Social Network Analysis and Mining (2024) 14:100
https://doi.org/10.1007/s13278-024-01263-4

ORIGINAL ARTICLE

Lgt: long-range graph transformer for early rumor detection

Jinghong Xia[1] · Yuling Li[1] · Kui Yu[1]

Received: 27 December 2023 / Revised: 22 April 2024 / Accepted: 23 April 2024

Abstract
Society is now situated in an epoch where the creation and spread of fake news have become remarkably effortless. Hence, conducting early rumor detection tasks is imperative. To handle this task, a key ideal is to model the interactive information between users who spread the news. To this end, existing methods usually use multiple stacked GNN layers to capture long-range user information. However, recent work has shown that traditional GNNs may struggle to capture important information when dealing with k-hop neighbors of users, thus hurting the performance of models. To address this problem, we propose a Long-range Graph Transformer for early rumor detection (LGT), which uses transformers to capture long-range dependencies between users. First, we use a graph convolutional attentive network to extract the publishing features. Second, we combine graph neural network and transformer to capture the long-range interaction features of users. Then, we employ the convolutional neural network to extract the text features and use the attention mechanism to fuse with the interactive information to obtain the aggregated interaction features. In addition, we collect the user's credibility score as additional information. Finally, the above three features are fused to generate a new representation. Extensive experiments using three authentic datasets demonstrate that, in comparison to the baseline, LGT has achieved significant improvement. It effectively identifies rumors quickly while maintaining an accuracy rate exceeding 94%.

Keywords Graph convolution · Attention mechanism · Long range · Graph transformer · Early rumor detection

1 Introduction

The surge in fake news on social media has led to panic and confusion spreading throughout society. Rumors can quickly spread and mislead public opinion due to the widespread use of social media. However, due to the insufficient personal knowledge, ordinary people cannot accurately verify the authenticity of each news in a short period. Therefore, it is necessary to develop advanced tools and technologies for early rumor detection to minimize the negative impact caused by false news.

Early rumor detection (ERD) aims to promptly identify rumors by capturing features such as news texts, images, attributes of participating users, and propagation patterns. Existing ERD models can be divided into two main categories: news content-based models (Hu et al. 2021; Przybyla 2020; Yu et al. 2017), and social context-based models (Shu et al. 2017; Jin et al. 2016; Giachanou et al. 2019). The news content-based models typically utilize the news content features such as the specific emotions expressed by the news (Giachanou et al. 2019), the writing styles (Przybyla 2020), and the overly exaggerated headlines or images to detect early rumors. Given that substituting sentiment words, imitating the writing styles of real news, and fabricating headlines or images of fake news based on the real news topics are relatively easy, existing news content-based models are difficult to identify well-designed fake news. Therefore, recent studies focus on the social context-based models, which are more robust because deceiving such models requires creating fake user accounts or constructing social networks with structures similar to real news dissemination.

Existing social context-based models focus on learning the contextual information representations of social events by modeling the source news and relevant user behaviors. For example, Ma et al. (2016) employed recurrent neural

✉ Jinghong Xia
2022171259@mail.hfut.edu.cn

Yuling Li
lyl95@mail.hfut.edu.cn

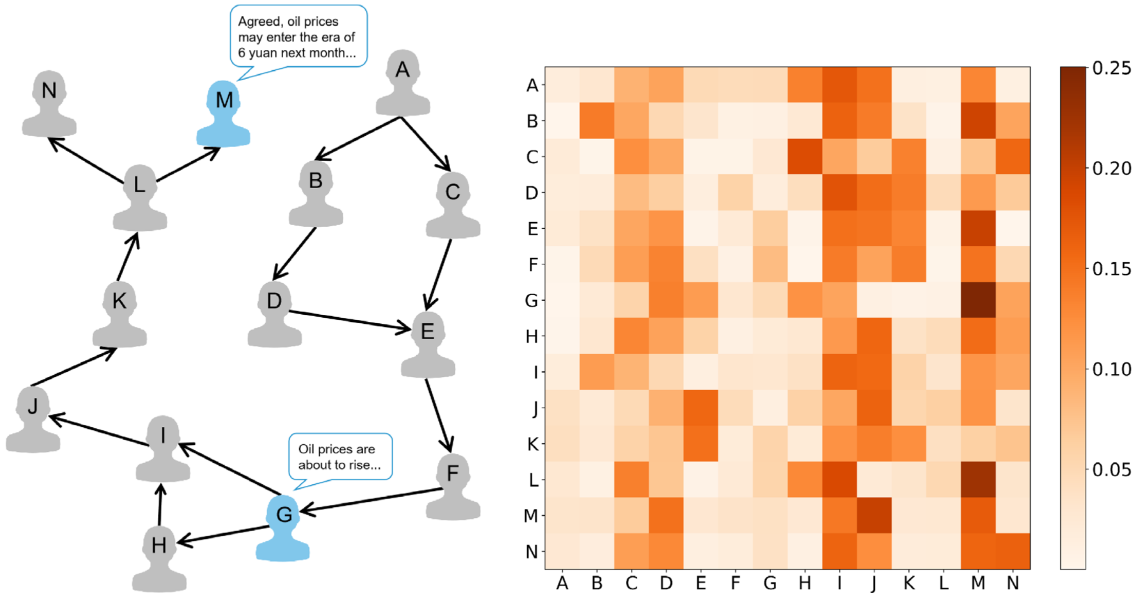
Kui Yu
yukui@hfut.edu.cn

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China

Published online: 13 May 2024

networks to model source posts and relevant comments. Although (Ma et al. 2016) is relatively effective, it ignores the impact of rumor propagation. To address this issue, Liu and Wu (2018) used CNN-based methods to obtain information from the local structure of rumor propagation. Subsequently, considering the need to capture global structural information on the graph, Bian et al. (2020) proposed a bidirectional graph convolutional network to learn the propagation pattern and capture the diffusion structure of rumors. Yuan et al. (2019) modeled the global relationships between all source tweets, retweets, and users as a heterogeneous graph to capture rich structural information. Despite graph structure models success, they have the limitation that they employ multiple stacked GNN layers to aggregate the information of k-hop neighboring users into the source user, which is not sufficient to capture long-range dependencies between users.

Recent work has shown that traditional GNNs may struggle to capture important information when dealing with k-hop neighbors of users (Xu et al. 2018), and GNN performance significantly decreases with increasing depths of neighbors (Li et al. 2018). Chen and Wong (2020) identified the issue of ineffective long-range dependency capture in GNN-based sequence recommendation methods, suggesting that the limited number of GNN layers fails to capture the long-range dependency relationships between items in sessions. Alon and Yahav (2020) pointed out a bottleneck in GNNs when aggregating information from distant nodes, where with the increase in layers, the number of k-hop neighbors of a node grows exponentially, but more information is compressed into a fixed-length vector compared to the previous layers. In cases where the shortest path exceeds the number of GNN layers, the information from distant nodes cannot be effectively transmitted. Li et al. (2021) proposed the challenges faced by training deep GNNs, namely, the disappearance of gradients leads to almost no updates of network parameters, making it difficult for the network to converge, and the explosion of gradients leads to excessive parameter updates, causing the network to lose stability. Therefore, only using k-hop neighbors to enhance the semantic representations of user is not sufficient to capture long-range dependencies between users, thereby hurting the performance of early rumor detection methods. Figure 1a shows the relationship graph of some user nodes randomly selected from the Twitter15 dataset, where each node represents a user. Figure 1b shows the attention map of how neighbor nodes of user nodes contribute to the semantic information when representing the target user node. In Fig. 1a, we can observe that user G posted a viewpoint about the imminent rise in oil prices, while user M expressed agreement in the repost and triggered many followers to repost. This behavior is reflected in Fig. 1b as a higher score of node G's attention to node M. Therefore, even as a 5-hop neighbor of node G, the semantic information of user node M is crucial for the representation of the target user node G. To achieve this, existing ERD methods typically train stacked 5-layer GNNs.



(a) Sample graph from Twitter15

(b) Attention map from graph transformer

Fig. 1 Sample graph and attention map in our graph transformer, randomly selected from Twitter15 validation set. The attention map is obtained from the transformer module in our graph transformer. A~N represent user nodes on the propagation path. The horizontal axis represents the target user nodes and the vertical axis represents the source user nodes

Such practice may lead to an exponential growth in receptive field width and sparse signal representation, making it difficult to capture long-range dependencies (Wu et al. 2021). Hence, it is necessary to design a new technique to aggregate the information of neighboring users.

To solve the above problems, we propose a novel model, namely Long-range Graph Transformer (LGT), for early rumor detection. The proposed LGT consists of two modules: a bot detection module and a rumor detection module. The bot detection module aims to learn the possibility of a user being human by encoding the attributes information of the user. The rumor detection module aims to model a broader range of neighbors to capture long-range dependencies between users and learn various features that determine the news being a rumor. In the rumor detection module, we first present a graph convolutional attentive network, which combines the advantages of graph convolution for modeling graph-structured data and the attention mechanism to dynamically weight and aggregate information to obtain the correlation between publishers and news. Second, we design the long-range graph transformer to capture the user's interaction information from the news propagation. Finally, we employ a convolutional neural network (CNN) to extract the text information of the news and introduce an attention mechanism to fuse the extracted news information with the interaction information. The experimental results show that the proposed LGT is beneficial for real-time identification and prevention of rumor propagation.

