}, where C_i is a conversation in the dataset and y={R, N} indicates the conversation label, R is a rumor and N is a non-rumor. In C_i = {s_i: r_{i1}, r_{i2}, r_{i3}…r_{im}}, m refers to the number of reply tweets, s_i is the source tweet, and each r_{ij} represents the reply tweets. Figure 1a shows a rumor conversation in the Ottawa shooting event consisting of a source tweet and eight reply tweets. C_i can be shown with reply tree as graph G= (V, E), where V represents the graph nodes (source tweet and reply tweets) and E represents the edge of the graph [9]. Figure 1b illustrates the reply tree of Fig. 1a. which is related to the rumor conversation of the shooting event in Ottawa. The tweets are tree nodes and the edges are directional, their direction being from one tweet that is in response to another. In this tree, five tweets directly replied to the original source tweet and three tweets replied to other tweets. In u_i= {u_{si}: u_{ri1},u_{ri2},…u_{rik}} where each u_{rij} is the user of reply tweet, u_{si} is the user of the source tweet, and k is the number of users participating in a conversation. Since a user can have more than one tweet in a conversation, the number of participating users is usually less than the number of conversation tweets. The users participating in C_i can be shown with the graph G= (V, E), where V represents the nodes and E represents the edge of the graph [9]. Figure 1c shows the directional user graph of Fig. 1a, which is related to the rumor conversation of the shooting event in Ottawa. In this graph, the nodes represent the users, and the edge exists if one user replies to another. As observed, the edges of this graph are weighted, and their weights indicate the number of tweets connecting users to each other. In this diagram, five users directly replied to the user of the source tweet. A total of seven users participating in the conversation are connected via nine tweets.

In general, rumor detection is a binary classification problem that determines whether a conversation is a rumor or not. The classifier can be formalized as a function where y is rumor or not rumor. In this study, in order to have a more efficient classifier, the reply tree and the user graph were modeled on GCN.

4 The proposed method

In this section, details of the proposed method are provided. This model consists of three modules: User, Reply tree and Concatenate. Figure 2 illustrates the proposed method. The two modules are obtained by modeling the user graph and reply tree on GCN. In Concatenate, the features obtained in the two modules, mentioned above, are combined in a fully connected layer to detect the rumor conversation. The details of the model are described below.

4.1 Graph convolutional networks (GCN)

GCN is a multi-layered neural network designed directly on graphs and provides embedding vectors of nodes based on properties of their neighborhoods [8]. The objective of a GCN is to learn a function of features on a graph G= (V, E). The features of the graph nodes are shown using the feature matrix $X \in R^{n*m}$, where n is the number of nodes and m is the dimension of the feature vectors. GCN takes feature matrix

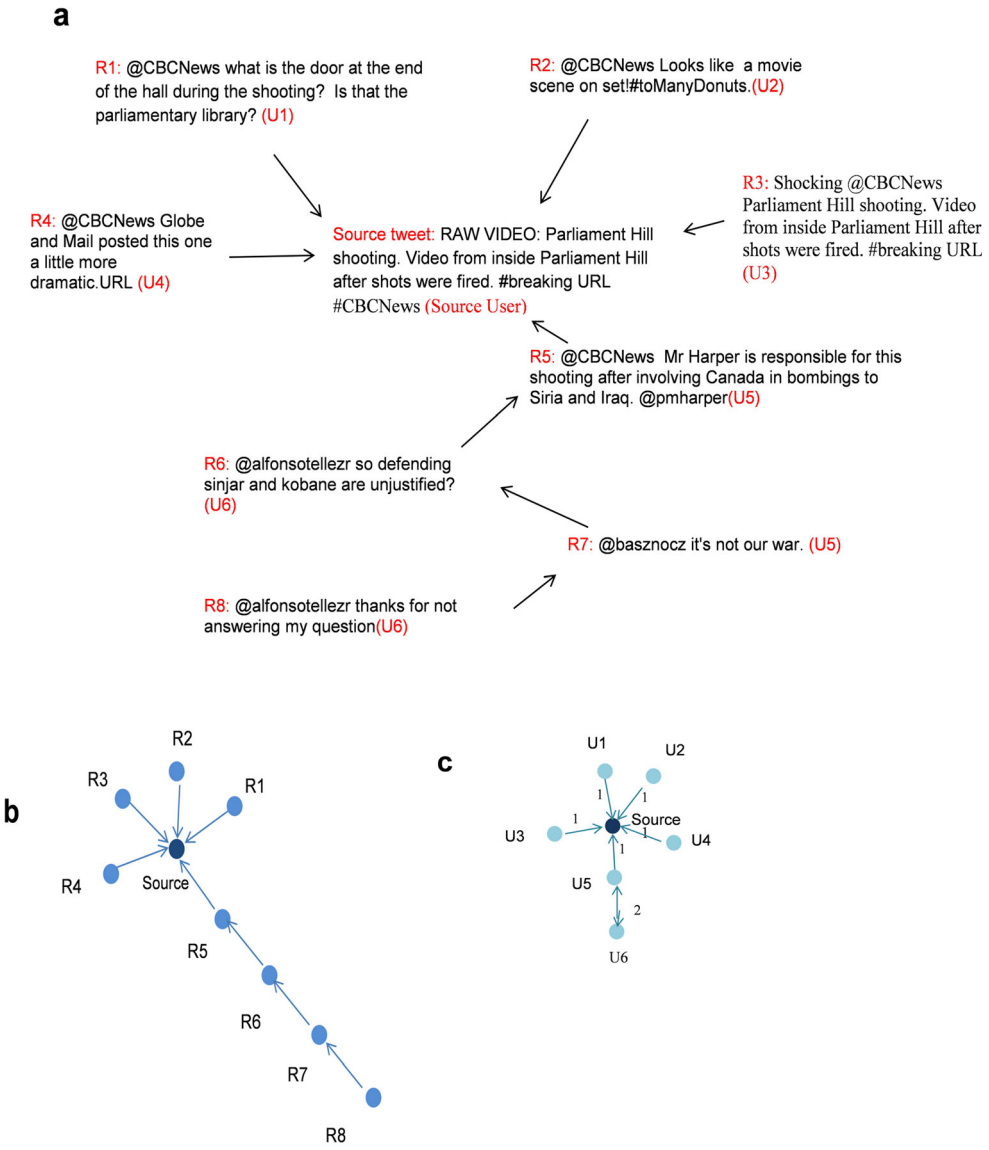

Fig. 1 **a** The source tweet and reply tweets of a rumor conversation of the Ottawa Shooting. **b** Illustration of the reply tree of **a**, and **c** Illustration of the directional user graph of **a**

and adjacency matrix A as inputs. In addition, GCN produces an output feature matrix $Z \in R^{n*c}$ where n is the number of nodes and c indicates the number of output features. Each hidden layer in the GCN can be expressed as follows in Eq. 1.

