Multi-view learning with distinguishable feature fusion for rumor detection

Chen, X.; Zhou, F.; Trajcevski, G.; Bonsangue, M.M.

Knowledge-Based Systems 240 (2022) 108085

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-view learning with distinguishable feature fusion for rumor detection

Xueqin Chen [a,c], Fan Zhou [a,*], Goce Trajcevski [b], Marcello Bonsangue [c]

[a] University of Electronic Science and Technology of China, Chengdu, China
[b] Iowa State University, Ames, United States
[c] Leiden University, Leiden, The Netherlands

ARTICLE INFO

Article history:
Received 4 July 2021
Received in revised form 24 November 2021
Accepted 25 December 2021
Available online 4 January 2022

Keywords:
Rumor detection
Rumor spreading
User-aspect
Multi-view learning
Distinguishable

ABSTRACT

Researchers, enterprises, and governments have made great efforts to detect misinformation promptly and accurately. Traditional solutions either examine complicated hand-crafted features or rely heavily on the constructed credibility networks to extract useful indicators for discerning false information. However, such approaches require insightful domain expert knowledge and intensive feature engineering that are often non-generalizable. Recent advances in deep learning techniques have spurred learning high-level representations from textual and image content and discovering diffusion patterns with various neural networks. Despite the progress made by these methods, they still face the problem of overdependence on the content features and fail to discriminate against the influence of each user involved in the process of rumor spreading. Different user-aspect information plays different roles in various stages of rumor diffusion, effectively extract features from each aspect, and aggregate the learned features into a unique representation, which has not been well investigated. To address these limitations, we propose a novel model, UMLARD (User-aspect Multi-view Learning with Attention for Rumor Detection), to effectively learn the representation of different views of the users who engaged in spreading the tweet, and fuse the learned features through the distinguishable fusion mechanism. Finally, we concatenate the learned user-aspect features with content features to form a unique representation and feed it into a fully connected layer to predict the label of rumors. Our experiments conducted on real-world datasets demonstrate that UMLARD significantly improves the rumor detection performance compared to state-of-the-art baselines. It also allows explainability of the model behavior and the predicted results.

1. Introduction

The last decade has witnessed an emergence of numerous online social media (OSM) tools, such as Twitter, Facebook, Instagram, Reddit, Weibo, etc. These OSM gradually became the primary source of information in people's daily lives and have fundamentally changed our way of sharing information. However, OSM is a double-edged sword. On the one hand, they allow for social connectedness in a time of social distancing and facilitate the diffusion of knowledge in various contexts. On the other hand, they may induce the sharing of quick and superficial thoughts and the speedy diffusion of unverified facts such as rumors i.e., fake news and misinformation.[1] The explosive spread of rumors poses a threat to the internet credibility and has serious negative effects on individuals and the society — e.g., affecting national stability [1] and fairness of elections [2], and manipulating the stock market [3]. For example, numerous pieces of rumors related to COVID-19 pandemic circulating through mediums. A bizarre but notable recent example is the "corona-virus 5G conspiracy theory" [4], spreading the rumor that 5G networks generate radiation that triggers the virus, which has been peddled by conspiracy theorists and celebrities on OSMs in early March 2020. As a result, arsonists in the UK have launched over 70 arson attacks on phone masts. While a conspiracy theory and its various falsities and inaccuracies may be baseless, they still may lead to real-world harms. The severity of the impact of rumors spurs the need for effective detection of misinformation in OSMs and has encouraged many research works in the recent years [5,6].

* Corresponding author.
E-mail addresses: nedchen0728@gmail.com (X. Chen), fan.zhou@uestc.edu.cn (F. Zhou), gocet25@iastate.edu (G. Trajcevski), m.m.bonsangue@liacs.leidenuniv.nl (M. Bonsangue).

https://doi.org/10.1016/j.knosys.2021.108085
0950-7051/

[1] Despite the differences in the intentions of the spreaders, the term "fake news" has been used broadly for and interchangeably with "misinformation", "propaganda", "disinformation" and "rumors" in the community.