The main contributions of this paper can be summarized as follows:

- We propose a novel LGT model that takes user credibility as additional information to capture the graph structure information of news spread through the graph convolutional attentive network and graph transformer. Different from existing ERD methods that utilize multiple stacked GNNs resulting in sparse signal representation, the proposed LGT model can utilize transformers to capture the information of distant users, which reduces the loss of long-range information.
- We conducted a series of experiments on three real-world datasets. The experimental results demonstrate that our model achieves remarkable improvements in rumor classification and early prediction tasks compared to state-of-the-art models.

2 Related work

The task of fake news detection involves evaluating the authenticity of news circulated on social media platforms. This is done by analyzing various factors, including news content, user behavior, and propagation patterns, to provide a more reliable information environment for the public. Existing approaches can be divided into two categories (Shu et al. 2017): (1) news content-based methods; and (2) social context-based methods.

2.1 News content-based methods

The news content-based methods can be further categorized into two main subcategories: knowledge-based, as well as style-based. The knowledge-based method initially requires a fact-checking database to rectify the opinions and objective things described in news articles, which involves tasks like knowledge representation and knowledge reasoning. Then the knowledge base or knowledge graph is utilized to judge the authenticity of the new news content. Hu et al. (2021) focused on knowledge-based fake news detection by utilizing external knowledge sources. The style-based approach utilizes the writing style inherent to the news content itself. It captures sentence grammatical information by employing context-independent grammar rules or rhetorical structure theory (RST) (Mann et al. 1987) dependencies to extract the sentence's syntactic structure and other grammatical details. Przybyla (2020) explored stylistic features for fake news detection. Yu et al. (2017) presented a convolutional approach to identifying misinformation, which includes analyzing linguistic features. dEFEND (Shu et al. 2019) employed textual features and interpretable models for fake news detection, focusing on explaining its decisions. DTCA (Wu et al. 2020) utilized textual content and attention mechanisms to verify claims, emphasizing explainability through decision tree integration.

However, knowledge-based methods often face challenges related to incomplete or outdated information in knowledge bases, limiting their effectiveness in detecting emerging or context-specific rumors. On the other hand, when rumors imitate the writing style of trusted sources, style-based methods may struggle to accurately distinguish rumors from legitimate content, leading to potential false positives. Furthermore, they may not effectively capture the semantics of the text, making them vulnerable to context changes and changing rumor styles.

2.2 Social context-based methods

The social context-based methods can be divided into two types: stance-based and propagation-based. The former is mainly based on user operations on content (such as comments, likes, reports) to build a matrix or graph model. et al. Jin et al. (2016) explored the verification of news by considering conflicting microblog viewpoints. Giachanou et al. (2019) leveraged emotional signals in their work on credibility detection, which is closely related to capturing user stance. Castillo et al. (2011) investigated information credibility on Twitter, which involves understanding how users perceive and evaluate information. The method based on propagation behavior models the object and tracks the trajectory of the news. Zhou and Zafarani (2019) explored network-based fake news detection, which involves studying the patterns of how fake news spreads in a network. Bian et al. (2020) investigated rumor detection focusing on bi-directional graph convolutional networks, which inherently consider the propagation behavior. Song et al. (2021) designed a temporally evolving graph neural network to capture the evolving nature of fake news propagation. Sun et al. (2022) used a hyperedge learning method to represent the temporal propagation structure and a fusion neural network to jointly learn the content, structural, and temporal features of rumor propagation. Liu et al. (2022) proposed a novel rumor detection framework based on structure-aware retweeting graph neural network. Meng et al. (2023) constructed a global heterogeneous transition graph to integrate user-news relationships and overall user historical click news sequences.

In the research on early rumor detection, Liu and Wu (2018) used recurrent and convolutional neural networks to detect fake news by analyzing its propagation patterns on social media. Chen et al. (2018) focused on modeling rumor propagation behaviors using deep attention-based recurrent neural networks. Yuan et al. (2019) proposed a method that jointly embeds local and global relations in a heterogeneous graph to enhance rumor detection by considering various aspects of rumor propagation behavior. Xia et al. (2020) introduced a network model that considers the evolving nature of rumors and their propagation on social media for early detection. Yuan et al. (2020) proposed a novel structure-aware multi-head attention network (SMAN) that combines news content, publishing, and reposting relationships to jointly optimize fake news detection and credibility prediction tasks. Subsequently, Huang et al. (2022) proposed a social bot-aware graph neural network called SBAG. The model pre-trains multi-layer perception networks to obtain features of social bots, and then constructs multiple graph neural networks by embedding features to model the early propagation of posts, further used for detecting rumors. Note that SBAG is considered one of the state-of-the-art models in the current field.

The modeling of graph structures is beneficial for capturing local and global features of rumor spreading. However, these methods all use stacking multiple GNN layers to aggregate the information of k-hop neighbors into the source. The information of neighbors gradually becomes blurred as their depth increases, making the model inefficient in capturing interactions that occur over longer distances. In social networks, user interaction often involves more complex interaction paths, and these methods may be difficult to model and understand this complexity. Therefore, we propose the LGT model to address the limitations of these models. Specifically, we design a long-range graph transformer that uses traditional GNN subnetworks as the backbone, but leaves long-range dependent learning to transformer subnetworks. Our transformer application focuses each node on other nodes, motivating the transformer to learn the most important node-node relationships, instead of favoring nearby nodes (the latter task has been offloaded to the previous GNN module).

3 Problem formulation

Let $\mathcal{B} = \left\{b_1, b_2, ..., b_{|B|}\right\}$ denote a set of users consisting of both bots and real users, $\mathcal{N} = \left\{n_1, n_2, ..., n_{|N|}\right\}$ denote a set of news, and $\mathcal{U} = \left\{u_1, u_2, ..., u_{|U|}\right\}$ denote a set of users participating in the propagation of news. Among users, there are further distinctions between publishers and retweeters. We first use dataset $\mathcal{B}$ to pre-train the bot detection model, allowing the model to learn features and representations about bot behavior. After pre-training, we use dataset $\mathcal{U}$ as input for the bot detection model to evaluate the user's credibility score. This score reflects the probability that the user is identified as a real user. Next, we transfer this credibility score to the rumor detection module as auxiliary information, helping the rumor detection model more accurately identify and eliminate the influence of bot users. The user publishing process can be represented as $G_P = \langle V_P, V_N, E_P \rangle$, where G_P represents the publisher-news relationship graph, V_P is the set of all publishers, V_N is the set of all news, E_P is the set of edges, and an edge $(u_i, n_j) \in E_P$ indicates that user u_i publishes news n_j. Similarly, the user interaction process can be represented as $G_I = \langle V_I, E_I \rangle$, where G_I represents the user-user relationship graph, V_I is the set of all users, E_I is the set of edges, and an edge $(u_i, u_j) \in E_I$ indicates that user u_i replies to user u_j. We use Graph Convolutional Network (GCN) to process the publisher-news relationship graph G_P to obtain the publisher's representation, and use Graph Transformer to process the user user relationship graph G_I to obtain the user interaction representation. Finally, the publisher representation, user interaction representation, and news text representation are concatenated to form a comprehensive representation vector for the final rumor detection.

In this paper, in order to better distinguish real information communicators from false information communicators, we design a bot detection module to score users and utilize user or publisher credibility information for fake news detection. For the bot detection task, our goal is to learn a function $p(c_1|u_i, \mathcal{B};\theta_1)$ to predict the credit score of the user u_i. For the fake news detection task, our goal is to learn a function $p(c_2|n_j, \mathcal{N}, \mathcal{U};\theta_2)$ to predict whether the news n_j is a rumor, where c_1 and c_2 are the class labels of the users and news respectively, and θ_1 and θ_2 represent all the model parameters.

4 The proposed LGT algorithm

In this section, we propose the LGT algorithm for early fake news detection and its framework is shown in Fig. 2. The LGT algorithm has two main components: a bot detection module and a rumor detection module. The bot detection module aims to learn the user credibility score, which represents the probability that the user is human. We use a position feedforward network (FFN) to encode the attribute information of robots and humans, and the obtained user credibility score is used as auxiliary information to pass into the rumor detection module. The rumor detection module aims to determine whether news is a rumor. We modeled a news dissemination graph and used GCN to extract news release features, GraphTransformer to extract user interaction features, and CNN and pooling layers to extract text features. Next, we will introduce the LGT algorithm in detail.

4.1 Bot detection module

Like spammer detection (Sun et al. 2021; Liu et al. 2020), rumor detection also involves identifying and dealing with bad user behavior in online social networks, especially when it comes to dealing with anonymous users and disinformation. The behavior of users depends not only on their personal preferences, but also on the social influence of their direct or indirect social friends (Sun et al. 2023a). Rating users is crucial to help the system identify users who may disrupt or mislead other users by spreading false information (Sun et al. 2023a, b). Therefore, we added a bot detection module to better distinguish the difference between real information communicators and false information communicators by rating users, thereby improving the accuracy and efficiency of rumor detection.