$$H^{l+1} = f(H^l, A) \tag{1}$$

A simple layer-wise propagation rule in the GCN is as follows in Eq. 2:

$$H^{l+1} = \rho(AH^l W^l) \tag{2}$$

Where $H^0 = X$ and W^l is a weight matrix for layer l and ρ is an activity function like ReLU: $\rho(x) = \max(0, x)$. Each layer H^l corresponds to an $N \times F^1$ feature matrix where each row is a feature representation of a node. At each layer, these features are aggregated to form the next layer's features using the propagation rule. There are two major improvements to Eq. 2; in the first case, each node being connected to itself, so each node will include their own features in the aggregate. As a result, the diagonal elements of the adjacency matrix are set to 1. In the second case, the feature representations can be normalized by multiplying the adjacency matrix A with the inverse degree matrix D [8]. Both of these improvements bring us to the final propagation rule in Eq. 3. In this equation, $\widetilde{A}$ is the normalized symmetric adjacency matrix which is calculated as $\widetilde{A} = D^{\frac{-1}{2}} A D^{\frac{1}{2}}$ and degree matrix, D, is calculated through the adjacency matrix A as $D_{ii} = \sum_j A_{ij}$.

$$H^{l+1} = p\left(\widetilde{A}, H^l, W^l\right) \tag{3}$$

The more GCN layers are aggregated on each other, the more information can be obtained on the neighboring nodes.

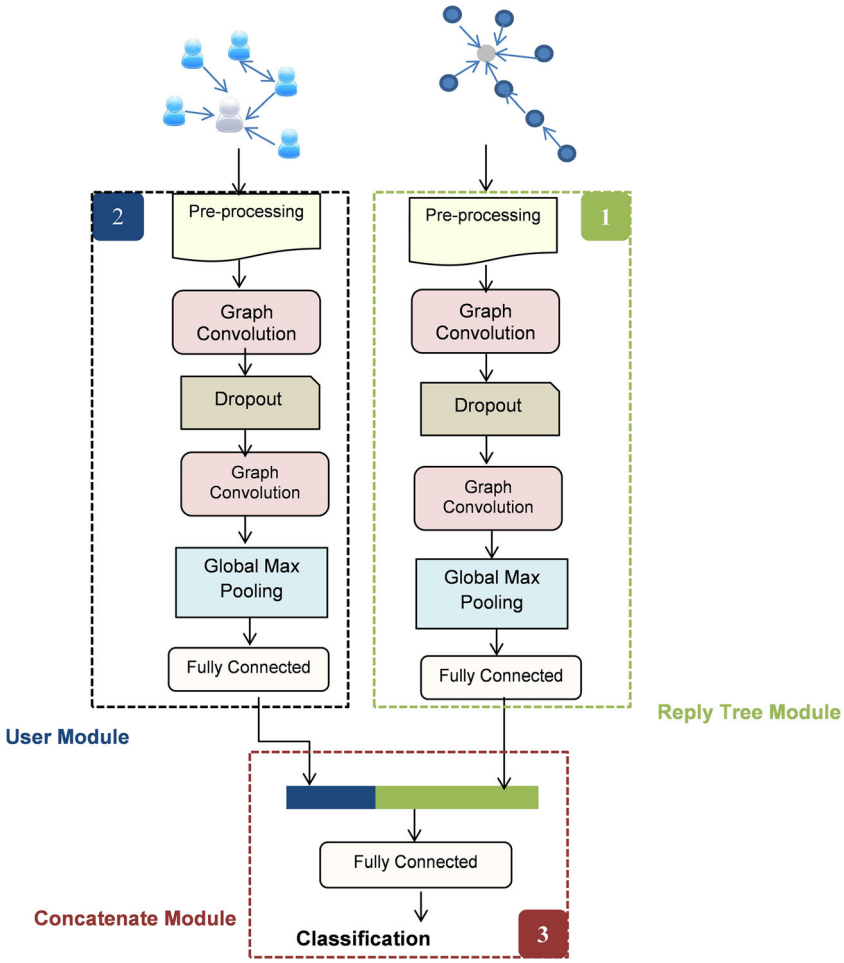

Fig. 2 An illustration of the proposed model

The output of GCN layers yield high-level node representations which use a softmax layer as the output layer enabling GCN to perform node classification task. To obtain a compact representation at the graph level, GCN is often combined with pooling and readout operations and able to perform graph classification task [42, 43].

4.2 Pre-processing

Dealing with graphs with a variable number of nodes and representing a group of graphs in batch mode requires padding A and X to a fixed dimension [44]. In Fig. 2, in the pre-processing operation on each graph, dimensions of the adjacency matrix transforms to a fixed dimension. If the dimensions of the adjacency matrix are less than the fixed dimensions, the dimensions are added using the zero method. If the number of dimensions is greater, the extra dimensions will be removed. Pre-processing in the conversation tweets involves the removing of stop words defined in NLTK as well as removing mentions, hashtag signs and additional figures. Tokenizing and stemming operations are also performed on the conversation tweets.

