Neural Processing Letters (2023) 55:9831–9850
https://doi.org/10.1007/s11063-023-11229-w

A Rumor Detection Model Incorporating Propagation Path Contextual Semantics and User Information

Lin Bai[1] · Xueming Han[1] · Caiyan Jia[1]

Accepted: 7 March 2023 / Published online: 25 March 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
Currently, social media is full of rumors. To stop rumors from spreading further, rumor detection has received increasing attention. Recent rumor detection methods treat all propagation paths and all nodes on the paths as equally important, resulting in models that fail to extract the key features. In addition, most methods ignore user features, leading to limitations in the performance improvement of rumor detection. To address these problems, we propose a Dual-Attention Network model on propagation Tree structures named DAN-Tree, where a node-and-path dual-attention mechanism is designed to organically fuse deep structure and semantic information on the propagation structures of rumors, and path oversampling and structural embedding are employed to enhance the learning of deep structures. Finally, we deeply integrate user profiles into the propagation trees in DAN-Tree, thus proposing the DAN-Tree++ model to further improve performance. Empirical studies on four rumor datasets have shown that DAN-Tree outperforms the state-of-the-art rumor detection models learning on propagation structures, and the results on two datasets with user information validate the superior performance of DAN-Tree++ over other models using both user profiles and propagation structures. What's more, DAN-Tree, especially DAN-Tree++, has achieved the best performance on early detection tasks.

Keywords Rumor detection · Attention mechanism · Propagation structure · User feature

1 Introduction

In this era of rapid development and widespread use of Internet technology, massive amount of information spreads through human communication channels at an extremely fast pace.

✉ Caiyan Jia
cyjia@bjtu.edu.cn

Lin Bai
22120349@bjtu.edu.cn

Xueming Han
19120358@bjtu.edu.cn

1 School of Computer and Information Technology and Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China

While Internet technology provides convenience to people, it also enables rumors to have a huge negative impact on individuals, countries and societies through larger data size, faster dissemination and more undetectable disguises. For example, a week after the Boston bombings in the United States in 2013, hackers hijacked the Associated Press Twitter account and posted a fake message: "Breaking: Two explosions at the White House, Obama injured". Before the AP and the White House came out to clarify, U.S. stocks fell more than 140 points in just a few minutes, losing billions of dollars. With the advent of COVID-19 at the end of 2019, many epidemic-related rumors have appeared on social media all over the world, leading to a very bad impact on epidemic prevention and social stability efforts worldwide. Therefore, it is imperative to develop automatic and efficient rumor detection methods.

Existing rumor detection methods can be broadly classified into two categories: traditional feature engineering-based methods and deep learning-based methods. Traditional machine learning-based models use traditional classifers including SVM (support vector machine), decision tree, random forests, etc. They heavily rely on manually extracted features. Although these methods have made some progress in rumor detection tasks, feature engineering is a very time-consuming and laborious task, and some potential features are easily ignored. On the contrary, deep learning methods can automatically learn various hidden features in datasets during the training process, which greatly alleviates the excessive manpower requirements and improves the adaptability of the models in each scenario. The current state-of-the-art (SOTA) rumor detection methods include deep learning methods using propagation structures and text semantics of rumors, such as RvNN [1], PLAN [2], Bi-GCN [3], etc., and deep learning methods further incorporating user profiles, such as UMLARD [4], HGARD [5], etc. Although these models have achieved great success, the current works still suffer from the following problems.

First, the current SOTA models have insufficient learning ability of fusing propagation structure and text semantics. Most of the existing studies using propagation structures focus too much on the explicit direct response relationships between posts and ignore the implicit indirect relationships, which make the models oversimplify the interactivity between users. In addition, due to the different importance of the contained information, existing work does not focus on the fact that different posts have different importance to the branching paths of the propagation structure, and each branching path has different importance to a propagation structure. Second, the current SOTA methods using propagation structures and text semantics can not capture the credibility of individual posts induced by users. As is well known, posts are published by users, the credible user will lead to high credibility of his/her posts, thus further boosting the performance of rumor detection if the models can also well learn the user features.

Therefore, in this study, we propose DAN-Tree and DAN-Tree++ to address the shortcomings of the existing works on rumor detection mentioned above. The experimental results show that both of these models achieve advanced performance on a variety of rumor detection datasets [6–8] and DAN-Tree++ which further fuse users' features into DAN-Tree has the best performance.

The main contributions of our work are summarized as follows.

1. We propose DAN-Tree, a dual-attention network model on propagation tree structures. DAN-Tree utilizes Transformer encoding blocks as feature extractors to model the implicit relationships among posts on the propagation path. It also further focuses on the features of key post nodes and key paths through post-level attention and path-level attention mechanisms.

2. Based on the DAN-Tree model, we propose the rumor detection model DAN-Tree++ that fuses user features and propagation structures. DAN-Tree++ introduces user features in two aspects. One is to fuse user features on text features and introduce user's trustworthiness information. The other is to fuse global user features on propagation structure features to introduce the overall characteristics of user feature sequences in the rumor propagation process.
3. We conducted a series of experiments on multiple real data sets. The experimental studies show that our model exhibits superior performance compared to other baseline models and achieves the best results on early detection tasks.

The rest of the paper is organized as follows. Section 2 presents the related work. Next, Sect. 3 introduces the proposed method. The experimental results and analysis are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related Work

In the context of the reality that rumors have caused many adverse effects on normal life, rumor detection has gradually become a hot research content in academia. Propagation information-based approaches, which only use post texts and post-post relations (in terms of chronology or propagation structure), have been validated to be very effective for debunking rumors, especially methods learning on propagation structures. The current studies have also proved that further fusing user profiles, user-post relations with propagation information can promote rumor detection performance. In the following section, we will briefly introduce these two kinds of methods in the literature.

2.1 The Propagation Information-Based Methods

The propagation information-based methods are based on wisdom of the crowd (i.e., other users' comments on the authenticity of posts). Not limited to extracting representative features from the textual content of posts, propagation-based approaches make more use of the social contextual information of posts, use the comments of users in real scenarios as an auxiliary judgment, and effectively exploit users' sentiments and factual positions. Ma et al. [8] modeled the social contextual information of posts as variable-length sequences according to published time of posts and applied Recurrent Neural Network (RNN) to detect rumors for the first time. Chen et al. [9] introduced a soft attention mechanism based on the RNN structure for learning the importance of different posts in sequential text inputs. Although RNN structures are naturally suitable for learning temporal features of sequential text, they suffer from long-distance dependent information loss and inability to accelerate operations in parallel. The Transformer structure proposed by Vaswani et al. [10] used an attention mechanism to model long-distance interactions of words to establish connections for arbitrary token pairs in sequential sequences. Its excellent results have been achieved in many natural language processing tasks such as machine translation [11], language modeling [12] and sentiment analysis [13]. Accordingly, Khoo et al. [2] proposed the Post-Level Attention Network (PLAN), which adopted the Transformer structure on the time series of post events to construct an implicit relationship between any post pair in the input sequences. Thanks to the advantages of the Transformer structure, PLAN could make posts in the rumor propagation path pay attention to the long distance posts, enhance the interaction among posts, and achieve good detection results.