A series of existing studies have focused on automatically detecting rumors, mainly falling into three categories: (1) *Handcrafted features based approaches* mainly focus on identifying and incorporating complicated manual features for rumor detection, such as: lexical features [7–9], syntactic features, [7,10,11], visual features [12,13], user features [7,14,15], and network features [8, 16]. The respective performances highly depend on the effectiveness of extracted features which, however, require extensive domain knowledge and cannot generalize the features from one OSM to another. (2) *Credibility propagation-based approaches* [1, 12,17] aim to find the truth with conflicting information and leverage the inter-entity relations to identify the misinformation relying heavily on a constructed credibility network. However, most of the users spread rumors unintentionally, which increases the difficulties of rumor detection. In addition, the initial credibility values obtained from the feature-based classifiers make these kinds of methods face the same problems with feature-based methods. (3) *Deep learning-based approaches* have enhanced the ability on high-level representation learning, and can automatically learn sequential features [18,19], visual features [20], and structural features [21] from the rumor contents and propagation. Despite the significant progresses, current deep learning based methods still confront several limitations:

(L1) *Lack of systematic user-aspect rumor modeling*: Research [22] reveals that humans are the principal "culprits" in spreading false news. Existing studies either directly aggregate users' profile information as model inputs [7,23–25], which only pay attention to the local structure correlations among propagated users and the sequential propagation patterns [21] (i.e., user temporal features), or focus on learning the global structure of rumor diffusion [26] (i.e., user structural features). Essentially, these works learn the rumor representation from an event-level, and still lacks a unified framework that can learn rumor diffusion while extracting meaningful features from user-aspect.

(L2) *Entangled high-level feature learning*: Existing works learn high-level representations (e.g., structural or temporal) for rumor detection by exploiting user profiles or pre-trained textual features as model inputs. While improving detection performance, it is difficult to demonstrate the effectiveness and high-level representations of the model, because (1) the learned representations are entangled with the original input [25], and (2) the models use the same input [27] to learn different high-level features.

(L3) *Indistinguishable importance of features and users*: Different features play different roles in rumor detection at different phases of propagation. As information spreads, for example, the effect of structural information and temporal information on discriminating rumors becomes different [23,28]. Also, users may either unconsciously forward some unproven news, or deliberately propagate the fake news in the information spread [14]. Understanding the efficacy of features and individual users would help detect rumors, which, however, has not been well investigated in existing studies.

(L4) *Limited interpretability*: Most existing studies focus on explaining the news content, e.g., discover the important sentence in the articles or emotional words in comments [29], to interpret the detection results. However, these works cannot explain critical features beyond text and determine user's roles in rumor propagation.

To address the limitations L1–L4 above, we propose a new model UMLARD — User-aspect Multi-view Learning with Attention for Rumor Detection. Multi-view learning is a promising learning paradigm that jointly models different views of the same input data for improving learning performance [30]. For example, a web page can be described in forms of text, video, and image [31] simultaneously. By exploring the complementarity and consistency of different views, it can further improve the model performance [32]. Inspired by recent progress in multi-view learning [33–35], we initiate the attempts to capture the principal characteristics of users and rumors by learning multiple distinct features. Specifically, we exploit different views to represent an instance for comprehensively describing the information of the instance. We first abstract the user-aspect features of the users engaged in the diffusion process as user profile-view, user structural-view, and user temporal-view, and then incorporate different views to predict the credibility of the given information. Specifically, UMLARD exploits different embedding methods to learn the view-specific high-level representations of a given post from the hierarchical diffusion process and user profiles. To understand the importance of each view and the role of the user, UMLARD employs a view-wise attention network and a capsule attention network to incorporate both view-level and user-level features. It allows us to better discriminate feature influence and the effect of user behaviors in spreading rumors.