4.3 Reply tree module

In this module, the purpose was to obtain content, propagation time, and structural information of the reply tree in the conversation. Based on the response relationships, a reply tree was constructed for each conversation. For modeling the reply tree on GCN, the feature matrix and adjacency matrix of the graph were generated and the pre-processing operation was performed on it. In the feature matrix, the high frequency words of the source tweets were extracted from the training file conversations according to the importance of the source tweet in rumor detection. Moreover, based on the importance of time in identification of rumors [45], the propagation time interval between the reply tweets and the source tweet was obtained by calculating the propagation time interval between the two nodes in minutes and time was considered as a feature in the feature matrix. The feature matrix X of the reply tree is composed of high frequency words of the source tweets and propagation time. All of these high frequency words were indicated by a binary feature. In the feature matrix, X_{ij} is set to 1 if there is high frequency word j in the tweet i.

$$X_{ij} = \begin{cases} 1 & \textit{if there is high frequency word } j \textit{ in the tweet } i \\ t & \textit{the propagation time interval between the reply tweet } i \textit{ and the source tweet} \\ 0 & \textit{otherwise} \end{cases} \quad (4)$$

The values of the adjacency matrix A are defined in Eq. 5. In the adjacency matrix, Aij is set to 1 if tweet i replies to tweet j and the elements of the diagonal elements of the adjacency matrix are set to 1.

$$A_{ij} = \begin{cases} 1 & \textit{if tweet } i \textit{ replies to tweet } j \\ 1 & i = j \\ 0 & \textit{otherwise} \end{cases} \quad (5)$$

In the current study, two simple layers of GCN were used [8]. Results of the experiments indicated that two layers were more efficient than one layer and more than two layers of GCN. In GCN layers, a dropout layer was used to prevent overfitting and generalization of the model. The GCN layer output gives high-level node representations that combine content, propagation time and structural features. Here, max pooling operators were employed to aggregate information from the node representations. By examining different pooling layers (sum pooling, mean pooling, and max pooling), it was found that the max pooling layer was more efficient than the other layers. In Section 5.2.1, the results of the experiments are shown. The output of the max pooling layer is a vector of features, each feature having a maximum value in the feature matrix of the GCN layer output. For greater model performance, the max pooling output vector was given to the fully connected layer.

4.4 User module

In this module, user features and graph structure information was obtained. Based on the interaction of users in the conversation, a weighted user graph was constructed for each conversation. For modeling the user graph on GCN, the feature matrix and adjacency matrix of the graph were generated and the pre-processing operation was performed on it. The feature matrix X of these users was composed of the number of followers, the number of friends, the number of lists, the user's verified identity on Twitter, the number of user tweets liked by others, the user's profile description, the user's location information and length of time that the user had signed up for Twitter. In the adjacency matrix in Eq. 6, the value of Aij is set to m which shows the weight of edge between node i and node j in the user graph. Edge weight shows the number of tweets that user i has sent to user j in the conversation. Aij is set to 1 if the element is related to the diagonal elements of the adjacency matrix.

$$A_{ij} = \begin{cases} m & \textit{user } i \textit{ has sent } m \textit{ tweet to user } j \textit{ in conversation} \\ 1 & i = j \\ 0 & \textit{otherwise} \end{cases} \quad (6)$$

In this module, the same operation as the reply tree module was performed on the GCN layer output.

4.5 Concatenate

In this step for each conversation, the output vector of the user module was combined with the output vector of the Reply tree module and given to a fully connected layer to calculate the label of the conversation. In combining the above two modules, more accuracy was obtained as the model simultaneously learns from the structural information of the propagation tree and users. This module almost comprehensively covers information regarding rumor detection: contents, propagation time, users, and propagation.

4.6 Model training

The final purpose for each training sample was to minimize the rate of error between the predicted value and the true value. In this paper, the binary cross entropy was used for the loss function as follows in Eq. 7:

$$H(Y,P) = -\frac{1}{N}\sum_{i=1}^{N} y_i.\log\Big(\mathrm{p}(\mathrm{y_i}) + (1-\mathrm{y_i}).\log(1-\mathrm{p}(\mathrm{y_i}))\Big) \quad (7)$$

where P is the predicted value of the class, y is the true value of the class and N is the number of sample. All parameters of the model were trained using back propagation; the RMSprop optimization algorithm was used on all parameters in training [46].

5 Experiments and results

In this section, the performance of the proposed GCN-based model is experimentally investigated in comparison with several baseline models to detect rumor conversations in Twitter events.

[0] https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619.

5.1 Dataset

In the present research experiments, five sets of real-life conversations from PHEME related to a piece of breaking news were used: the Charlie Hebdo shooting, Ferguson unrest, Ottawa shooting, Sydney hostage crisis, and Germanwings crash. This dataset is part of the PHEME project and is available to the public[1] [14].

Annotation of conversations for all the events created a set of 5,802 conversations: 1,972 conversations related to rumor conversations and 3830 conversations related to non-rumor conversations. As seen in Table 1, there are different distributions for these five events. For example, in the Germanwings crash more than 50 percent of the conversations concerned rumors while in the Ottawa shooting, this figure declined to less than 25%. Since the data was an imbalanced one, the Random oversampling technique was used to balance the training file [47].

5.2 Parameter settings

The conversation graphs and GCN layers were implemented using NetworkX and Spektral libraries. The number of features in the feature matrix of the reply tree and the weighted user graph were 800 and 8, respectively and the number of neurons in the two fully connected layers were set to 512 and 2.

5.2.1 Experiments on model depth

Here, another factor must be taken into account when using GCN: performing multiple graph convolutions to investigate classification performance. Kipf and Welling [8] reported effects of the model depth for GCN. In the present study experiments on model depth were evaluated on the k-fold data of the GAN-GRU method [30] with 5-fold cross-validation. On each cross-validation split, 300 epochs were trained for (early stopping set to 50 and restore model weights from the epoch with the best value of the monitored quantity) using the RMSprop optimization [46]. The dropping rate in the first and last layers were set at 0.5. The batch size was set to 10 for all tasks and validation set at 0.1 which included the training file. In addition, instead of max pooling layer in the proposed method, the sum pooling and mean pooling were examined. The output of the sum pooling (or mean pooling) layer is a vector of features; each feature having a sum (or mean) value in the feature matrix of the GCN layer output. Results are summarized in Fig. 3 which illustrate mean classification accuracy (training vs. testing) for 5-fold cross-validation.