As we know, the actual propagation process of posts on social media is not simply arranged chronologically into a one-dimensional time series, but there is an important nonlinear propagation structure consisting of reply relationships among posts. Naturally, another group of propagation information-based methods learn rumor representations on propagation structures which describe who-replies-to-whom relations. Zubiaga et al. [14] proposed that posts can "self-correct" by sharing opinions, speculations, and evidence among users. It is shown that comments between users on the propagation structure can provide useful information for rumor detection tasks. Wu et al. [15] also validated the effectiveness of propagation structures in the field of rumor detection. Ma et al. [1] generated non-sequential tree-like rumor propagation structures based on the response relationships between posts and used recurrent neural networks (RvNN) to model rumor propagation trees in top-down or bottom-up manner. For the first time, the structural information of the propagation process and the semantic information of the texts were fused together. Khoo et al. [2] proposed a variant of PLAN, Sta-PLAN, which utilized a single additional variable to describe the structural relationships among posts. Bian et al. [3] constructed the rumor propagation process as a propagation graph and a dispersion graph from both top-down and bottom-up perspectives. The global representation of the two subgraphs was obtained and fused using a bi-directional graph convolutional network (Bi-GCN). The learning of semantic features of the source post text was also enhanced in the model. Likewise, Yang et al. [16] creates a propagation tree and a diffusion tree. The improved graph attention network (GAT) is utilized to extract propagation features and diffusion features in two different directions, while the multi-head attention mechanism is utilized to extract the semantic information of the source tweet. Zhang et al. [17] performed a summation operation on the word representation as the propagation path representation, further obtained the propagation tree representation by pooling operation, and used a neural topic model, Wasserstein Auto-Encoder (WAE) [18], to mine the hidden stance topics. Wei et al. [19] introduced the learning of edge uncertainty in propagation trees by Bayesian deep learning method based on Bi-GCN. Lv et al. [20] aggregates the propagation structure and text features of rumors using GAT, then records the historical state of the propagation structure using a temporal attention mechanism, and finally captures the features of the propagation structure over time using GRU. Those methods showed very good performance for uncovering the veracity of rumors using only post semantics and propagation structures. We believe those are the most effective rumor detection methods with less information.

2.2 Integrating User Information Methods

In social media, user information can reflect user behavioral characteristics and users' influence on the public [21]. Meanwhile existing work experimentally verified the enhancing effect of user information on rumor detection. Lu et al. [22] used CNN, RNN and GCN structures to obtain different representations of user feature sequences based on the extraction of source post text features and achieved better results on a binary classification task to predict the veracity of rumors. Bing et al. [23] used a dual co-attention module to fuse source post features with reply post features and user features, respectively. Yuan et al. [24] added the user nodes of published posts to form a heterogeneous graph based on connecting the text of each post through propagation relations. The global structural information and local semantic information were learned through a multi-headed attention mechanism. The User-aspect Multi-view Learning with Attention for Rumor Detection (UMLARD) model proposed by Chen et al. [4] obtained the representation of users under different views from three aspects: pictorial view, structural view and temporal view, and fused them with the text

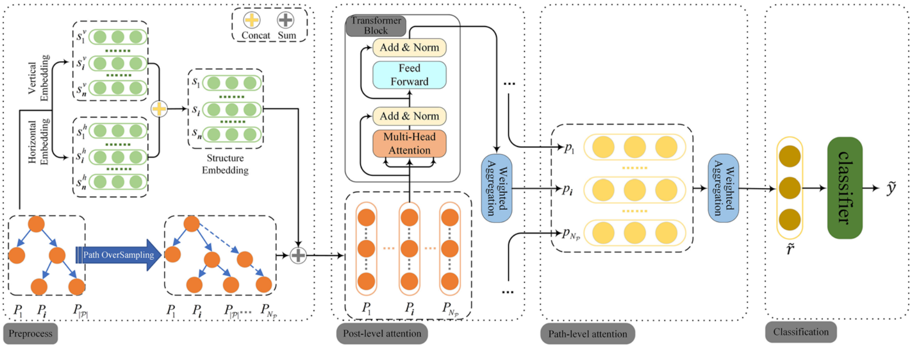

Fig. 1 The architecture of DAN-Tree

features of source posts to acquire the final rumor representations. Huang et al. [5] proposed the Heterogeneous Graph Attention network for Rumor Detection (HGARD) model to construct a source post-word-user heterogeneous graph, which could incorporate user features into the learning of global semantics of text content. Liu et al. [25] utilize the clues in users' comments, use the attention mechanism to fuse source microblog (tweet) with the comment-retweet information and extract interactive semantic features from it. These methods have achieved the SOTA performance in the literature. Thus, it may deserve fuse user information with propagation structure-based methods to further enhance the accuracy of these methods.

3 Proposed Methods

3.1 Problem Statement

We assume a rumor event contains a set of posts, $E = \{T_1, T_2, \cdots, T_{|E|}\}$, talking about the rumor and its corresponding category label is Y. As shown in Fig. 1"Preprocess", We construct E as a propagation tree $P = \{P_1, P_2, \cdots, P_{|P|}\}$ consisting of multiple propagation paths (consists of several comments with reply relationship), where $|P|$ denotes the number of propagation paths owned by E. P_i is the i-th propagation path of E and $P_i = \{T_{i1}, T_{i2}, \cdots, T_{i|P_i|}\}$ which means the post T_{ij+1} replies the post T_{ij} $(j = 1, 2, \cdots, |P_i| - 1)$, $|P_i|$ denotes the depth of the propagation path P_i, which is also the number of post nodes along the path, T_{i1} denotes the root node of P_i, which is also the root node of the whole propagation tree P (i.e., the source post node), $T_{i|P_i|}$ denotes the leaf node of P_i.

The goal of the rumor detection task is to learn a classifier f that, as shown in Eq. (1), maps a rumor event E to its corresponding category label Y.

$$f : E \rightarrow Y. \tag{1}$$

For the convenience of the readers, the major notations used in this paper are listed in Tab. 1.

Table 1 Major notations

Notations	Meaning
E	Rumor events
T	The source post of the rumor or reply post
P_i	The i-th propagation path of rumor
$\|P_i\|$	The depth of the propagation path
s	Structural embedding of post nodes
R	The sequence of fixed-length propagation Paths after oversampling
W	Words in the post
t	Post text representation
$\hat{t}$	Post text representation after adding structure embedding
t'	Post text representation after Transformer
p	Representation of propagation path P
$\tilde{r}$	Representation of propagation tree
U	User information node
u	Extracted user features
h	New text features after fusion with User features
S_p	Representation of propagation tree (Same as $\tilde{r}$)
S_u	The global user feature representation

3.2 The DAN-Tree Structure

The overall structure of DAN-Tree model is shown in Fig. 1, which consists of a preprocess model including structure embedding, path oversampling, post-level attention module, path-level attention module and rumor classification module.