In order to incorporate bot behavior information for fake new detection, we first pre-train the model on a large sample of bots and human beings to encode the user attribute information. The architecture of this module is illustrated in Fig. 2a.

To compute the probability that a user is a human being, we employ a position-wise feed-forward network (FFN) to encode the user features. Specifically, given a user feature vector $c \in \mathbb{R}^v$ containing diverse user profiles, e.g., username length, follower counts, and friend counts. The credibility score $\hat{Y}_u$ can be computed as follows:

$$c' = ReLU(W_1^T c + b_c) \tag{1}$$

$$\tilde{c} = LayerNorm(W_2^T c' + c) \tag{2}$$

$$\hat{Y}_u = softmax(W_u^T \tilde{c} + b_u) \tag{3}$$

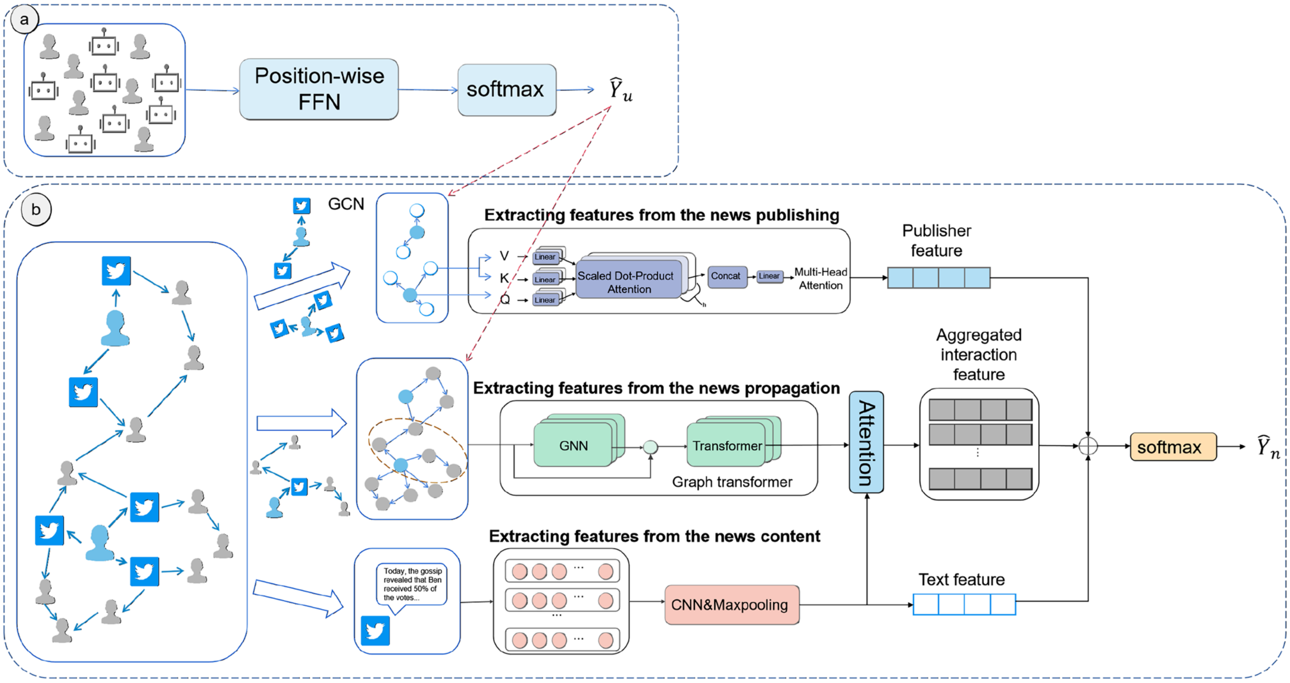

a
Position-wise FFN
softmax
$\hat{Y}_u$
b
GCN
Extracting features from the news publishing
V
K
Q
Linear
Linear
Linear
Scaled Dot-Product Attention
h
Concat
Linear
Multi-Head Attention
Publisher feature
Extracting features from the news propagation
GNN
Transformer
Graph transformer
Attention
Aggregated interaction feature
softmax
$\hat{Y}_n$
Extracting features from the news content
Today, the gossip revealed that Ben received 50% of the votes...
CNN&Maxpooling
Text feature

Fig. 2 Overview of LGT. **a** bot detection module. **b** Rumor detection module

where $W_1, W_2 \in \mathbb{R}^{v\times v}$, $W_u \in \mathbb{R}^{v\times 2}$, $b_c \in \mathbb{R}^v$ and $b_u \in \mathbb{R}^2$ are the parameters of the FFN, $\hat{Y}_u \in \mathbb{R}^2$ is the predicted probability distribution of the user class.

The bot detection module calculates the user credibility score in the range [0,1] and transfers it to the rumor detection module as auxiliary information.

4.2 Rumor detection module

In the rumor detection module, we seek to capture different types of news features by modeling the propagation graph of news in social networks. The rumor detection module mainly consists of three steps: (1) extracting features from the news publishing, (2) extracting features from the news propagation, and (3) extracting features from the news content. In our method, since the user-news publishing graph has a maximum of one hop, we used a simple GCN, while the user-user interaction graph has complex node features and interaction relationships. Therefore, we chose to use a graph transformer to capture long-range dependencies and learn the complex relationships between nodes in the graph. Figure 2b shows the architecture of the rumor detection module. Next, we will introduce each component in detail.

4.2.1 Extracting features from the news publishing

In the news publishing, we aim to capture the features of the users who publish news by modeling the publisher-news graph. Graph neural networks such as GCN and Graph Attention Network (GAT) have been proposed to extract important information from graphs, and have been applied in many fields and have made great progress (Li et al. 2018; Bian et al. 2020). We describe the relationship between publisher-news pairs as graph-structured data, where the central node is the publisher node and the neighbor nodes are all news nodes. When aggregating information, only edge relationships between publisher and news are handled (i.e., the publisher has published a certain piece of news). According to our definition, the publisher-news graph is a heterogeneous graph with at most one hop and only one edge relationship, exhibiting good local homogeneity. Therefore we believe that GCN is sufficient to extract effective features from publisher-news graphs. Different from the recent work that use multi-head attention to learn the node representation from the publishing graph (Yuan et al. 2019, 2020), we use graph convolutional attentive network to capture the structural information of news publishing. Since the publishers of news has a certain degree of commonality, the publishers who frequently publish fake news are more likely to publish rumors. In order to focus on the publishers that are likely to publish rumors, we combine GCN and multi-head attention to model the correlation between news nodes and publisher nodes in the publisher-news graph, and perform the differentiated information aggregation on news nodes to generate new node representations.

Formally, let $P \in \mathbb{R}^{|U|\times d}$ denote the initial embedding of the user nodes, $N \in \mathbb{R}^{|N|\times d}$ denote the initial embedding of the news nodes, the user nodes and adjacent news nodes form an adjacency matrix $A \in \mathbb{R}^{|U|\times|N|}$. In order to capture the impact of bot behavior in the news publishing process, we consider the credibility scores of users as biases. The formula for computing the aggregated feature N' is as follows:

$$N' = \sigma(\hat{A} \cdot N \cdot W + \hat{s}) \tag{4}$$

where $\hat{A} = \tilde{D}^{-(1/2)}\tilde{A}\tilde{D}^{-(1/2)}$ is the regularized adjacency matrix, $\tilde{A} = A + I$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ represent the degree of the i-th node, W is the learnable weight matrix, $\hat{s} \in \mathbb{R}^{|U|\times d}$ is the user credibility matrix.

Next, we calculate the attention weight between each user node u and news node n to determine which nodes are more important in information dissemination. Then, the output features of multi-head attention are concatenated to get aggregated node representation $\hat{N}$. Finally, $\hat{N}$ is summed with the initial user node representation to obtain the final publishing feature. The formulas are as follows:

$$\begin{aligned} &Attention(P, N', N')_h \\ &\quad = softmax\left(\frac{(W_u \cdot P)(W_n \cdot N')^T}{\sqrt{d}}\right) \cdot N' \end{aligned} \tag{5}$$

$$\hat{N} = Relu\left(\overset{H}{\underset{h=1}{||}} Attention(P, N', N')_h \cdot W\right) \tag{6}$$

$$\hat{P} = \hat{N} + P \tag{7}$$

where W_u, W_n and W are the learnable transformation matrices, $\hat{P} \in \mathbb{R}^{|U|\times d}$ is the final publishing feature.

4.2.2 Extracting features from the news propagation

In the news propagation, we aim to use the correlations between users to help reveal the authenticity of news. Existing methods use local neighborhood aggregation, which has limitations in handling complex information dissemination among users. For example, users reposting content from others on their social media platforms leads to wider information dissemination, large-scale events or topics trigger collective behavior among users. Traditional methods employ stacked GCN layers, which only consider users' direct neighbors, limiting their ability to handle long-distance information dissemination among users. To address this problem, We design a long-range graph transformer to learn user interaction features from user-user interaction graphs. This approach allows the model to dynamically capture long-range dependencies between users and comprehensively integrate the influence of k-hop neighborhood.