Six Convolutional layers were considered for the proposed model here; the best result was obtained with a 2-layer model as the number of layers increases, the accuracy of the model decreases. However, in the User-Reply-GCN (sum pooling) and User-Reply-GCN (mean pooling) models, best results were obtained with a 4- and 1-layer model. As shown in Fig. 3, the choice of pooling layer affects the number of convolution layers.

[1] https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619.

Table 1 Frequency of rumor and non-rumor conversations

Events	Rumors	Non-rumors
Charlie Hebdo shooting	458 (22.0%)	1,621 (78.0%)
Ferguson Unrest	284 (24.8%)	859 (75.2%)
Germanwings crash	238 (50.7%)	231 (49.3%)
Ottawa shooting	470 (52.8%)	420 (47.2%)
Sydney hostage	522 (42.8%)	699 (57.2%)

5.2.2 Experiments on the number of graph nodes

For graphs with a variable number of nodes and representing a group of graphs in batch mode requires padding where dimensions of the adjacency matrix and feature matrix transform to a fixed dimension. Experiments on the proposed model were performed to determine the number of nodes in the tree reply and user graph. In this experiment, the proposed model on the k-fold data of the GAN-GRU method [30] with 5-fold cross-validation similar to Section 5.2.1 was evaluated. The accuracy was calculated in five intervals based on the number of tweets in conversations. Results are summarized in Fig. 4 where the x-axis is the number of tweets in conversations and the y-axis on the left represents the accuracy. As observed the greatest accuracy of the number of tweets is equal to a constant 10; to increase the number of tweet mean classification accuracy (training vs. testing) is reduced because more rows with a value of zero are added to the adjacency and feature matrix.

5.3 Baselines

The proposed model was evaluated by several basic models including the feature-engineering-based (CRF, DT, RF) and neural network-based (GAN-GRU,CNN,GRU) methods as outlined below:

CRF: for the source tweet, social features and content are extracted and sequential classification based on CRF is used [14].

DT: in this method, a Decision Tree classifier is used to validate the Twitter information. The set of used features includes the message-based, user-based, topic-based, and propagation-based features [10].

RF: the Random Forest is applied to detect rumors using the linguistic, structural and temporal features [12].

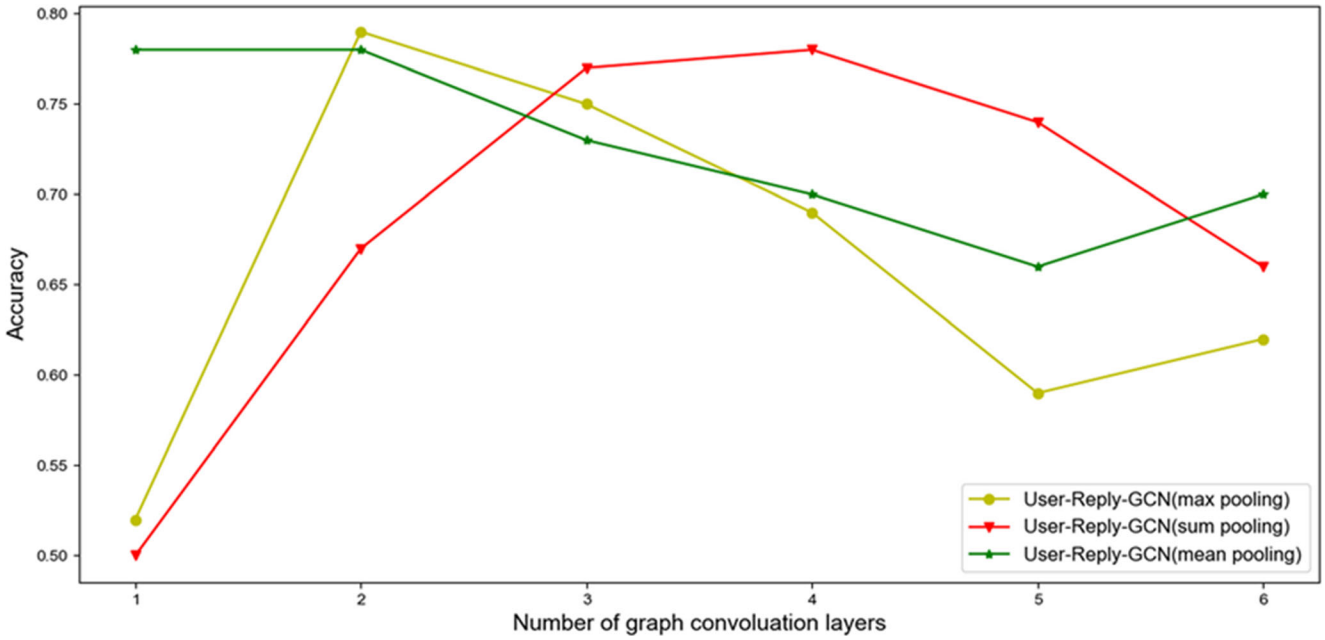

Fig. 3 Influence of model depth (number of layers) on classification performance in the proposed model with different pooling layers

A 5-fold cross validation similar to the CRF method [14]was performed. In each run, four events were used in training of the proposed model like the CRF method and the fifth event was then used to evaluate the performance of the model in terms of recall, f1 and precision. In the present study, the 5-fold cross validation in RF, DT and the User-Reply-GCN of the proposed model was performed in similar to the CRF method. The results of the experiments, the micro-averaged scores of all five runs of precision, recall, and F1 of the rumor and non-rumor classes are shown in Table 2. The bold values in Table 2 indicate the efficacy of the best classifier compared to other classifiers. As can be observed in Table 2, recall and F1 results are higher in the rumor class of the proposed model and the precision in CRF has a greater efficiency, while F1 in the non-rumor class is the same in both methods. RF and DT methods are based on feature engineering and have a worse performance than the other two methods since their structural features are related to the retweet propagation tree whereas this dataset is focused on conversations that have reply tree graphs. In the CRF method, the problem with this method is that if the initial tweets are not properly labeled, the efficiency of the classifier in rumor detection is