3.2.1 Structure Embedding

Nguyen et al. [11] proposed a method of hierarchical embedding. To some extent, it solves the problem of Transformer's difficulty in handling tree-structured data. We apply the above method to the rumor detection task so that it can learn the horizontal and vertical location information in rumor propagation structures in space. We call this structure embedding. An illustrative example is showed in Fig. 2.

We compute a corresponding structure embedding representation for each post node in each path of a tree (e.g., the tree showed in Fig. 2a) by using Eq. (2) (the paths of the tree in Fig. 2a are drawn in Fig. 2b).

$$s_{ij} = e_x^v \otimes e_y^h, \quad x = |V_j^i| \text{ and } y = |H_j^i|. \tag{2}$$

where $s_{ij} \in \mathrm{R}^d$ represents the structure embedding representation of post node T_{ij} in path P_i of a propagation tree. $e_x^v, e_y^h \in \mathrm{R}^{d/2}$ represent the x-th and y-th row vectors in the trainable vertical position embedding matrix $E^v \in \mathrm{R}^{h_P \times d/2}$ and horizontal position embedding matrix

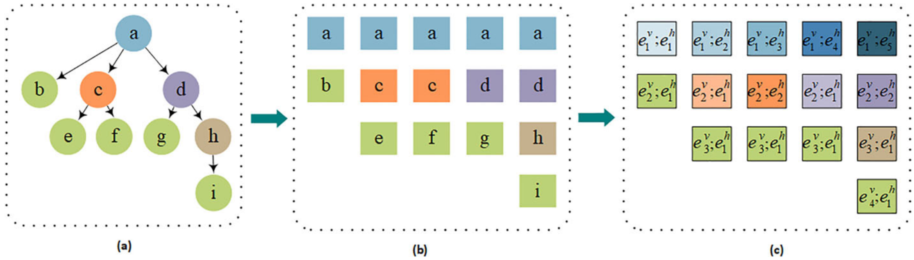

Fig. 2 **a** represents a rumor propagation tree structure, and each node represents a realistic post node. **b** is a path view composed of the propagation paths corresponding to **a**. According to the path view of **b**, we can encode information about the position of the nodes in the tree structure from both horizontal and vertical views, as shown in **c**

$E^h \in \mathrm{R}^{|P|\times d/2}$, respectively. $\otimes$ denotes the concatenation operation, and h_P represents the maximum value of depth in $|P|$ propagation paths. $x = |V_j^i|$ represents the vertical position information of T_{ij} (top-down encoding) in the subordinate propagation path, where $V_j^i = \{T_{ij} | 1 \le k \le j\}$ represents the set of nodes in the path during the flow of information from the root node T_{i1} to the node T_{ij}. $y = |H_j^i|$ represents the total number of times T_{ij} has appeared while in the current propagation path (left-to-right encoding), i.e., the horizontal position information in the propagation tree structure, where $H_j^i = \{T_{kj} \mid 1 \le k \le i \text{ and } T_{kj} = T_{ij}\}$ represents the set of occurrences of T_{ij} in path P_i and all its previous paths.

From the example in Fig. 2, for any node in the tree, we can encode its structure information in the tree by a horizontal position vector and a vertical position vector to portray the whole tree structure information showed in Fig. 2c.

3.2.2 Path Oversampling

A small number of propagation paths may make it difficult to get enough useful user feedback. To solve the above problem, this section proposes path oversampling method by randomly resampling the original variable-length propagation path sequence P to obtain a fixed-length propagation path sequence $R = \{R_1, R_2, \cdots, R_{|P|}, \cdots, R_{N_P}\}$ with a fixed length N_P. We adopt larger sampling probability for deeper propagation paths in the process of random resampling so that the model can get a richer feedback to the source node. For variable-length propagation path sequences with length $|P|$ which is greater than N_P, we use only the first N_P propagation paths. The details of the path oversampling method are shown in Algorithm 1, where $2|P_i| - 3$ represents the number of oversampling paths. This is our manually set parameter weights. We put the subscript i of the $2|P_i| - 3$ paths P_i into the list I to be sampled. The more propagation path subscripts in I, and the greater the probability of $|P_i|$ is sampled.

For convenience, the sequence of these fixed-length propagation paths R is still denoted by P with fixed number of paths N_P in the following. According to Sect. 3.2.1, each post node T_{ij} has a corresponding structural embedding representation s_{ij}. In the process of path oversampling, for a path P_k obtained by sampling from the original path P_i, we set the structure embedding corresponding to the post node T_{kj} still be s_{ij} (i.e., $s_{kj} = s_{ij}$).

Algorithm 1 Path oversampling algorithm

Input: The original sequence of variable-length propagation paths P
Output: The sequence of fixed-length propagation paths after oversampling R
Initialize an empty set of path subscripts $I = \{\}$
for $i = 1 \rightarrow N_P$ **do**
  if $i \leq |P|$ **then**
    $R_i \leftarrow P_i$
    for $j = 1 \rightarrow (2|P_i| - 3)$ **do**
      $I \leftarrow I \cup \{i\}$
    end for
  else
    $j \leftarrow Random(I)$
    $R_i \leftarrow P_i$
  end if
end for

3.2.3 Post-Level Attention

The post T_{ij} consists of a series of words denoted as $T_{ij} = \left\{W_{ij1}, W_{ij2}, \cdots, W_{ij|T_{ij}|}\right\}$, where W_{ijk} represents the k-th word in T_{ij} and its corresponding word embedding is $w_{ijk} \in \mathrm{R}^d$, and d is the dimension of the word embedding.

We obtain the post representation t_{ij} of the post T_{ij} by the maximum pooling method showed in Eq. (3).

$$t_{ij} = \text{MaxPooling}\left(\left\{w_{ij1}, w_{ij2}, \cdots, w_{ij|T_{ij}|}\right\}\right). \tag{3}$$

To take into account the spatial information of the corresponding propagation tree structure, we add the structural embedding s_{ij} showed in Eq. (2) to the original post representation t_{ij}. This process can be formulated by the Eq. (4).

$$\hat{t}_{ij} = t_{ij} + s_{ij}. \tag{4}$$

In terms of the time scale of the propagation path, a certain reply post T_{ij} is influenced not only by the parent node to which it directly replies, but also most likely by all the earlier posts (T_{ik}, $1 \leq k \leq j$), especially the root node. Therefore, for a propagation path P_i, in order to learn the long-range implicit relationships among the individual posts where P_i contains and also to allow parallelized training of the model, we apply the encoding blocks in the Transformer structure [10] to the sequence of post representations on P_i via Eq. (5).

$$\left\{t'_{i1}, t'_{i2}, \cdots, t'_{i|P_i|}\right\} = \text{Trans}\left(\left\{\hat{t}_{i1}, \hat{t}_{i2}, \cdots, \hat{t}_{i|P_i|}\right\}\right). \tag{5}$$