We initialize the embedding of each user node as $U^{(0)} = \left\{u_0^{(0)}, u_1^{(0)}, ..., u_{|V_u|-1}^{(0)}\right\} \in \mathbb{R}^{|V_u| \times d}$, where V_u is the number of retweeters and d is the node embedding dimension. First, in order to capture the relationship information between neighbor users, we use GNN layers to encode the information of user nodes and neighbor nodes. A general GNN layer can be expressed as:

$$u_i^{(l)} = f_l\left(u_i^{(l-1)}, \left\{u_j^{(l-1)} | j \in N_{(i)}\right\}\right), l = 1, 2, ..., L \tag{8}$$

where L is the total number of GNN layers, $N_{(i)}$ is the neighborhood of i, and $f_l(\cdot)$ is some function parameterized by a neural network, such as relu activation function.

Then, in order to capture long-range dependencies between users, we use transformer layers to encode the information of user nodes and all related user nodes. In addition, we incorporate the output from the bot detection module in Sect. 4.1 to capture the impact of bot behavior on news propagation. Specifically, we obtain the credibility score s_i and s_k of user u_i and user u_k, and take their mean value as the edge weight e_{ik}. The formulas are as follows:

$$e_{ik} = \frac{s_i + s_k}{2} \tag{9}$$

$$a_{ik}^{(l)} = \frac{(W_Q^{(l)} u_i^{(l-1)})^T (W_K^{(l)} u_k^{(l-1)} + W_E e_{ik})}{\sqrt{d}} \tag{10}$$

$$\alpha_{ik}^{(l)} = softmax(a_{ik}^{(l)}) \tag{11}$$

$$u_i^{(l)} = \sum_{k \in U} \alpha_{ik}^{(l)} W_V^{(l)} u_k^{(l)} \tag{12}$$

$$\hat{u}_i^{(l)} = \overset{H}{\underset{h=1}{||}} \sigma(u_i^{(l,h)}) \tag{13}$$

where $W_Q^{(l)}, W_K^{(l)}, W_E$ and $W_V^{(l)}$ are learnable parameters, $\alpha_{ik}^{(l)}$ is the attention weight of neighbor node k to target node i at the l-th layer. Finally, the interaction features $U^{(L)} = \left\{\hat{u}_0^{(L)}, \hat{u}_1^{(L)}, ..., \hat{u}_{|V_u|-1}^{(L)}\right\}$ are obtained.

Our graph transformer model uses traditional GNN subnetworks as the backbone to learn nearby node relationships, and leaves learning long-range dependencies to the transformer subnetwork. The transformer application lets each node attend to every other node, which motivates the transformer to learn the most important node-node relationships, thereby reducing the loss of remote information.

4.2.3 Extracting features from the news content

In this section, in order to capture the text features of news, we use CNN and max-pooling layers to encode the source news, which is consistent with the baseline models like SBAG (Huang et al. 2022) and SMAN (Yuan et al. 2020). We represent news i of length L as $X^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, ..., x_L^{(i)}\right\} \in \mathbb{R}^{L \times d}$. Then, the CNN layer (uses d filters with varying receptive field $h \in \{3, 4, 5\}$) and max-pooling layer are applied to the matrix $X^{(i)}$. The formulas are as follows:

$$f_i = ReLU(W \cdot X^i) \tag{14}$$

$$\hat{f}_h = max\left([f_1, f_2, ..., f_{L-h+1}]\right) \tag{15}$$

where $W \in \mathbb{R}^{h \times d}$ is a convolution kernel with size h. Finally, we concatenate the output of each filter $\hat{f}_h$ to form the textual features $\tilde{X} \in \mathbb{R}^{l \times d}$.

4.2.4 Output layer

For a piece of news n, the publishing feature is represented as $\tilde{P}_n \in \mathbb{R}^{1 \times d}$, the interaction feature $\tilde{U} \in \mathbb{R}^{|V_u| \times d}$ is obtained from $U^{(L)}$, and the text feature is $\tilde{X}_n \in \mathbb{R}^{1 \times d}$. To distinguish the importance of different retweeters to the news, we apply an attention mechanism to build the connection between source tweets and retweeters. Specifically, we treat the news $\tilde{X}_n$ as the key information and use it to focus on the retweeters $\tilde{U}$ to calculate attention scores for each retweeter. This score is used to generate aggregated interaction feature $\tilde{U}_n \in \mathbb{R}^{1 \times d}$. The formulas are as follows:

$$s = softmax(\tilde{U} A \tilde{X}_n^T) \tag{16}$$

$$\tilde{U}_n = s^T \tilde{U} \tag{17}$$

where $s \in \mathbb{R}^{|V_u| \times 1}$ is the attention weight vector, and $A \in \mathbb{R}^{d \times d}$ is the trainable matrix.

Finally, we concatenate three types of features, i.e., $\tilde{P}_n, \tilde{U}_n$, and $\tilde{X}_n$, to obtain the final features of the news and calculate the probability of whether the news n is rumor. The probability function is as follows:

$$\hat{Y}_n = softmax\left(W_n^T [\tilde{X}_n || \tilde{P}_n || \tilde{U}_n]^T + b_n\right) \tag{18}$$

where W_n is the transformation matrix, and b_n denotes the bias.

4.3 Training

We use training data with real labels to minimize the cross-entropy loss, optimizing the bot detection task and rumor detection task. The loss functions are as follows:

$$L_b = -\sum_{i=1}^{|B|} Y_{b_i} log \hat{Y}_{b_i} \tag{19}$$

$$L_n = -\sum_{j=1}^{|N|} Y_{n_j} log \hat{Y}_{n_j} \tag{20}$$

where L_b is the cross-entropy loss of the bot detection task, $Y_{b_i} = 1$ means user u_i is human, $Y_{b_i} = 0$ means user u_i is bot, L_n is the cross-entropy loss of the rumor detection task, Y_{n_j} is the ground truth label of news n_j.

4.4 Potential limitations in real-world scenarios

Although we consider multiple factors as much as possible when designing the model, the distribution of data in real-world scenarios is inherently complex and dynamic. For example, the emergence of new types of bots may challenge the adaptability of the bot detection module to promptly address these changes. Malicious users might employ adversarial strategies, deliberately generating deceptive information to evade detection by the model. Moreover, the utilization of user behavior data in bot and rumor detection may raise privacy concerns, necessitating cautious handling of such data in practical applications.

5 Experiment

5.1 Datasets

For the bot detection task, we have chosen 11 datasets from the Bot Repository (botometer.osome.iu.edu/bot-repository). These datasets are divided into training, testing, and validation sets with an 8:1:1 ratio. The statistics of the datasets are shown in Table 1.

For the rumor detection task, we utilize three real datasets: Twitter15 (Ma et al. 2017), Twitter16 (Ma et al. 2017) and Weibo16 (Ma et al. 2016). In the Weibo dataset, authenticity is categorized as either true rumor (TR) or false rumor (FR). In the Twitter dataset, authenticity is classified into four categories: TR, FR, unverified rumor (UR), and non-rumor (NR). The statistics of the three datasets are shown in Table 2.

Table 1 Statistics of the bot detection datasets

Dataset	# Bots	# Human beings
gilani-17	1070	1366
cresci-17	4912	3474
cresci-stock	7102	6174
cresci-rtbust	353	340
celebrity	0	20983
pronbots	16515	0
vendor-purchased	911	0
botwiki	698	0
midterm-2018	12055	8092
verified	0	1951
TwiBot-20	3632	4646
Total	**45693**	**45693**

Table 2 Statistics of the datasets

Statistic	Twitter15	Twitter16	Weibo16
# News	1490	818	4664
# Non-Rumor(NR)	374	205	2351
# False Rumor(FR)	370	205	2313
# Unverified Rumor(UR)	374	203	0
# True Rumor(TR)	372	205	0
# User	276,663	173,487	2,746,818
# Retweet	331,612	204,820	3,805,656

5.2 Experimental settings

For the bot detection module, considering the lack of user features for Twitter15 and Twitter16, we use Twitter API to retrieve user profiles based on user ID. The details are shown in Table 3.

For the rumor detection module, we have implemented and conducted experiments using the PyTorch 1.13 framework. The specific initialization values of the hyperparameters are shown in Table 4.

5.3 Baselines

To evaluate the performance of LGT, we compare LGT with the following methods:

- DTR (Zhao et al. 2015) is a decision tree-based ranking approach, which clusters news by combining news features and then ranks the clustered results.
- DTC (Castillo et al. 2011) is a decision tree model that uses hand-crafted features to detect rumors.
- RFC (Kwon et al. 2017) is a random forest classifier that detects rumors by learning user, linguistic and structural features of news.