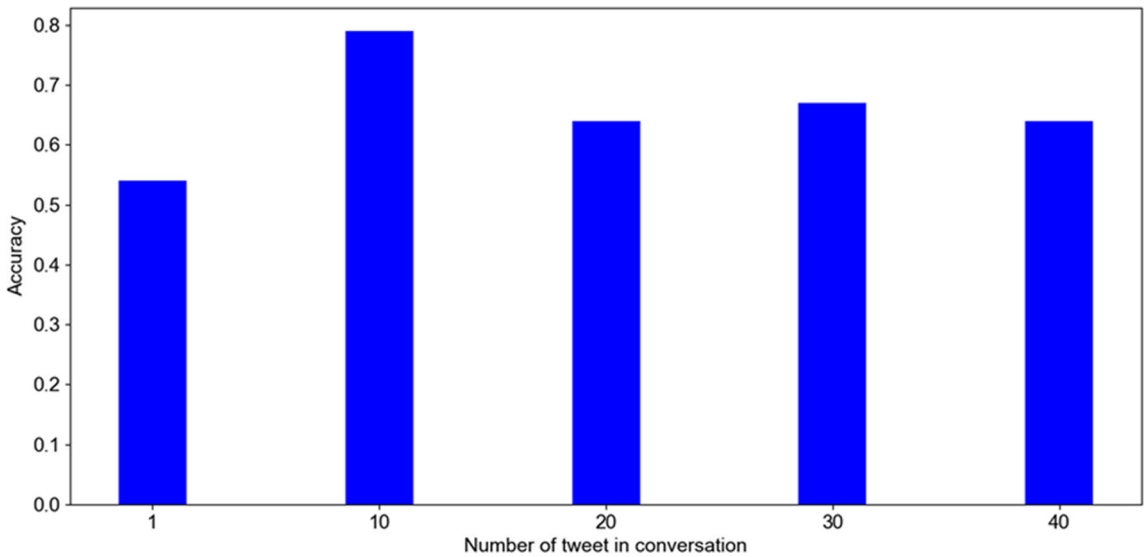

Fig. 4 The effect of selecting the number of tweets in conversations on classification performance. The y-axis denote mean classification accuracy (training vs. testing) for 5-fold cross-validation

reduced. Thus, if the previous tweets are not appropriately classified, the classifier efficiency will decline. Compared to previous methods, the proposed method classifies each conversation with ten tweets at most independently.

GAN-GRU: Ma et al. [30] provided a method for detecting rumors using Generative Adversarial Networks with a GRU-based discriminator.

CNN: Alsaeedi et al. [30] used CNN to detect rumors. The proposed CNN model includes four dense layers and one Conv1D layer. They obtained the best hyperparameter for their model on the PHEME dataset.

GRU-RNN: Ma et al. [26] used the RNN with GRU unit to obtain the representation of the tweet set by modeling the sequential dynamics of reply tweets.

Ma et al. [30] using the GAN-GRU method evaluated their model on the dataset used in this paper and filtered out conversations with less than 10 tweets and balanced the number of samples of the two classes. Their k-fold data is available to the public[2]. In this study, CNN, GRU-RNN and the proposed methods were evaluated on the k-fold data of the GAN-GRU method using 5-fold cross-validation and precision,recall and F1 were used as evaluation metrics for the two classes of rumors and non-rumors. For CNN and GRU-RNN methods, the default values of the parameters given in the cited papers were used. If the values of the parameters were not given explicitly, the parameters were tuned to find the optimal parameters. For comparative methods, the word2vec algorithm was used to learn representations of tweet content, and the preprocessing operation was performed on the conversation tweets according to Section 4.2. In CNN, the final set of their optimized parameters were used. Thus, embedded dimension was set to 400, Max length of sequences to 350, the number of filters in Conv1D layer to 100, window size to 5, number of the dense layer to 4, number of units in the dense layer to 100, activation function to tanh, Pooling layer to GlobalMaxPooling, Epochs to 10, and Optimizer to Adagrad [48].In GRU-RNN, the layer of the GRU network was set to 2, the size of GRU unit to 100, and the embedding size to 100. All neural network based models were implemented using keras.

The results of experiments are shown in Table 3. The bold values in Table 3 denote the highest values in the rumor and non-rumor classes. As can be observed, the precision, recall and F1 of the rumor class performed better in the proposed model. In the non-rumor class, the precision and F1 of the proposed method were more efficient and the recall was lower. GRU-RNN and CNN methods could both learn deep latent features from data. In Table 3, the superior performance of CNN to detect rumors compared to GRU-RNN was as a result of the correct setting of hyperparameters in this network and the use of dense layers. The recall of the non rumor class in CNN was better than other methods. However, the GAN-GRU method displayed greater accuracy than the CNN method. According to the analysis, it can be concluded that detection performance is significantly improved by modeling the user graphs and reply tree on GCN.

[2] https://github.com/majingCUHK/Rumor_GAN.

Table 2 Comparison results of Micro-averaged scores F1, recall, and precision of the rumor and non-rumor classes across all five runs of baseline models and the proposed model

Method	Rumor			Non-Rumor		
	Presicion	recall	F1	presicion	recall	F1
RF	0.39	0.26	0.32	0.67	0.79	0.72
DT	0.39	0.42	0.41	0.68	0.66	0.67
CRF	0.66	0.55	0.60	0.79	0.86	0.82
User-Reply-GCN	0.63	0.64	0.63	0.81	0.82	0.82

Moreover, in this research, performance of the Reply tree module and that of the user module were individually evaluated on the k-fold data of the GAN-GRU method using 5-fold cross-validation. Investigating the results of precision, recall and F1 of the rumor and non-rumor classes of these two modules in Table 3, it was found that the reply tree had a greater efficiency compared to the user graph in the rumor class and the non-rumor class, indicating that the content of tweets in the conversations and their propagation structure play a more effective role in rumor detection compared to the user features and how they interact. However, after combining these two modules, the performance of the proposed model increases.