Since different posts may have different importance for the representation of the propagation path they belong to, in order to measure this importance we obtain the post-level context vector c^t_{ij} for the post representation t'_{ij} using the attention method of Eq. (6), where we use $LeakyReLU$ as the activation function, a_t for the weight vector, and W_t for the weight matrix. The normalized importance weights β^t_{ij} are obtained by Eq. (7). After that, the post representations are weighted and summed by Eq. (8) to obtain the representation p_i of the propagation path P_i.

$$c^t_{ij} = a_t^T \cdot \text{LeakyReLU}\left(W_t t'_{ij}\right). \tag{6}$$

$$\beta^t_{ij} = \frac{\exp\left(c^t_{ij}\right)}{\sum_{k=1}^{|P_i|} \exp\left(c^t_{ik}\right)}. \tag{7}$$

$$p_i = \sum_{j=1}^{|P_i|} \beta_{ij}^{t} t_{ij}'. \tag{8}$$

3.2.4 Path-Level Attention

Clearly, in the structure of a rumor propagation tree, each post does not express the same information, and thus the importance of propagation paths comprised of multiple posts to the entire tree varies. To enable the model to capture critical paths, here unlike Ma et al. [1] who used the $MaxPooling$ method directly or Bian et al. [3] who used the $MeanPooling$ method to obtain the final rumor representation, we again adopt the attention mechanism by introducing a path-level context vector c_i^p via Eq. (9). to measure the importance of the path representation p_i to the whole propagation tree structure representation, where the $LeakyReLU$ function is still used as the activation function a_p represents the weight vector and W_p denotes the weight matrix. The normalized importance weights β_i^p are obtained by Eq. (10). Finally, the weighted summation of the path representations is obtained by Eq. (11) for the propagation tree representation $\tilde{r}$.

$$c_i^p = a_p^T \cdot LeakyReLU\left(W_p p_i\right). \tag{9}$$

$$\beta_i^p = \frac{\exp\left(c_i^p\right)}{\sum_{k=1}^{N_P} \exp\left(c_k^p\right)}. \tag{10}$$

$$\tilde{r} = \sum_{i=1}^{N_P} \beta_i^p p_i. \tag{11}$$

3.2.5 Rumor Classification Module

In the rumor classification module, we input the obtained rumor propagation tree representation $\tilde{r}$ to a layer of feed forward neural network with a $Softmax$ layer, and calculate the model's predicted label $\tilde{y}$ for the corresponding rumor by Eq. (12).

$$\tilde{y} = \mathrm{Softmax}\left(W_r \tilde{r} + b_r\right). \tag{12}$$

We use cross entropy as a classification loss to measure the similarity between the true labels and the labels predicted by the model.

$$\mathrm{L}(y, \tilde{y}) = -[y \log \tilde{y} + (1-y)\log(1-\tilde{y})] + \lambda\|\theta\|_2^2. \tag{13}$$

Where y represents the true label of a rumor event, $\|\cdot\|_2^2$ denotes the L_2 regularization operation for all parameters θ in the model, and λ is the balance coefficient.

We use the Adam optimizer [26] to optimize the parameters in the model. Also, to suppress the degree of overfitting of the model, we add Dropout regularization method to the model.

3.3 The DAN-Tree++ Structure

For further improving the performance of DAN-Tree, we propose the DAN-Tree++ model by considering the users' credibility characterized by users' profiles. In summary, the DAN-Tree++ model consists of three main modules, user feature extraction and encoding, global user feature encoding, and rumor classification. The overall structure of the DAN-Tree++ model is shown in Fig. 3.

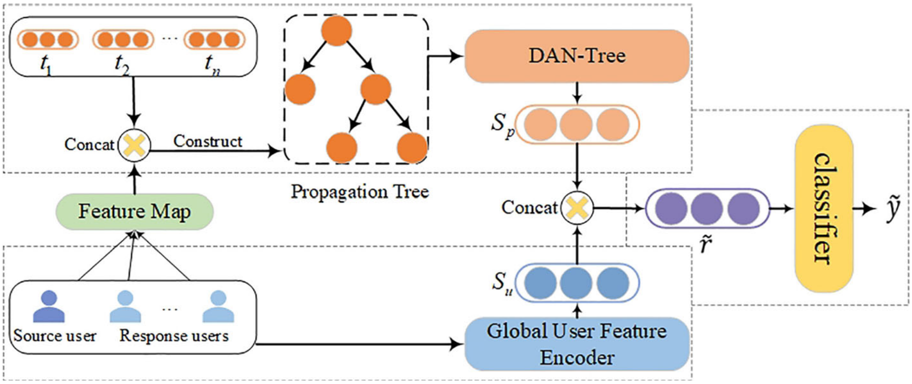

Fig. 3 The architecture of DAN-Tree++

Table 2 User Profile

No.	Feature	Variable type
1	Number of followers	Int
2	Number of followings	Int
3	Number of posts	Int
4	Number of likes	Int
5	Registration Time	Date
6	Verify	Bool
7	Geographic location	Bool

3.3.1 User Feature Extraction and Encoding

In order to represent the discretized user information in the form of feature vectors that can be trained by the model, we extract some of the user information of a user U_i into user features u_i in this subsection.

Considering the various types of information contained in user profiles, we selected seven types of user information shown in Tab. 2for the extraction of user features. For the boolean type variables such as whether to verify and whether to show geographic location, the feature extraction is performed using the unique hot coding method. For user information with discrete values such as number of followers x_{ij}, users with greater influence on the public usually have much more followers than ordinary users, so we eliminate the effect of extreme values by maximum-minimum normalization as shown in Eq. (14).

$$x'_{ij} = \begin{cases} \frac{\log x_{ij} - \log x_{ij}^{\min}}{\log x_{ij}^{\max} - \log x_{ij}^{\min}}, & x > 0 \\ 0, & x = 0 \end{cases}. \tag{14}$$

Where x_{ij} denotes the original value before the normalization operation, $x^{'}_{ij}$ denotes the standard value after normalization, and $x_{ij}^{\max}$ and $x_{ij}^{\min}$ denote the maximum and minimum values of the user information belonging to x_{ij} in the data set, respectively. After that, the values of these 7 transformed user information are concatenated to get $x_i = [x'_{i1}, x'_{i2}, \cdots, x'_{i7}]$ as the initial user characteristics of user U_i.

Before further operating on the user features, we found that the dimension size of x_i is only 7, which will result in the low-dimensional user features not being useful in the model training because dimension difference of user features compared with that of the text features is too large. Therefore, the feed forward neural network shown in Eq. (15) is further used to map the user features x_i to $u_i \in \mathrm{R}^{d_u}$.

$$u_i = \mathrm{ReLU}\left(W_u x_i + b_u\right), \tag{15}$$

where $W_u \in \mathrm{R}^{7 \times d_u}$ denotes the weight matrix and $b_u \in \mathrm{R}^{d_u \times 1}$ denotes the bias vector.

To effectively utilize the trustworthiness information reflected by user features, we fuse them with the corresponding text features to guide the learning of rumor features during the training process.