Table 3 User characteristics selection

User characteristic	Twitter15	Twitter16	Weibo16
Length of username	√	√	√
Length of screenname	√	√	√
Length of description	√	√	√
Followers count	√	√	√
Friends count	√	√	√
Listed count	√	√	
Favorites count	√	√	√
Statuses count	√	√	√
Days of Registration	√	√	√
URL	√	√	
Protected	√	√	
Geo enabled	√	√	√
Verified	√	√	√
Profile use background image	√	√	
Default profile	√	√	

Table 4 User characteristics selection

Parameters	Values of Twitter	Values of Weibo
Optimizer	Adam	Adam
Batch size	16	128
Learning rate	0.001	0.001
Regression	0.00001	0.00001
Epochs	20	50
Dropout rate	0.4	0.6

- SVM-RBF (Yang et al. 2012) is an SVM model with RBF kernel, which classifies rumors based on statistical features of news.
- SVM-TS (Ma et al. 2015) is a linear SVM model that uses a dynamic series-time structure to capture social context features over time.
- cPTK (Ma et al. 2017) is an SVM model that uses the tree-based kernel to evaluate the similarity of propagation tree structures.
- GRU (Ma et al. 2016) explores the temporal characteristics of these features based on the time series of rumor's life cycle.
- RvNN (Ma et al. 2018) models the spread process of rumors as a tree structure and uses RNN to learn its propagation pattern.
- PPC (Liu and Wu 2018) incorporates recurrent and convolutional networks to capture user characteristics based on time series.
- GLAN (Yuan et al. 2019) proposes a global–local attention network to encode local semantic and global structural information jointly.
- SMAN (Yuan et al. 2020) proposes a structure-aware multi-head attention network to optimize fake news detection and credibility prediction tasks jointly.
- SBAG (Huang et al. 2022) proposes a graph neural network that combines social robot detection and bot-aware graph rumor detection for early rumor detection.

5.4 Experimental result

5.4.1 Analysis of bot detection

For the bot detection module, as mentioned in Table 3, we use 15 user characteristics for Twitter and 10 for Weibo. Hence, we pre-train two bot detection modules: FFN-15d and FFN-10d. For comparison, we consider the following baseline models:

- Botometer-v4 (Sayyadiharikandeh et al. 2020) is a supervised machine learning tool for detecting whether a social media account is a bot.
- MLP (Huang et al. 2022) extracts user features and uses the MLP model to evaluate the user's robot score.

The experimental results are shown in Table 5. FFN-15d and FFN-10d exhibit higher accuracy than baseline models, highlighting their strong user identification capabilities. FFN-15d outperforms FFN-10d due to the richer user information input provided to FFN-15d. The superiority of the FFN models over models like Botometer-V4 and MLP in user identification highlights the advantage of FFN in learning effective user representations. This advantage stems from FFN's ability to capture complex patterns and relationships in the data, adapt to varying feature dimensions, and potentially generalize better to new datasets or user profiles.

5.4.2 Analysis of bot behavior

We present the relationship between rumors and publishers on the test sets of Twitter15, Twitter16, and Weibo16. Specifically, we calculate the ratio of bot-behavior publishers within each source post class. As shown in Fig. 3, we can see that among users who post non-rumor content, the model identifies less than 3% as bot-behavior users. In contrast, bot-behavior users make up nearly half of the total ratio for false rumors. Additionally, users who publish unverified rumors tend to have a high bot ratio, whereas users who share true rumors have a

Table 5 Result of bot detection

Metric	Botomometer-v4	MLP-10d	MLP-15d	FFN-10d	FFN-15d
Acc	0.722	0.849	0.852	0.901	**0.904**

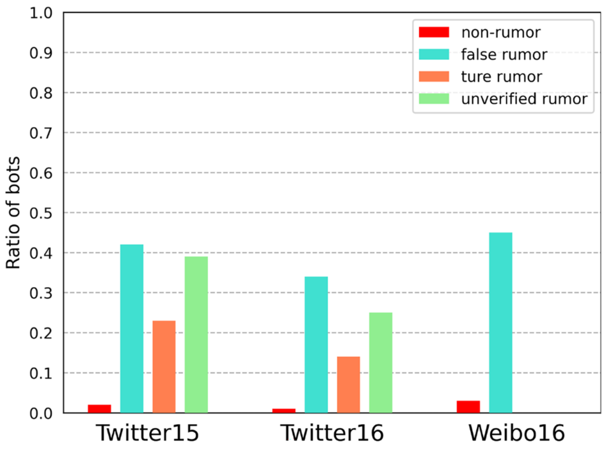


Fig. 3 Relationship between rumors and publishers

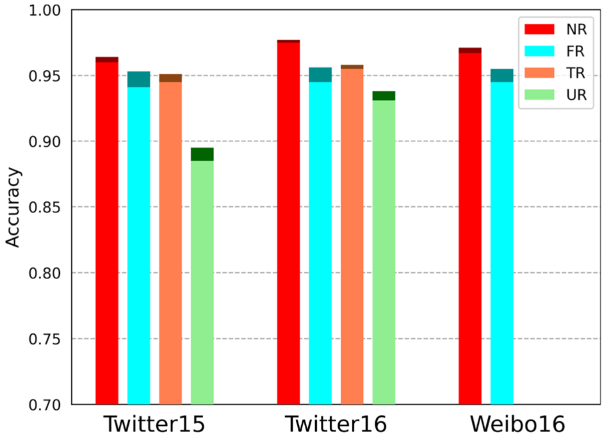


Fig. 4 Add the bot detection module

relatively lower bot ratio. The experimental results show that many bots are created to spread rumors.

We demonstrate the prediction accuracy with the bot detection module in Twitter 15, Twitter 16, and Weibo 16, where dark colors indicate the accuracy after adding the bot detection module. As shown in Fig. 4, we observe that the accuracy of the FR and UR categories has significantly improved, corresponding to the ratio of bot in Fig. 3. Experimental results show that the accuracy of rumor detection has improved after adding the bot detection module.

We also calculate the average ratio of users with bot behavior among all participants for each type of source news. As shown in Fig. 5, bot-behavior users tend to be highly active within 5 min after the source post is published, gradually declining over the following hour. The activity of bots is more evident in false and unverified rumors than in true rumors and non-rumors. The experimental results indicate that users exhibiting bot-behavior are more active right after a post is published. This heightened activity may stem from their design to monitor and swiftly engage with emerging topics or events on social media, thereby ensuring their early involvement in the dissemination of information. Furthermore, the presence of bots is more conspicuous in instances involving false or unverified rumors compared to true rumors and non-rumors. This suggests that bots tend to share negative or controversial information, which might speed up the spread of false news on social media. Over time, the activity levels of these bots gradually diminish, indicating bots are intentionally reducing their participation. This behavior may be attributed to their efforts to avoid detection by the platform or to avoid drawing attention from human users.

5.4.3 Analysis of rumor detection

For ease of comparison, accuracy (Acc.), precision (Prec.), recall (Rec.), and F1-score (F1) are used as indexes for evaluating models. Tables 6, 7 and 8 show the experimental results of LGT and baseline models on three datasets, respectively.

Tables 6 and 7 show the experimental results of the above models on the Twitter15 and Twitter16 datasets. The accuracy of the proposed model is 94.0% and 95.7%, respectively, which is better than other models. Table 8 shows the experimental results of the above model on the Weibo16 dataset. The proposed method performs best, with an accuracy of 96.3% and an F1-score of 96.3%, which is 0.6% higher than the best baseline.

The results show that methods based on hand-crafted features, such as DTR, DTC, RFC, SVM-RBF, and SVM-TS, exhibit limitations in capturing pertinent features. Notably, RFC and SVM-TS perform significantly better due to their incorporation of supplementary structural or temporal features. However, these methods still fall notably behind models that eschew the need for feature engineering.

Within the propagation tree-based method, cPTK extracts linguistic and structural features from the propagation tree, followed by classification through a support vector machine. Since RvNN models the spread process of rumors as a tree structure, it is better suited for modeling the propagation tree. However, the tree structure's limitations may cause information loss and incomplete representation when modeling the propagation process, which is less adaptable and comprehensive than graph structure-based methods.

Within the deep learning-based approach, GRU uses a recurrent neural network to grasp semantic associations and temporal patterns among comments. PPC models the propagation process by combining user features with the propagation path features so that PPC can more comprehensively capture changes in user features. However, PPC relies on sequence modeling, which makes it difficult to capture complex relationships in nodes when processing graph structure information.