By comparing the performance of the proposed model in Tables 2 and 3, it was found that the precision, recall and F1 values in Table 3 were higher than Table 2. One reason for this is that the number of reply tweets in the conversation of Table 3 provides more information while in Table 2, the number of reply tweets in many of the conversations is either lower than 10 or without a reply tweet. Results of the comparison indicated that the efficiency of the proposed model in the rumor class is much better than that of baseline methods.

Table 3 The precision, recall, F1 in detecting rumors and non-rumor conversations of the proposed model and other models

Method	Rumor			Non-Rumor		
	Precision	recall	F1	precision	recall	F1
GAN-GRU[30]	0.77	0.80	0.78	0.79	0.77	0.78
GRU-RNN	0.71	0.70	0.70	0.71	0.72	0.71
CNN	0.77	0.75	0.76	0.76	0.79	0.77
Reply tree	0.75	0.80	o.77	0.79	0.74	0.77
User graph	0.56	0.55	0.55	0.56	0.57	0.56
User-Reply-GCN	0.77	0.82	0.80	0.82	0.76	0.79

Table 4 Comparison results of different features on the feature matrix of reply tree

Method	Non-Rumor			Rumor		
	Precision	recall	F1	precision	recall	F1
Emotion	0.63	0.61	0.62	0.61	0.63	0.62
Reply tweets	0.77	0.79	0.78	0.78	0.76	0.77
Linguistic	0.60	0.56	0.58	0.57	0.61	0.59

5.4 Further experiments

To further assess the model performance, the feature matrix of the reply tree was evaluated using three different methods. In the first method, ten groups of basic emotions were used as features while in the second and third methods, high frequency words of reply tweets and linguistic features were determined as the features respectively. Finally, the proposed method was examined for early detection of rumors. All experiments were evaluated on the k-fold data of the GAN-GRU method with 5-fold cross-validation.

Emotions features The purpose of analyzing emotions is to find opinions and identify the emotions that users express in different situations. Rumors in different societies are evoked by certain emotions, and people spread rumors in groups to attract attention, express their anger, fatigue, or positive feelings [1]. Therefore, the conversation tweets were studied to find their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) [12]. To extract the emotions of conversation tweets in this paper, the National Research Council Canada (NRC) Emotion Lexicon was used. NRC Emotion Lexicon is a list of English words along with their associations with eight basic emotions and two sentiments [49]. Therefore, in the feature matrix of the reply tree, 10 features were considered for each node.

Features of high frequency words In this section, the high frequency words of the reply tweets included the top-800 words extracted from the training file as the features were considered in the feature matrix of the reply tree. All of these high frequency words were indicated by a binary feature.

Linguistic features Linguistic features are extracted from the characteristics of the content of conversation tweets. To investigate the effect of linguistic features on the detection of rumor conversations, linguistic features were defined in two of the dependent and independent features of Twitter for the feature matrix of the reply tree. These features include Twitter-dependent and Twitter-independent features. Twitter-dependent features comprise of hashtags, mentions, and urls and Twitter-independent features consist of question marks, exclamation marks, abbreviations, average word complexity, tweet length, happy and sad emoji, and capital letters on tweets, first person pronouns, second person and third person in the tweets [10]. A total of 16 features were defined in the feature matrix of the reply tree.

Results of the experiments of precision, recall and F1 of the rumor and non-rumor classes of these three methods are presented in Table 4. The bold values in Table 4 show the greatest efficiency. The use of high frequency words of reply tweets in the feature matrix had a higher efficiency in rumor and non-rumor classes compared to the methods using Emotions and Linguistic Features. A comparison of the results in Table 4 with the results of the proposed model in Table 3 demonstrates that the use of high frequency words of the source tweet and propagation time has a greater efficiency in the model performance, indicating the importance of the source tweet in detection of rumors.

Early rumor detection One of the most important metrics for evaluating the effectiveness of the proposed method is early detection of rumors, which helps prevent the spread of rumors in the early stages. To construct an early detection task, a series of detection deadlines were established and only the reply tweets

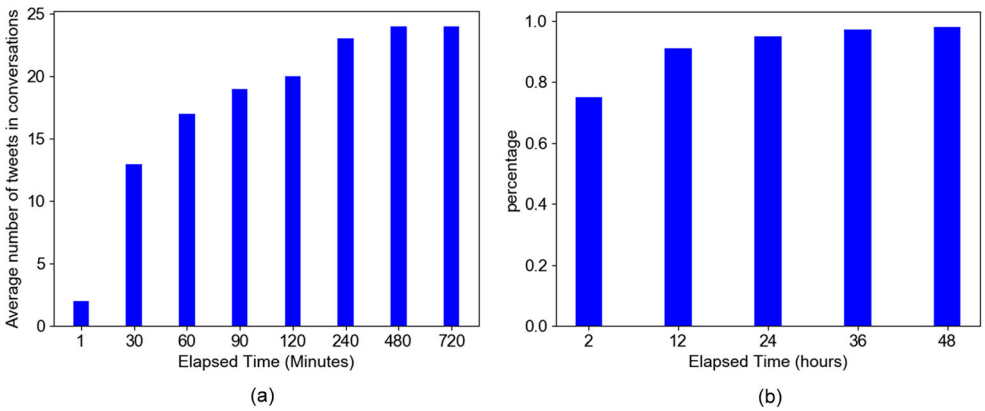

Fig. 5 The statistics of all tweet sets in conversations in the testing set. **a** The statistic about the average number of tweets in conversations, **b** The statistic about tweets count