For the text representation t_{ij} corresponding to the post T_{ij} at the j-th node on the i-th propagation path P_i in a rumor propagation tree, it is fused with the user feature u_{ij} corresponding to the user U_{ij} who made the post as shown in Eq. (16) to obtain the new combined feature $h_{ij} \in \mathrm{R}^{d_p}$, where $d_p = d_t + d_u$, and d_t denotes the dimension of the text feature before fusion with the user features, $\otimes$ means concatenation operator.

$$h_{ij} = t_{ij} \otimes u_{ij}. \tag{16}$$

The combined features h_{ij} obtained from Eq. (16) are then fed into the DAN-Tree model as the substitute of the original t_{ij} so that the text features fused with user trustworthiness information are further fused with structural features. This process is formulated as Eq. (17), where $S_P \in \mathrm{R}^{d_p}$ is the new representation of the corresponding rumor which initially integrates the propagation strucutre, semantics of posts and users' profiles.

$$\begin{aligned} S_P = \text{DAN-Tree}(\{\{h_{11}, \cdots, h_{1|P_1|}\}, \cdots, \\ \{h_{i1}, \cdots, h_{i|P_i|}\}, \cdots, \{h_{|P|1}, \cdots, h_{|P||P_{|P|}|}\}\}). \end{aligned} \tag{17}$$

3.3.2 Global User Feature Encoding

Because in the rumor detection tasks, the simpler and usual method is to sort the posts by time series and then use LSTM or Transformer to obtain the post representation. We follow this idea and apply it to DAN-Tree++ to get a global representation of user features over time series, in order to further exploit the overall characteristics of user features in the rumor propagation process, the structure diagram of which is showed in Fig. 4.

Firstly, user features x_i are passed through a feature map described in Eq. (15) to obtain the user's representation u_i (Same as Fig. 3). Secondly, all u_i $(i = 1, \cdots, |E|)$ of the corresponding rumor are arranged in the temporal order of rumor posts. Then, these user features are passed through the Transformer encoder [10] and learned interactively over the time series to obtain the sequence result described in Eq. (18).

$$\left\{u_1', u_2', \cdots, u_n'\right\} = \mathrm{Trans}\left(\{u_1, u_2, \cdots, u_n\}\right). \tag{18}$$

Thirdly, in order to obtain the global user feature representation from the user feature sequence, we use the attention mechanism again to fuse the user feature sequence. In detail, the context vector c_i^u of user features is obtained by Eq. (19), where $a_u \in \mathrm{R}^{d_u \times 1}$ denotes the weight vector, $W_u' \in \mathrm{R}^{d_u \times d_u}$ denotes the weight matrix, and the activation function uses the $LeakyReLU$ function. After that, the normalized importance weights β_i^u are obtained by Eq. (20) and finally the global user feature representation $S_u \in \mathrm{R}^{d_u}$ is obtained by weighting

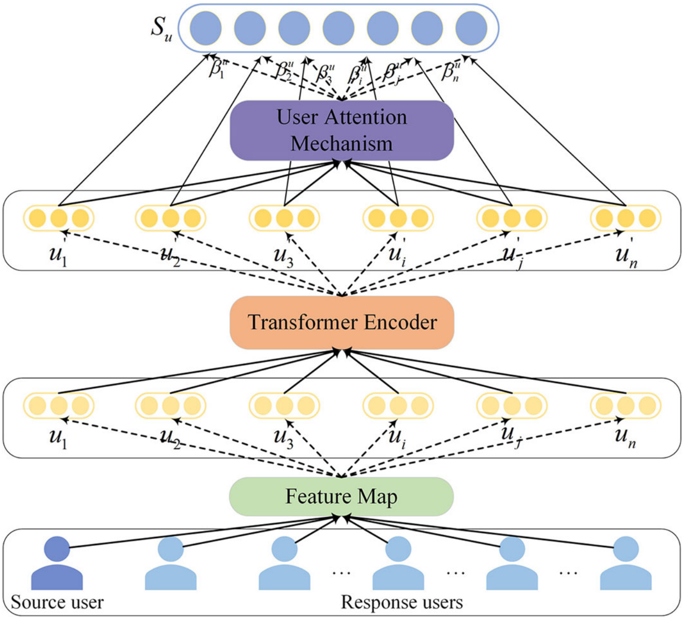

Fig. 4 The architecture of global user feature encoder

and summing all the user features in the sequence by Eq. (21).

$$c_i^u = a_u^T \cdot \text{LeakyReLU}\left(W_u' u_i'\right). \tag{19}$$

$$\beta_i^u = \frac{\exp\left(c_i^u\right)}{\sum_{k=1}^{n} \exp\left(c_k^u\right)}. \tag{20}$$

$$S_u = \sum_{i=1}^{n} \beta_i^u u_i'. \tag{21}$$

3.3.3 Rumor Classification

This section further combines the rumor representations S_p and S_u in the way of Eq. (22) for entirely fusing the information of propagation structure, and user features. After obtaining all rumor representations like $\tilde{r} \in \mathrm{R}^{d_r}$ with $d_r = d_p + d_u$, the rumors are also classified in the manner of Sect. 3.2.5 with the same loss function.

$$\tilde{r} = S_p \otimes S_u. \tag{22}$$

4 Experiments

4.1 Datasets

To evaluate the proposed model in this study, we use the following four classical rumor datasets, Twitter15 [6], Twitter16 [6], PHEME [7] and Weibo [8], to compare the performance of the DAN-Tree model with existing related models. In addition, we use Twitter15, Twitter16 with user data to further compare the performance of the DAN-Tree++ model.

Twitter15, Twitter16, PHEME, and Weibo contain 1490, 818, 4664, and 1972 rumor propagation tree structures, respectively, each consisting of retweeted posts and commented posts. Since the original datasets of Twitter15 and Twitter16 do not have publicly available comment texts, we re-crawled the comment texts using the Twitter API based on the publicly available comment post IDs, and removed some of the comment nodes that have been deleted or blocked due to deletion or blocking of Twitter Platform. Twitter15 and Twitter16 contain four types of tags: non-rumor (NR), false rumor (FR), true rumor (TR) and unverified rumor (UR). Weibo dataset contains only two tags: false rumor (FR) and true rumor (TR). PHEME dataset contains three tags: false rumor (FR), true rumor (TR), and unverified Rumor (UR). We mixed the five events in the PHEME dataset to form a dataset similar to Twitter15, Twitter16 and Weibo. What's more, we choose the top 5000 words with the highest frequency in the datasets for model training.