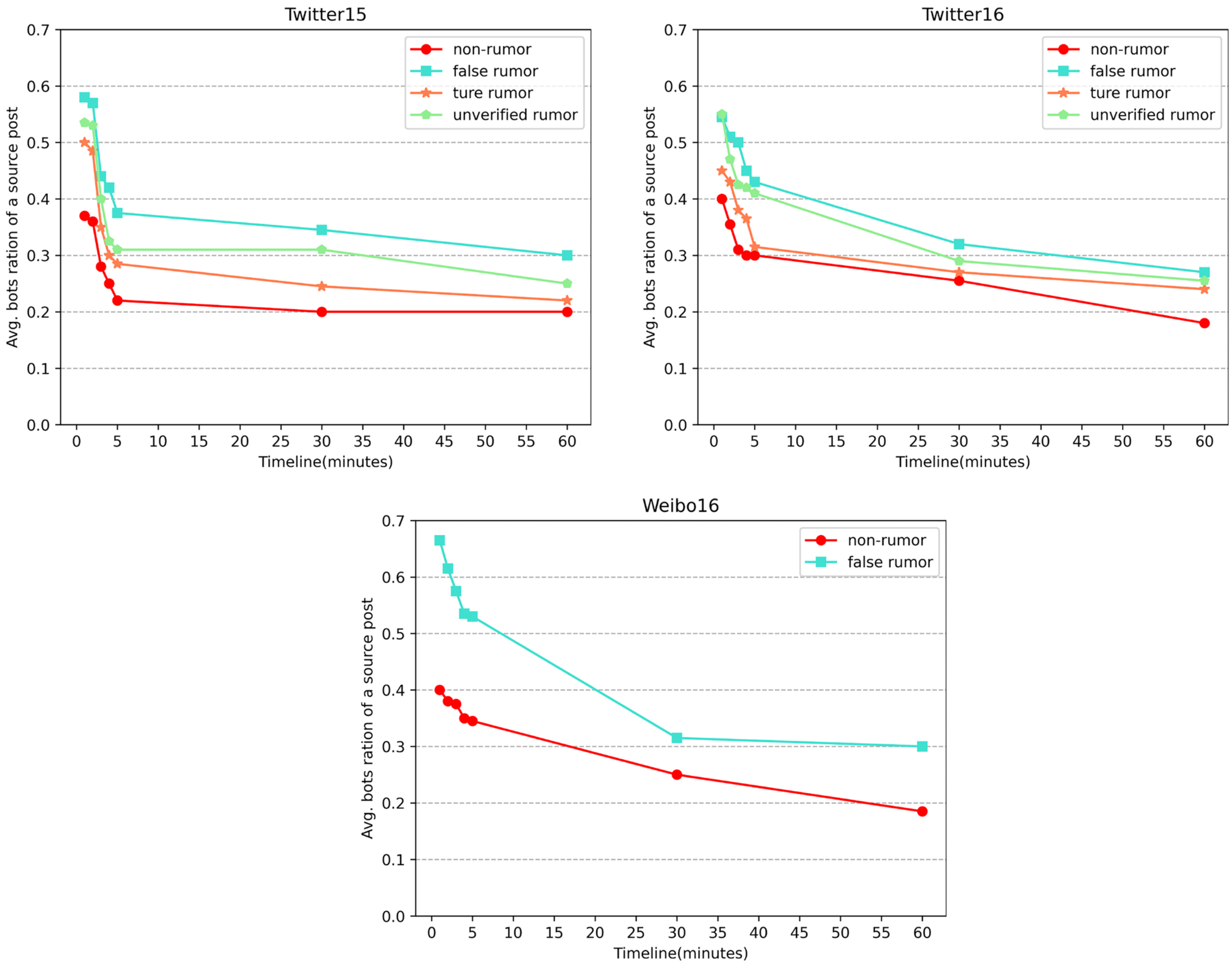


Fig. 5 Results of early rumor detection on Twitter15, Twitter16 and Weibo16

In addition, the approaches based on user propagation features or user credibility, such as GLAN, SMAN, and SBAG, model news and users as a heterogeneous graph, leveraging user credibility to enhance rumor detection. We also observe that SBAG surpasses GLAN and SMAN in effectiveness because SBAG has heightened accuracy in identifying rumors propagated by social bots.

In summary, our model has assimilated the strengths of these models and made improvements to achieve higher precision and accuracy in the rumor detection task. The design of the LGT model takes into account the rumor propagation structure on different social media platforms. Specifically, we design a long-range graph transformer that uses a traditional GNN subnetworks as the backbone to collect information from close neighbors and leaves long-range dependency learning to the transformer subnetworks. Our transformer application focuses each node on other nodes, motivating the transformer to learn the most important node-node relationships. Therefore, our model can flexibly adapt to different network topologies and effectively apply to different types of social media platforms. The accuracy rates on the three datasets reached 94.0%, 95.7% and 96.3%, outperforming all other baseline models. The results show that our model more effectively distinguishes rumor by capturing the graph structure information of news spread through the graph convolutional attentive network and structure-aware graph transformer.

5.4.4 Analysis of early detection

Early detection holds significant importance for rumor detection as it aligns with the imperative of timely intervention. The primary objective of early detection is to swiftly identify rumors from genuine information as they begin to spread. In early detection, the key challenge lies in correctly discerning rumors as they initiate their dissemination.

Table 6 Experimental results on Twitter15 dataset

Twitter15					
Method	Acc.	F1			
		NR	FR	TR	UR
DTR	0.409	0.501	0.311	0.364	0.473
DTC	0.454	0.733	0.355	0.317	0.415
RFC	0.565	0.810	0.422	0.401	0.543
SVM-RBF	0.318	0.455	0.037	0.218	0.225
SVM-TS	0.544	0.796	0.472	0.404	0.483
cPTK	0.750	0.804	0.698	0.765	0.733
GRU	0.646	0.792	0.574	0.608	0.592
RvNN	0.723	0.682	0.758	0.821	0.654
PPC	0.842	0.811	0.875	0.818	0.790
GLAN	0.905	0.924	0.917	0.852	0.927
SMAN	0.929	0.922	0.945	**0.915**	0.933
SBAG	0.938	0.963	0.950	0.904	0.933
LGT	**0.940**	**0.964**	**0.959**	0.906	**0.933**

Table 7 Experimental results on Twitter16 dataset

Twitter16					
Method	Acc.	F1			
		NR	FR	TR	UR
DTR	0.414	0.393	0.272	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-RBF	0.321	0.423	0.085	0.419	0.037
SVM-TS	0.574	0.755	0.420	0.571	0.526
cPTK	0.732	0.740	0.709	0.836	0.686
GRU	0.633	0.772	0.489	0.686	0.593
RvNN	0.737	0.662	0.743	0.835	0.708
PPC	0.863	0.820	0.898	0.843	0.837
GLAN	0.902	0.921	0.869	0.847	0.968
SMAN	0.935	0.936	0.920	0.915	**0.979**
SBAG	0.946	0.943	**0.933**	0.926	0.978
LGT	**0.957**	**0.948**	0.930	**0.968**	0.978

To evaluate the performance of LGT early detection, we set different detecting deadlines, where we only utilize users' interaction behavior preceding these deadlines to evaluate the early detection performance. Figure 6 shows the early detection results on Twitter15, Twitter16, and Weibo16 across varying dissemination intervals. Within 0 to 4 h, LGT achieves 90% accuracy on Twitter15 and 95% accuracy on Twitter16 and Weibo16, outperforming other baselines, demonstrating that our model has exceptional proficiency in early detection. When the time delay varies from 4 to 24 h, as news propagates, the augmentation of intricate user interaction behavior can potentially introduce more noise. However, our model demonstrates a tendency towards stability. Consequently, the research results show that the model boasts enhanced stability and robustness.

5.5 Ablation study

In order to evaluate whether long-range information is truly essential for rumor detection, we conducted a study on the ablation of the performance varying hop-range of graph transformer, allowing nodes to focus on the 1-, 3-, 5-, and 7-hop neighborhoods within the graph transformers. Results are included in Table 9. We can see that the transformer module plays an important role in extracting features from the local fields, and long-range information helps with the final prediction of the LGT model.

To discern the individual contribution of each module or feature to the overall model performance and to facilitate model optimization, we conducted an ablation study, and the experiments are as follows:

1. -Trans: Removing the graph transformer while keeping GNN part for user-user interaction graphs.
2. -CA: Removing the news publishing module and only using text features and aggregated interaction features to detect rumors.
3. -GT: Removing the news propagation module and only using text features and publishing features to detect rumors.
4. -C-G: Deleting two components mentioned 2) and 3) and only using text features to detect rumors.
5. -Text: Removing the news content module and only using publishing and interaction features to detect rumors.
6. -Score: Removing the bot detection module and not using user credibility scores as additional information.

As shown in Table 10, we can observe that:

We first evaluate the impact of removing the graph transformer while keeping GNN part for user-user interaction graphs. We can see that the performance of the model on the three datasets decreased by 1.4%, 2.7%, and 0.9%, respectively. The results indicate that using the transformer module to capture long-range information has a positive effect on rumor detection.

Then, we evaluate the impact of the user publishing and interaction module. Removing one of the modules results in a 1 to 4 percent drop in performance on each of the three datasets while removing both modules result in a 5 to 8 percent drop. The results show that user behavior features significantly affect rumor detection.