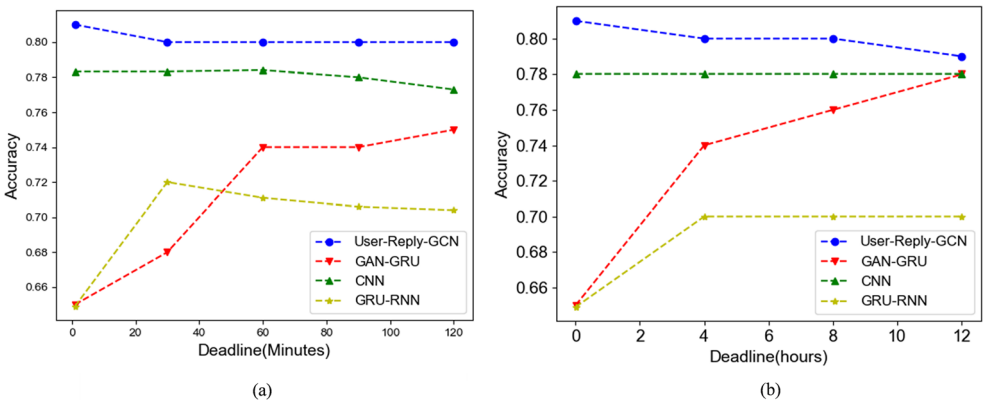

Fig. 6 Results of rumor early detection with different methods. **a** Elapsed time (from 1m to 120m), **b** Elapsed time (from 0.01h to 12h)

before the deadlines were used to evaluate the accuracy of the proposed method and other methods. Figure 5 shows the statistics of all tweet sets in conversations in the testing set. The X-axis in Fig. 5a represents the elapsed time and the Y-axis represents the average number of tweets in conversations. As shown in Fig. 5a, conversations had an average of 13 tweets in 30 minutes from the start conversation. The X-axis in Fig. 5b represents the elapsed time and the Y-axis represents the percentage. As demonstrated in Fig. 5b, approximately 75% of the tweet sets had a time span of less than 2 h and approximately 92% of the tweet sets had a time span of under 12 h, indicating the need for early detection of rumors in the early hours of propagation.

The performances of the proposed mehod for early rumor detection were evaluated. According to the Fig. 5, the elapsed time from 1 m to 120 m and from 0.01 h to 12 h were considered. In early detection task, the training file contains all the tweets, but in the test set only the reply tweets before the deadlines were used to evaluate the accuracy of the proposed method and other methods. Only neural network based models were evaluated since the performances of the neural network was far superior to the classification approaches which are based on traditional machine learning in rumor detection tasks. Figures 6a and b show the performances of the proposed mehod of comparative neural network methods at different checkpoints in terms of elapsed time according to minutes and hour. It is lucid that the proposed method reaches relatively high accuracy at a very early period after the source post initial broadcast. As illustrated in Fig. 6a, the performance of each model improves rapidly within 1 m after the source tweet is posted and in all models over time, the performance of the models becomes more stable. Experimental results show the effectiveness of the structural features for the early detection of rumors.

6 Conclusion

In this paper, a GCN-based method to detect rumor conversations in Twitter events was proposed. This method covers effective features in detection of rumors including the aspects of propagation, content, propagation time, users, and user behaviors. The proposed model consists of three modules: user, reply tree and concatenate. In the user and reply tree modules, useful information regarding the propagation structure and user behaviors was obtained and then combined in the concatenate module to detect the rumors. Results of the experiments compared to the baseline work indicated that the proposed method has a higher efficiency in detecting rumors and the proposed model also has the capability of detecting early detect rumors. In future research, the attention mechanism in GCN can be used to improve model efficiency in detecting rumors.

References

1. Java A, Song X, Finin T, Tseng B (2007) Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pp 56–65
2. Starbird K, Palen L, Hughes AL, Vieweg S (2010) Chatter on the red: what hazards threat reveals about the social life of microblogged information. In: Proceedings of the 2010 ACM conference on Computer supported cooperative work, pp 241–250
3. Cai G, Wu H, Lv R (2014) Rumors detection in chinese via crowd responses. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). IEEE, Piscataway, pp 912–917
4. Zubiaga A, Aker A, Bontcheva K, Liakata M, Procter R (2018) Detection and resolution of rumours in social media: A survey. ACM Comput Surv 51(2):1–36
5. Ritter A, Cherry C, Dolan B (2010) Unsupervised modeling of twitter conversations. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, pp 172–180
6. Bondielli A, Marcelloni F (2019) A survey on fake news and rumour detection techniques. Inf Sci 497:38–55
7. Cao J, Guo J, Li X, Jin Z, Guo H, Li J (2018) Automatic rumor detection on microblogs: A survey. arXiv preprint arXiv: 180703505
8. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907
9. Cogan P, Andrews M, Bradonjic M, Kennedy WS, Sala A, Tucci G (2012) Reconstruction and analysis of twitter conversation graphs.