4.2 Experimental Setup

The experimental environment uses pytorch, the values of β_1 and β_2 in the Adam algorithm are set to 0.9 and 0.999, the initial value of learning rate is set to 0.01, the dropout probability is set to 0.5, the balance coefficient λ is set to 0.01, and the hidden layer dimension is set to 300. The word embedding in the model is initialized as a 300-dimensional word vector, and the word vector is kept in trainable mode during the model training process. The length of the fixed-length propagation path sequence N_P is set to 50, the number of N_t layers for post-level and path-lever attentions is set to 1, h in the multi-head attention is set to 4, d_u is set to 32, and the number of layers of the Transformer encoding block used in the global user feature encoding module is set to 1. The same experimental parameters are used for the Twitter15, Twitter16, and PHEME datasets. Since the average number of posts and the average number of propagation paths in the Weibo dataset are much higher than the other three datasets. Therefore, for the experiments on the Weibo dataset, we set h to 6 and N_P to 90 in multi-headed attention, and other parameters are set as in the other three datasets.

4.3 Evaluation Metrics and Baselines

4.3.1 Evaluation Metrics

From the task definition, rumor detection tasks belong to classification tasks under supervised learning. Therefore, research works in the field of rumor detection usually choose Accuracy, Precision, Recall and F1 value as evaluation metrics, and this choice will be followed in this study.

4.3.2 Baselines

We compare the performance differences between our model and existing classical models including traditional machine learning methods (DTR, DTC, RFC, SVM-RBF, SVM-TS, PTK) and propagation structure-based deep learning methods (RvNN, PLAN, Sta-PLAN, Bi-GCN), deep learning based methods further combined user features (PPC, UMLARD, HGARD). A brief description of the baseline models for comparison is given below.

- *DTR* [27] A decision tree based ranking model that identifies trending rumors by searching for query phrases.
- *DTC* [28] A decision tree based model that uses manually designed statistical features from posts to train decision trees for classification.
- *RFC* [29] A random forest classifier which uses manually selected features such as user, text and structure for classifier training.
- *SVM-RBF* [30] An SVM classifier with an RBF kernel, which also uses a set of manually designed statistical features.
- *SVM-TS* [31] An SVM classifier that models the change of feature values over time series.
- *PTK* [6] An SVM classifier with a propagation tree kernel, which captures the similarity between propagation tree structures for rumor classification through a kernel approach.
- *RvNN* [1] A top-down or bottom-up tree-structured recursive neural network for learning the propagation of rumors. The top-down network is selected since it has better performance.
- *RvNN** An improved version of RvNN by replacing Momentum Gradient Descent Algorithm with AdaGrad algorithm [32].
- *RvNN-GA* [33] An improved version of RvNN by using the global attention method for all nodes after recursive modeling.
- *PLAN* [2] A structure-aware hierarchical self-attention model by learning embedding vectors of propagation time series.
- *Sta-PLAN* [2] A variant of the PLAN model. It adds a variable to the PLAN model for describing the response relationship between posts.
- *Bi-GCN* [3] A novel bi-directional graph convolutional model by operating on both top-down and bottom-up propagation trees of rumors.
- *PPC* [22] A model that uses RNN and CNN structures on time series to obtain rumor representations by jointly modeling user features on the rumor propagation path.
- *UMLARD* [4] A model that learns the representation of users under different views in rumor propagation and connects them with text features to get the final representation of rumors.
- *HGARD* [5] A model that builds two subgraphs to incorporate user features into the learning of the global semantics of the text content.

4.4 Rumor Classification Performance

Tables 3, 4, 5, 6show the performance of our model and other baseline methods for rumor detection on four datasets, Twitter15, Twitter16, PHEME, and Weibo. We bold the optimal value of each evaluation metric in these Tables.

From the experimental results, the traditional methods using feature engineering (DTR, DTC, RFC, SVM-RBF, SVM-TS and PKT) did not work well enough on the Twitter15, Twitter16, and Weibo datasets, suggesting that the traditional methods lack the ability to extract higher-order representations from rumor data.

The deep learning method PLAN modeling the time-series features of rumors achieves good results among the baseline methods on the PHEME and Weibo datasets, indicating the importance of learning explicit temporal relationships between posts. Sta-PLAN coarsely exploits the structural information of posts based on the PLAN model and achieves better results than PLAN on the Twitter15 dataset.

Among the approaches that used rumor structure information, the RvNN model has the ability to learn the deep semantics of the propagation tree structure, and has improved in

Table 3 Experimental results on Twitter15

Method	Acc	F_1(NR)	F_1(FR)	F_1(TR)	F_1(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
PTK	0.667	0.619	0.669	0.772	0.645
RvNN	0.723	0.682	0.758	0.821	0.654
RvNN*	0.778	0.742	0.809	0.804	0.758
RvNN-GA	0.756	0.784	0.774	0.817	0.680
PLAN	0.845	0.823	0.858	0.895	0.802
Sta-PLAN	0.852	0.840	0.846	0.884	0.837
Bi-GCN	0.886	0.891	0.860	**0.930**	0.864
PPC	0.842	0.811	0.875	0.818	0.790
UMLARD	0.857	0.840	0.848	0.906	0.835
HGARD	0.892	0.915	0.897	0.907	0.845
DAN-Tree	0.902	0.891	**0.900**	**0.930**	**0.886**
DAN-Tree++	**0.909**	**0.943**	0.892	0.914	**0.886**

The bold value indicates the best result among all methods

Table 4 Experimental results on Twitter16

Method	Acc	F_1(NR)	F_1(FR)	F_1(TR)	F_1(UR)
DTR	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-RBF	0.321	0.423	0.085	0.419	0.037
SVM-TS	0.574	0.755	0.420	0.571	0.526
PTK	0.662	0.643	0.623	0.783	0.655
RvNN	0.737	0.662	0.743	0.835	0.708
RvNN*	0.788	0.763	0.778	0.853	0.761
RvNN-GA	0.764	0.708	0.753	0.840	0.738
PLAN	0.874	0.853	0.839	0.917	0.888
Sta-PLAN	0.868	0.826	0.833	0.927	0.888
Bi-GCN	0.880	0.847	0.869	0.937	0.865
PPC	0.863	0.820	**0.898**	0.843	0.837
UMLARD	0.901	**0.965**	0.855	**0.960**	0.822
HGARD	0.900	0.891	0.875	0.927	0.891
DAN-Tree	0.901	0.877	0.865	0.953	0.908
DAN-Tree++	**0.913**	0.927	0.868	0.927	**0.930**

The bold value indicates the best result among all methods

Table 5 Experimental results on PHEME

Method	Acc	Macro-F_1	F_1(FR)	F_1(TR)	F_1(UR)
RvNN	0.728	0.749	0.761	0.717	0.769
RvNN*	0.743	0.758	0.770	0.745	0.759
PLAN	0.785	0.772	0.753	0.828	0.735
Bi-GCN	0.722	0.677	0.570	0.792	0.675
DAN-Tree	**0.845**	**0.830**	**0.792**	**0.874**	**0823**