Table 8 Experimental results on Weibo16 dataset

Weibo16							
Method	Acc.	NR			FR		
		Prec	Rec	F1	Prec	Rec	F1
DTR	0.732	0.736	0.749	0.737	0.738	0.715	0.726
DTC	0.831	0.815	0.847	0.830	0.847	0.815	0.831
RFC	0.849	0.947	0.739	0.830	0.786	0.959	0.864
SVM-RBF	0.818	0.815	0.824	0.819	0.822	0.812	0.817
SVM-TS	0.857	0.878	0.830	0.857	0.839	0.885	0.861
GRU	0.910	0.952	0.864	0.906	0.876	0.956	0.914
PPC	0.921	0.949	0.889	0.918	0.896	0.962	0.923
GLAN	0.946	0.949	0.943	0.946	0.943	0.948	0.945
SMAN	0.951	0.967	0.936	0.951	0.937	0.967	0.952
SBAG	0.957	0.967	0.947	0.957	0.947	0.967	0.957
LGT	**0.963**	**0.971**	**0.955**	**0.963**	**0.955**	**0.971**	**0.963**

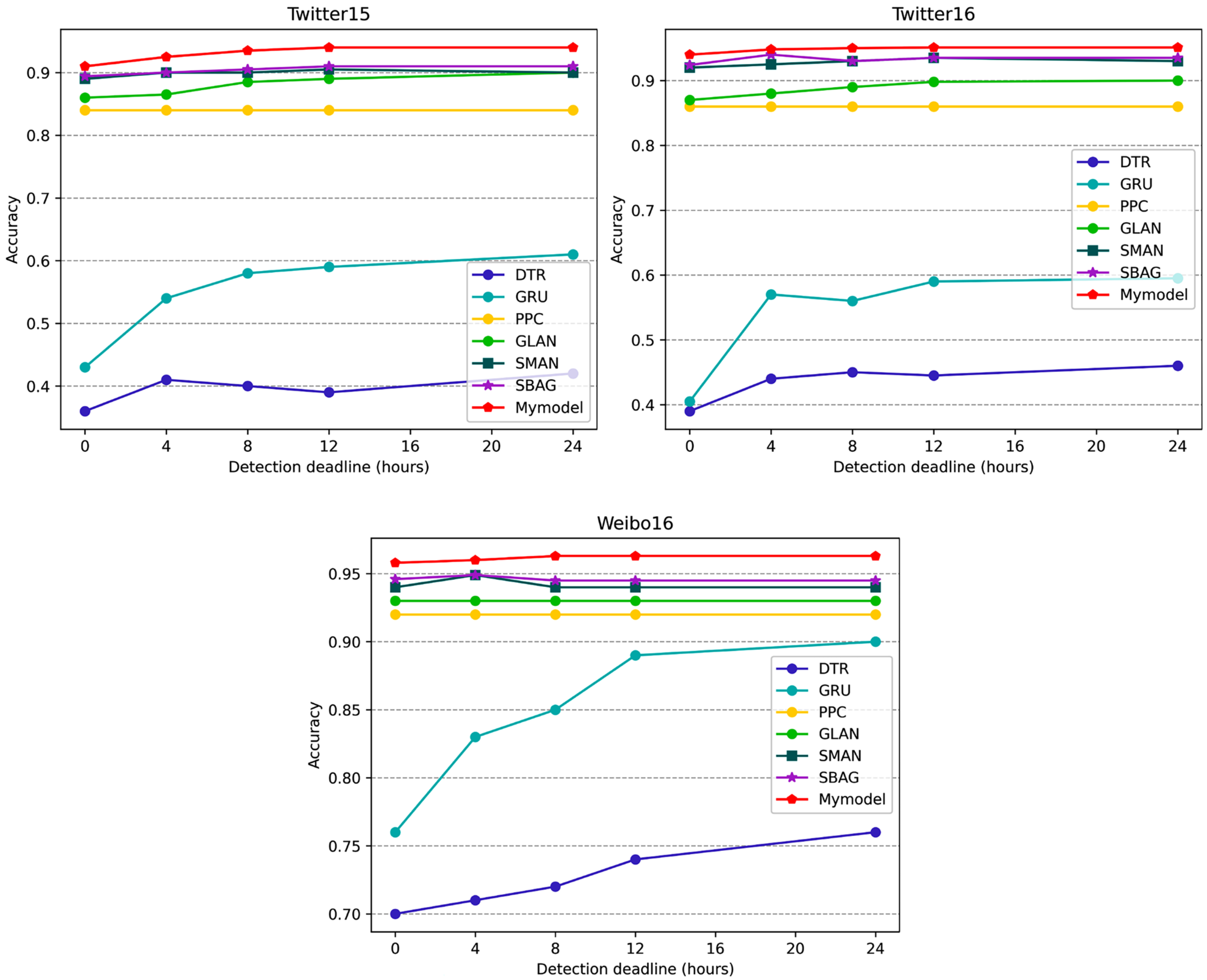


Fig. 6 Results of early rumor detection on Twitter15, Twitter16 and Weibo16

Table 9 Ablation Study on the hop-range of graph transformer (Acc.)

	Twitter15	Twitter16	Weibo16
1-hop	0.930	0.942	0.955
3-hop	0.936	0.949	0.960
5-hop	0.939	0.955	0.963
7-hop	**0.940**	**0.957**	**0.963**

Table 10 Ablation Study (Acc.)

Method	Twitter15	Twitter16	Weibo16
LGT	**0.940**	**0.957**	**0.963**
-Trans	0.926	0.930	0.954
-CA	0.925	0.924	0.953
-GT	0.909	0.929	0.954
-C-G	0.865	0.897	0.915
-Text	0.639	0.706	0.919
-Score	0.934	0.940	0.954

Next, we evaluate the impact of text features on rumor detection. The absence of text features resulted in a substantial decrease in performance across all datasets, with a notable drop of 25 and 30 percent on Twitter15 and Twitter16, respectively. The results show that the text features of news are indispensable for effective rumor detection.

We also evaluate the impact of user credibility scores on rumor detection. After incorporating user behavior and textual features, removing user credibility extracted by the bot detection module resulted in a decrease of around one percentage point in performance. The results show that including user credibility scores as additional information positively contributes to rumor detection.

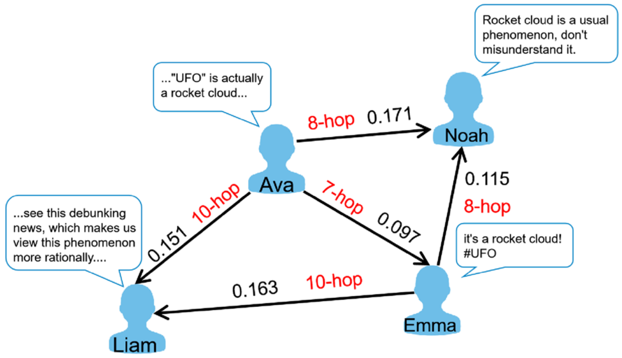

Rocket cloud is a usual phenomenon, don't misunderstand it.
..."UFO" is actually a rocket cloud...
8-hop 0.171
Noah
Ava
0.115
8-hop
7-hop 0.097
...see this debunking news, which makes us view this phenomenon more rationally....
0.151 10-hop
it's a rocket cloud! #UFO
0.163 10-hop
Emma
Liam

Fig. 8 Visualize some users with their weights

5.6 Case study

In order to visually understand the effectiveness of the transformer, we randomly selected some users from the Twitter15 dataset. Figure 7 shows the attention graph of the contribution of neighboring nodes of a user node to semantic information when representing the target user node. We observed that attention maps exhibit patterns similar to those found in NLP applications of transformers: some nodes obtain significant weights from many other nodes, regardless of their distance. For example, in Fig. 8, we found a high attention score between user node Ava and user nodes Noah, Liam, and Emma. Further analysis of the dataset reveals that these users have a high level of attention on social media, and their tweets are seen and forwarded by more people. Therefore, the transformer will give these users higher attention weights to capture the semantic information of these users. The discovery indicates that even if the user node Ava is 8

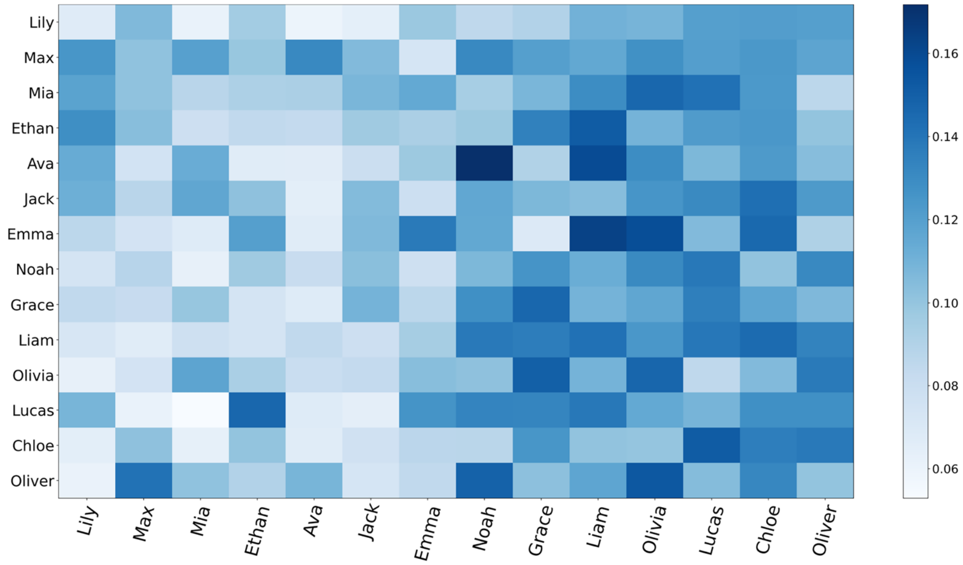


Fig. 7 Visualization of attention maps from self-attention in the transformer module

hops away from the user node Noah, 10 hops away from the user node Liam, and 7 hops away from the user node Emma, the transformer can still effectively capture semantic information between these users, further verifying the effectiveness of the transformer in capturing long-distance dependency relationships.