In: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, pp 25–31
10. Castillo C, Mendoza M, Poblete B (2011) Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web, pp 675–684
11. Qazvinian V, Rosengren E, Radev DR, Mei Q (2011) Rumor has it: Identifying misinformation in microblogs. In: Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, pp 1589–1599
12. Kwon S, Cha M, Jung K, Chen W, Wang Y (2013) Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th International Conference on Data Mining. IEEE, Piscataway, pp 1103–1108
13. Zhao Z, Resnick P, Mei Q (2015) Enquiring minds: Early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th international conference on world wide web, pp 1395–1405
14. Zubiaga A, Liakata M, Procter R (2016) Learning reporting dynamics during breaking news for rumour detection in social media. arXiv preprint arXiv:161007363
15. Vosoughi S, Roy D (2015) A human-machine collaborative system for identifying rumors on twitter. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, Piscataway, pp 47–50
16. Vosoughi S, Mohsenvand MN, Roy D (2017) Rumor gauge: Predicting the veracity of rumors on Twitter. ACM Trans Knowl Discov Data (TKDD) 11(4):1–36
17. Jahanbakhsh-Nagadeh Z, Feizi-Derakhshi M-R, Sharifi A (2020) A speech act classifier for persian texts and its application in identifying rumors. J Soft Comput Inf Technol (JSCIT) 9(1)
18. Giasemidis G, Singleton C, Agrafiotis I, Nurse JR, Pilgrim A, Willis C, Greetham DV (2016) Determining the veracity of rumours on Twitter. In: International Conference on Social Informatics. Springer, Berlin, pp 185–205
19. Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. IEEE Comput Intell Mag 13(3):55–75
20. Zhao L, Zhou Y, Lu H, Fujita H (2019) Parallel computing method of deep belief networks and its application to traffic flow prediction. Knowl Based Syst 163:972–987
21. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems. IEEE, Piscataway, pp 253–256
22. Jiang H, Jin W (2019) Effective use of convolutional neural networks and diverse deep supervision for better crowd counting. Appl Intell 49(7):2415–2433
23. Lin C-H, Wang S-H, Lin C-J (2019) Using convolutional neural networks for character verification on integrated circuit components of printed circuit boards. Appl Intell 49(11):4022–4032
24. Sundermeyer M, Schlüter R, Ney HL (2012) STM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association
25. Yogatama D, Dyer C, Ling W, Blunsom P (2017) Generative and discriminative text classification with recurrent neural networks. arXiv preprint arXiv:170301898
26. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp 3818–3824
27. Ajao O, Bhowmik D, Zargari S (2018) Fake news identification on twitter with hybrid cnn and rnn models. In: Proceedings of the 9th International Conference on Social Media and Society, pp 226–230
28. Xu N, Chen G, Mao W (2018) MNRD: A merged neural model for rumor detection in social media. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, Piscataway, pp 1–7
29. Chen T, Li X, Yin H, Zhang J (2018) Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, pp 40–52
30. Ma J, Gao W, Wong K-F (2019) Detect rumors on Twitter by promoting information campaigns with generative adversarial learning. In: The World Wide Web Conference, pp 3049–3055
31. Alsaeedi A, Al-Sarem M (2020) Detecting Rumors on Social Media Based on a CNN Deep Learning Technique. Arab J Sci Eng 1–32
32. Liu Y, Jin X, Shen H (2019) Towards early identification of online rumors based on long short-term memory networks. Inf Process Manag 56(4):1457–1467
33. Santhoshkumar S, Babu LD (2020) Earlier detection of rumors in online social networks using certainty-factor-based convolutional neural networks. Soc Netw Anal Min 10(1):1–17
34. Asghar MZ, Habib A, Habib A, Khan A, Ali R, Khattak A (2019) Exploring deep neural networks for rumor detection. J Ambient Intell Humanized Computing:1–19
35. Ouyang Y, Zeng Y, Gao R, Yu Y, Wang C (2020) Elective future: The influence factor mining of students' graduation development based on hierarchical attention neural network model with graph. Appl Intell. https://doi.org/10.1007/s10489-020-01692-6
36. Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. Comput Soc Netw 6(1):11
37. Zhao J, Liu X, Yan Q, Li B, Shao M, Peng H (2020) Multi-attributed heterogeneous graph convolutional network for bot detection. Inform Sci 537:380–393. https://doi.org/10.1016/j.ins.2020.03.113
38. Levie R, Monti F, Bresson X, Bronstein MM (2018) Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Trans Signal Process 67(1):97–109
39. Kipf TN, Welling M (2016) Variational graph auto-encoders. arXiv preprint arXiv:161107308
40. Vijayan R, Mohler G (2018) Forecasting retweet count during elections using graph convolution neural networks. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, Piscataway, pp 256–262
41. Dong M, Zheng B, Quoc Viet Hung N, Su H, Li G (2019) Multiple rumor source detection with graph convolutional networks. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp 569–578
42. Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J (2018) Hierarchical graph representation learning with differentiable pooling. NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp 4805–4815
43. Zhang M, Cui Z, Neumann M, Chen Y (2018) An end-to-end deep learning architecture for graph classification. In: Thirty-Second AAAI Conference on Artificial Intelligence
44. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2019) A comprehensive survey on graph neural networks. arXiv preprint arXiv:190100596
45. Kwon S, Cha M, Jung K (2017) Rumor detection over varying time windows. PLoS One 12(1):e0168344
46. Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw Mach Learn 4(2):26–31
47. Branco P, Torgo L, Ribeiro R (2015) A survey of predictive modelling under imbalanced distributions. arXiv preprint arXiv:150501658
48. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(7):2121–2159

49. Mohammad SM, Turney PD (2013) Nrc emotion lexicon. National Research Council, Ottawa

Serveh Lotfi received the B.Sc degree in computer engineering from Razi University, Kermanshah, Iran in 2003 and M.Sc. degree in computer engineering from the Arak branch, Islamic azad University, Arak, Iran, in 2008. Currently, she is pursuing her Ph.D. degree in computer engineering in Science and Research Branch, Islamic Azad University, Tehran, Iran. Her current research interests include data mining, machine learning, deep learning, and social network analysis. She is currently a faculty member at Marivan branch, Islamic azad University, Marivan, Iran.

Mitra Mirzarezaee received her B.S.degree in computer engineering from IAU Central Tehran Branch, M.S degree andPh.D. degree both in computer engineering from IAU science and research branch,in 2007 and 2012 respectively. She is currently a faculty member at Science andResearch Branch, Islamic Azad University, Tehran, Iran. Her current researchinterests include pattern recognition, machine learning, deep learning, andsocial network analysis.

Mehdi HosseinZadeh received his B.S.degree in computer hardware engineering, from Islamic Azad University, Dezfolbranch, Iran in 2003. He also received his M.Sc. and the Ph.D. degree in computersystem architecture from the Science and Research Branch, Islamic AzadUniversity, Tehran, Iran in 2005 and 2008, respectively. He is currently anAssociate professor in Iran University of Medical Sciences (IUMS), Tehran,Iran. He is the author/co-author of more than 120 publications in technicaljournals and conferences, and hisresearch interests include SDN, Information Technology, Data Mining, Big data analytics, E-Commerce, E-Marketing, and Social Networks.

Vahid Seydi is currently anAssistant Professor at the Department of AI at Azad University South TehranBranch. He received a B.Sc. (2005) in software engineering, Ms.Sc. (2007) andPhD (2014) in AI, from the Department ofComputer Science at Azad University, Science and Research Branch, TehranIran. He has been awarded a merit-based scholarship for attending in the schoolof AI, Rome, Italy (2019), Also, He has achieved full scholarship Award fromAzad University (2010-2014), KNTU ISLAB Research Fellowship (2007-2010). Hesecured the first rank among the graduates in the year 2004-2005. His currentresearch interests include domain adaptation, adversarial learning andmeta-learning.