The bold value indicates the best result among all methods

Table 6 Experimental results on Weibo

Method	Acc.	Class	Prec.	Rec.	F_1
DTR	0.789	FR	0.784	0.801	0.793
		TR	0.794	0.777	0.785
DTC	0.831	FR	0.847	0.815	0.831
		TR	0.815	0.824	0.819
RFC	0.855	FR	0.810	0.929	0.866
		TR	0.916	0.779	0.842
SVM-RBF	0.879	FR	0.777	0.656	0.708
		TR	0.579	0.708	0.615
SVM-TS	0.885	FR	**0.950**	0.932	0.938
		TR	0.124	0.047	0.059
PTK	0.891	FR	0.876	0.913	0.894
		TR	0.907	0.868	0.887
RvNN	0.908	FR	0.912	0.897	0.905
		TR	0.904	0.918	0.911
RvNN*	0.929	FR	0.949	0.909	0.928
		TR	0.911	**0.950**	0.930
PLAN	0.943	FR	0.939	0.948	0.943
		TR	0.946	0.937	0.942
Bi-GCN	0.912	FR	0.913	0.904	0.897
		TR	0.903	0.910	0.894
DAN-Tree	**0.958**	FR	0.946	**0.972**	**0.958**
		TR	**0.972**	0.945	**0.958**

The bold value indicates the best result among all methods

experimental results compared to the PTK model. RvNN-GA model achieves better results by focusing on the different importance of different post nodes to the propagation tree representation based on RvNN. Bi-GCN model uses a bivariate graph convolutional network to focus on the rumor propagation and diffusion processes, achieving the best results among the propagation structure-based baselines on the Twitter15 and Twitter16 datasets.

Among the approaches that fuse user information, PPC model shows good detection results indicating that user features also play an important role in rumor detection tasks. UMLARD model and HGARD model which fuse user information in multiple ways show superior experimental results compared to PLAN and Bi-GCN models, verifying that the strategy of fusing user features has the effect of improving detection performance for methods that rely on text semantics and propagation structures.

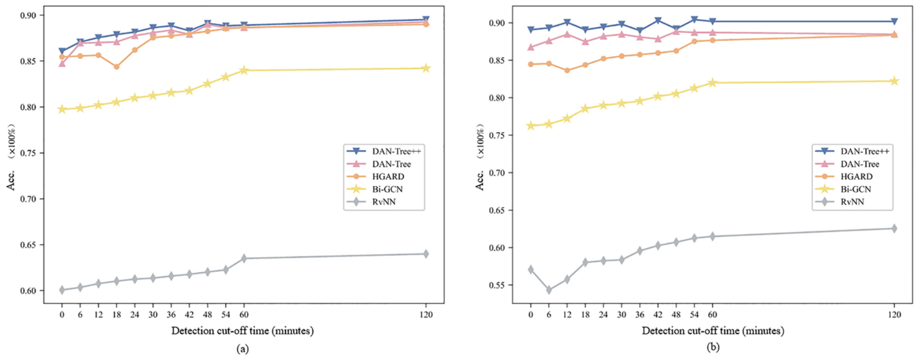

Fig. 5 **a** Result of Early Detection on Twitter15 dataset. **b** Result of Early Detection on Twitter16 dataset

Our proposed dual-attention model DAN-Tree based on rumor propagation tree structure uses Transformer structure to learn the implicit semantic relations of posts in the propagation paths, and uses attention mechanism to learn the post node attention and propagation path attention on propagation trees, which better captures the semantic information along the propagation tree structures. In addition, the path oversampling technique and the structural embedding method allow the model to better learn the deep structural information of the rumor propagation trees. Compared with the best results of the current existing work, the DAN-Tree model improves the accuracy on the Twitter15 and Twitter16 datasets from 88.6% and 88.0% to 90.2% and 90.1%, an improvement of 1.6% and 1.9%, and the F_1 on the PHEME dataset from 77.2% to 83.0%, an an improvement of 5.8%, and the accuracy on the Weibo dataset improved from 94.3% to 95.8%, an improvement of 1.5%. Meanwhile, the DAN-Tree++ model achieves 90.9% and 91.3% accuracy on the Twitter15 and Twitter16 datasets, which is 0.7% and 1.2% improvement compared to the experimental results of the DAN-Tree model. This indicates that the DAN-Tree++ model can effectively incorporate user features into the DAN-Tree model, which helps to boost rumor detection performance. Moreover, the DAN-Tree++ model achieves the best detection performance compared with all baseline models, validating the effectiveness of the DAN-Tree++ model for rumor detection tasks.

4.5 Early Detection

The goal of early detection is to identify rumors at the early stage when they start to spread, and is another important indicator of the comprehensive performance of rumor detection methods. This section compares the early experimental effects of DAN-Tree++, DAN-Tree and HGARD, Bi-GCN, and RvNN. Figure 5a and b show the early detection effects of the above models on the Twitter15 and Twitter16 datasets.

According to Fig. 5a and b, the DAN-Tree++ model proposed in this study outperforms the other models on the early detection tasks. As the cutoff time gradually increases from the 0 moment when the source post is just published, in 6-minute intervals, to 1 h, and then to the early stage when the rumor has spread for 2 h, the DAN-Tree++ model shows the best detection results at each cutoff time of rumor propagation. In particular, the DAN-Tree++ model achieves an accuracy of 90.1% on the Twitter16 dataset when using only 12 min of rumor data for training, outperforming the experimental results of other models using all

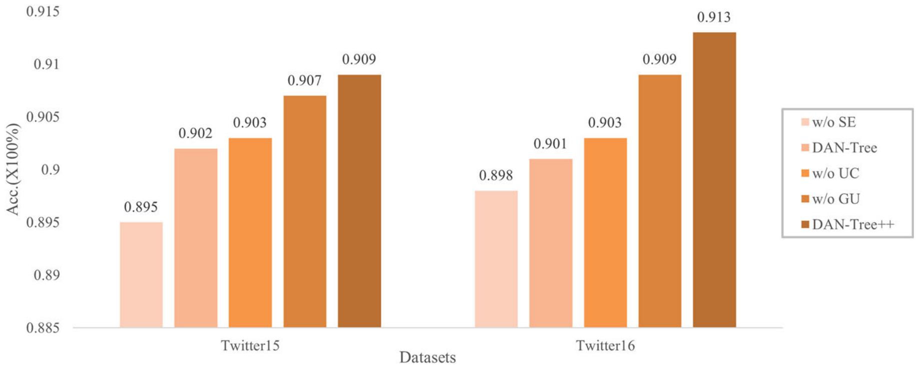

Fig. 6 Results of the ablation experiments on Twitter15 and Twitter16 datasets

data, demonstrating the DAN-Tree++ model's excellent ability to detect rumors at an early stage.

4.6 Ablations

In this section, a series of ablation experiments are conducted to analyze the impact of each module of the proposed model DAN-Tree and its extension DAN-Tree++ for the rumor detection tasks. The w/o SE represents the DAN-Tree model with the structural embedding "removed". w/o UC indicates that the text features are no longer additionally fused with user features in the propagation structure encoding module, i.e., the remaining module after the removal of User Credibility. w/o GU indicates the remaining module after removing the Global User Feature Encoder.