6 Conclusion and future work

To early detect and slow down the spread of rumors and mitigate their impact on society, this paper proposes an early rumor detection method that combines a graph convolutional attentive network and structure-aware graph transformer. Firstly, considering the impact of bots on rumor propagation, we extract users' credibility scores through a bot detection module to enhance user information. Secondly, by mining user features associated with the dissemination of true and false information and capturing complex information propagation among users, we can extract higher-quality news publishing features and interaction features for more efficient rumor detection. The model constructs a propagation graph for news, where the graph convolutional attentive network is employed to extract news publishing features in the publisher-news graph; the structure-aware graph transformer is utilized to capture interaction features during the propagation process; and CNNs are used to extract text features from news content. Furthermore, the model uses the attention mechanism to fuse the information extracted from user retweeting behaviors with source news to obtain aggregated interaction features. Finally, the model combines publishing features, aggregated interaction features, and text features to generate a new representation.

Experimental results on three real datasets demonstrate that the proposed LGT method achieves excellent performance in both rumor detection and early detection tasks, outperforming other baseline models. Furthermore, ablation experiments conducted on LGT provide additional validation of the effectiveness and rationality of its constituent modules.

In the future work, we plan to consider the dynamics of information dissemination, capturing the spatial and temporal structures of messages as dynamic propagation representations, so that the model can better adapt to new social media data and events. In addition, we will explore more efficient methods for adversarial attacks to ensure the robustness of the model is maintained when malicious users attempt to deceive or evade the model. With the rise of LLMs, Sun et al. (2023c), Sun et al. (2023d) put graph hint learning at the forefront of AGI technology, highlighting its innovation and potential in processing complex graph data. We can also use AGI to analyze data on social media, understand people's behaviors, attitudes and preferences, and make corresponding adjustments or decisions to optimize our models.

Author Contributions All authors contributed to the study conception and design.

Data availability and access The datasets and code used during this study are available upon reasonable request to the authors.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

Ethical and informed consent for data used Not applicable.

References

Alon U, Yahav E (2020) On the bottleneck of graph neural networks and its practical implications. In: international conference on learning representations

Bian T, Xiao X, Xu T, Zhao P, Huang W, Rong Y, Huang J (2020) Rumor detection on social media with bi-directional graph convolutional networks. In: proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 549–556

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

Chen T, Wong RC-W (2020) Handling information loss of graph neural networks for session-based recommendation. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 1172–1180

Chen T, Li X, Yin H, Zhang J (2018) Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In: trends and applications in knowledge discovery and data mining: PAKDD 2018 Workshops, BDASC, BDM, ML4Cyber, PAISI, DaMEMO, Melbourne, VIC, Australia, June 3, 2018, Revised Selected Papers 22, pp. 40–52. Springer

Giachanou A, Rosso P, Crestani F (2019) Leveraging emotional signals for credibility detection. In: proceedings of the 42nd international ACM sigir conference on research and development in information retrieval, pp. 877–880

Huang Z, Lv Z, Han X, Li B, Lu M, Li D (2022) Social bot-aware graph neural network for early rumor detection. In: proceedings of the 29th international conference on computational linguistics, pp. 6680–6690

Hu L, Yang T, Zhang L, Zhong W, Tang D, Shi C, Duan N, Zhou M (2021) Compare to the knowledge: graph neural fake news detection with external knowledge. In: proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (Volume 1: Long Papers), pp. 754–763

Jin Z, Cao J, Zhang Y, Luo J (2016) News verification by exploiting conflicting social viewpoints in microblogs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30

Kwon S, Cha M, Jung K (2017) Rumor detection over varying time windows. PloS one 12(1):0168344

Liu Y, Wu Y-F (2018) Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32

Liu B, Sun X, Ni Z, Cao J, Luo J, Liu B, Fu X (2020) Co-detection of crowdturfing microblogs and spammers in online social networks. World Wide Web 23:573–607

Liu B, Sun X, Meng Q, Yang X, Lee Y, Cao J, Luo J, Lee RK-W (2022) Nowhere to hide: Online rumor detection based on retweeting graph neural networks. IEEE transactions on neural networks and learning systems

Li Q, Han Z, Wu X-M (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: proceedings of the AAAI conference on artificial intelligence, vol. 32

Li G, Müller M, Ghanem B, Koltun V (2021) Training graph neural networks with 1000 layers. In: international conference on machine learning, pp. 6437–6449. PMLR

Mann WC, Thompson SA, INST UOSCMDRIS (1987) Rhetorical structure theory: a theory of text organization

Ma J, Gao W, Wei Z, Lu Y, Wong K-F (2015) 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, pp. 1751–1754

Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks

Ma J, Gao W, Wong K-F (2017) Detect rumors in microblog posts using propagation structure via kernel learning. Assoc Computat Lingu

Ma J, Gao W, Wong K-F (2018) Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics

Meng Q, Yan H, Liu B, Sun X, Hu M, Cao J (2023) Recognize news transition from collective behavior for news recommendation. ACM Trans Inform Syst 41(4):1–30

Przybyla P (2020) Capturing the style of fake news. In: proceedings of the AAAI conference on artificial intelligence, 34:490–497

Sayyadiharikandeh M, Varol O, Yang K-C, Flammini A, Menczer F (2020) Detection of novel social bots by ensembles of specialized classifiers. In: proceedings of the 29th ACM international conference on information & knowledge management, pp. 2725–2732

Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. ACM SIGKDD Explorat Newslett 19(1):22–36

Shu K, Cui L, Wang S, Lee D, Liu H (2019) defend: explainable fake news detection. In: proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 395–405

Song C, Shu K, Wu B (2021) Temporally evolving graph neural network for fake news detection. Inform Process Manag 58(6):102712

Sun X, Yin H, Liu B, Chen H, Cao J, Shao Y, Viet Hung NQ (2021) Heterogeneous hypergraph embedding for graph classification. In: proceedings of the 14th ACM international conference on web search and data mining, pp. 725–733

Sun X, Yin H, Liu B, Meng Q, Cao J, Zhou A, Chen H (2022) Structure learning via meta-hyperedge for dynamic rumor detection. IEEE transactions on knowledge and data engineering

Sun X, Cheng H, Liu B, Li J, Chen H, Xu G, Yin H (2023) Self-supervised hypergraph representation learning for sociological analysis. IEEE Transon Knowl Data Eng

Sun X, Cheng H, Dong H, Qiao B, Qin S, Lin Q (2023) Counter-empirical attacking based on adversarial reinforcement learning for time-relevant scoring system. IEEE Trans Knowl Data Eng

Sun X, Zhang J, Wu X, Cheng H, Xiong Y, Li J (2023) Graph prompt learning: A comprehensive survey and beyond. arXiv preprint arXiv:2311.16534

Sun X, Cheng H, Li J, Liu B, Guan J (2023) All in one: multi-task prompting for graph neural networks

Wu L, Rao Y, Zhao Y, Liang H, Nazir A (2020) Dtca: Decision tree-based co-attention networks for explainable claim verification. arXiv preprint arXiv:2004.13455

Wu Z, Jain P, Wright M, Mirhoseini A, Gonzalez JE, Stoica I (2021) Representing long-range context for graph neural networks with global attention. Adv Neural Inform Process Syst 34:13266–13279

Xia R, Xuan K, Yu J (2020) A state-independent and time-evolving network for early rumor detection in social media. In: proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 9042–9051

Xu K, Li C, Tian Y, Sonobe T, Kawarabayashi K-i, Jegelka S (2018) Representation learning on graphs with jumping knowledge networks. In: international conference on machine learning, pp. 5453–5462. PMLR

Yang F, Liu Y, Yu X, Yang M (2012) Automatic detection of rumor on sina weibo. In: proceedings of the ACM SIGKDD workshop on mining data semantics, pp. 1–7

Yuan C, Ma Q, Zhou W, Han J, Hu S (2019) Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In: 2019 IEEE international conference on data mining (ICDM), pp. 796–805. IEEE

Yuan C, Ma Q, Zhou W, Han J, Hu S (2020) Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. arXiv preprint arXiv:2012.04233

Yu F, Liu Q, Wu S, Wang L, Tan T, et al (2017) A convolutional approach for misinformation identification. In: IJCAI, pp. 3901–3907

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

Zhou X, Zafarani R (2019) Network-based fake news detection: a pattern-driven approach. ACM SIGKDD Explorat Newslett 21(2):48–60

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.