From the experimental results in Fig. 6, even with the structural embedding "removed", our model still achieves better results than all other baseline methods on the Twitter15, Twitter16 datasets. On the Twitter16 dataset, which has a shallow average depth, the structural embedding approach improves the accuracy by only 0.3%. However, for the Twitter15 dataset with deeper propagation trees, the structure embedding method improves 0.7%, indicating that the structure embedding method supplements the model with important spatial location information of post nodes in rumor propagation trees, compensates for the Transformer structure's insensitivity, and effectively utilizes the propagation structure information in the rumor propagation process, thus improving the effectiveness of the rumor detection task. In the w/o UC experiments, the detection effectiveness of the model decreases significantly on the Twitter15 and Twitter16 datasets, by 0.6% and 1.0%. This indicates that the credibility information learned from user features plays an important role in modeling the text features in rumor detection tasks.

5 Conclusions

In this study, we propose DAN-Tree, a dual-attention network on propagation tree structures. DAN-Tree utilizes Transformer encoding blocks as feature extractors to model the implicit relationships among posts on the propagation paths. It also further focuses on the features of key post nodes and key paths through post-level and path-level attention mechanisms. Based on the DAN-Tree model, we propose the rumor detection model DAN-Tree++ that fuses user features with propagation structures learned by DAN-Tree. DAN-Tree++ introduces user features in two aspects. One is to fuse user features with text features and introduce users' trustworthiness information. The other is to fuse global user features chronically in the rumor propagation process. We conducted a series of experiments on multiple real data sets. The experimental studies have shown that our model exhibits superior performance compared to other baseline models and achieves the best results on early detection tasks.

The following are a few perspectives for future work on the rumor detection tasks. (1) Building a larger scale dataset with richer features. (2) Exploring the role of external knowledge for the rumor detection task. (3) Introducing multi-task learning strategies.

Acknowledgements This work is supported in part by the National Key R&D Program of China (2018AAA0100302) and the National Nature Science Foundation of China (Nos. 61876016).

References

1. Ma J, Gao W, Wong K-F (2018) Rumor detection on Twitter with tree-structured recursive neural networks. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1, pp 1980–1989. https://doi.org/10.18653/v1/P18-1184
2. Khoo LMS, Chieu HL, Qian Z, Jiang J (2020) Interpretable rumor detection in microblogs by attending to user interactions. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8783–8790
3. 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
4. Chen X, Zhou F, Trajcevski G, Bonsangue M (2022) Multi-view learning with distinguishable feature fusion for rumor detection. Knowledge-Based Syst 240:108085. https://doi.org/10.1016/j.knosys.2021.108085
5. Huang Q, Yu J, Wu J, Wang B (2020) Heterogeneous graph attention networks for early detection of rumors on Twitter. In: 2020 international joint conference on neural networks (IJCNN), pp 1–8
6. Ma J, Gao W, Wong K-F (2017) Detect rumors in microblog posts using propagation structure via kernel learning. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1, pp 708–717. https://doi.org/10.18653/v1/P17-1066
7. Zubiaga A, Liakata M, Procter R, Wong Sak Hoi G, Tolmie P (2016) Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS One 11(3):0150989
8. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. IJCAI'16, pp 3818–3824
9. Chen T, Li X, Yin H, Zhang J (2018) Call attention to rumors: deep attention based recurrent neural networks for early rumor detection. Trends and applications in knowledge discovery and data mining. Springer, Cham, pp 40–52
10. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st international conference on neural information processing systems, Red Hook, pp 6000–6010
11. Nguyen X-P, Joty S, Hoi SC, Socher R (2020) Tree-structured attention with hierarchical accumulation. In: International conference on learning representations
12. Dai Z, Yang Z, Yang Y, Carbonell J, Le Q, Salakhutdinov R (2019) Transformer-XL: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 2978–2988. https://doi.org/10.18653/v1/P19-1285
13. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551
14. Zubiaga A, Aker A, Bontcheva K, Liakata M, Procter R (2018) Detection and resolution of rumours in social media: a survey. ACM Comput Surv. https://doi.org/10.1145/3161603
15. Wu K, Yang S, Zhu KQ (2015) False rumors detection on Sina Weibo by propagation structures. In: 2015 IEEE 31st international conference on data engineering, pp 651–662. https://doi.org/10.1109/ICDE.2015.7113322
16. Yang X, Ma H, Wang M, Xue X (2022) Rumor detection with bidirectional graph attention networks. Secur Commun Netw. https://doi.org/10.1155/2022/4840997

17. Zhang P, Ran H, Jia C, Li X, Han X (2021) A lightweight propagation path aggregating network with neural topic model for rumor detection. Neurocomputing 458:468–477. https://doi.org/10.1016/j.neucom.2021.06.062
18. Nan F, Ding R, Nallapati R, Xiang B (2019) Topic modeling with Wasserstein autoencoders. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 6345–6381. https://doi.org/10.18653/v1/P19-1640
19. Wei L, Hu D, Zhou W, Yue Z, Hu S (2021) Towards propagation uncertainty: Edge-enhanced Bayesian graph convolutional networks for rumor detection. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, pp 3845–3854 . https://doi.org/10.18653/v1/2021.acl-long.297
20. Lv Y, Sun X, Wen Y, Wang W (2022) Rumor detection based on time graph attention network. In: 2022 4th international conference on advances in computer technology, information science and communications (CTISC), pp 1–5. https://doi.org/10.1109/CTISC54888.2022.9849683
21. Chen W, Zhang Y, Yeo CK, Lau CT, Lee BS (2018) Unsupervised rumor detection based on users behaviors using neural networks. Pattern Recognit Lett 105:226–233. https://doi.org/10.1016/j.patrec.2017.10.014
22. Lu Y-J, Li C-T (2020) GCAN: graph-aware co-attention networks for explainable fake news detection on social media. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 505–514. https://doi.org/10.18653/v1/2020.acl-main.48
23. Bing C, Wu Y, Dong F, Xu S, Liu X, Sun S (2022) Dual co-attention-based multi-feature fusion method for rumor detection. Information 13(1):25
24. 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
25. Liu X, Zhao Z, Zhang Y, Liu C, Yang F (2022) Social network rumor detection method combining dual-attention mechanism with graph convolutional network. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2022.3184745
26. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3rd international conference on learning representations
27. 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. WWW '15, pp 1395–1405 . https://doi.org/10.1145/2736277.2741637
28. Castillo C, Mendoza M, Poblete B (2011) Information credibility on Twitter. In: Proceedings of the 20th International Conference on World Wide Web. WWW '11, pp 675–684. https://doi.org/10.1145/1963405.1963500
29. 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, pp 1103–1108. IEEE
30. 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. https://doi.org/10.1145/2350190.2350203
31. 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. CIKM '15, pp 1751–1754. https://doi.org/10.1145/2806416.2806607
32. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(61):2121–2159
33. Ma J, Gao W, Joty S, Wong K-F (2020) An attention-based rumor detection model with tree-structured recursive neural networks. ACM Trans Intell Syst Technol. https://doi.org/10.1145/3391250

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.