Our main contributions towards rumor detection problem provide:

- **User-aspect feature extraction (L1)**: We conceptualize user-aspect features as different views, including profile-view, structural-view, and temporal-view, and present a novel model to learn different views for each user who engaged in the information diffusion.
- **View-specific embedding methods (L2)**: UMLARD utilizes different embedding methods to learn view-specific high-level representations based on different inputs: (1) an attention-based layer aims to learn user profile-view by assigning different importance to features in user profiles; (2) an improved GCN-based network to learn structural-view from the diffusion network while considering the direction of information dissemination, taking the adjacency matrices of diffusion networks as input; and (3) a time-decay LSTM considers the influence of users and is used for temporal-view learning based on the diffusion path taking two types of embeddings as inputs, i.e., static-embedding and dynamic-embedding.
- **Distinguishable hierarchical feature fusion (L3)**: We design a hierarchical feature fusion mechanism to unify the knowledge from different perspectives, which consists of two components: (1) a view-wise attention layer to capture the features from different views; and (2) a capsule attention layer to differentiate the most related users.
- **Explainable prediction results (L4)**: UMLARD explains the significance of features according to the learned attention values. Specifically: (1) the dimensional-wise attention network shows the importance of different characteristics in the user profiles; (2) the view-wise attention results tell how the users play different roles in different phases of rumor propagation; and (3) from the capsule attention results, one can easily understand which users play critical roles in detecting the rumors.

We conduct extensive evaluations on three benchmark datasets. The results demonstrate that UMLARD significantly outperforms the state-of-the-art baselines while providing intuitive explanations on both model behavior and detection results.

2. Related work

The problem of rumor (or fake news/information, misinformation) detection is an important research topic in recent social media studies and receives increased attention in various disciplines including politics [2], finance [36], marketing [37], healthcare [13], etc. "Rumor" is usually defined as a misleading

story or misinterpret of information, circulating among communities and pertaining to an object, event, or issue in public concern [22]. Existing methods for rumor detection generally fall into three categories, i.e., feature-based, credibility-based, and deep learning-based approaches.

2.1. Hand-crafted features-based approaches

Most of earlier works extracted various hand-crafted features from raw data, which can be typically summarized as two types: (1) content features extracted from both text (e.g. characters, words, sentences and documents) and visual elements (e.g. images and videos), which can be further partitioned as lexical features [7–9], syntactic features [7,10,11], topic features [38], visual statistical features [12,13], and visual content features [39]; and (2) social context features extracted from the user behavior and the diffusion network, which reflect the relationship among users and describe the diffusion process of a rumor, including user features [7,14,15], propagation features [8,28,38], and temporal features [8,23]. After feature engineering, the selected features are used in discriminative machine learning algorithms (e.g., random forest, naive Bayes, and support vector machines) to classify the news or tweets.

Rumors aim to arouse much attention and stimulate the public mood. Therefore, their texts/images/videos tend to have certain patterns in contrast to truth. Zhao et al. [9] discovered two types of language patterns in rumors, i.e., inquiry and correction patterns, and detected the patterns of rumor messages through supervised feature selection on a set of labeled messages. Wu et al. [38] defined a set of topic features to summarize semantics and trained a Latent Dirichlet Allocation (LDA) model for detecting rumors on Weibo. Towards a more comprehensive understanding of the text on social media, existing works also come up with textual features derived from social media platforms, apart from general textual features, such as source links [13] and emotions [7]. As for visual content features, Jin et al. [39] found that images in rumors and non-rumors are visually distinctive on their distributions and propose five visual features to measure the rumors, i.e., visual clarity score, visual coherence score, visual similarity distribution histogram, visual diversity score, and visual clustering score. Social context features are derived from the social connection characteristics of social media. Rumors are usually created by a few users and spread by a large number of users. Therefore, user profiles are commonly used to measure the user's characteristics and credibility. For example, Castillo et al. [7] first identified the credibility of tweets in Twitter, and Kwon et al. [8] extended it by proposing 15 structural features extracted from the diffusion network and the user friendship network. In the work [23], the authors proposed a method for discretizing time and capturing the variation of temporal features associated with rumors.

However, the performance of feature-based approaches heavily depends on the hand-craft features, which lacks a standard and systematic way to design general features across platforms and to deal with different types of rumors. In fact, the conclusions of existing works usually contradict each other, primarily due to the differences between different types of datasets. For example, Yang et al. [24] designed a set of features (e.g., client-based features and location-based features) based on Weibo, whose users are mainly restricted to China. It is therefore difficult to use these features for detecting rumors spread on Twitter and Facebook due to the differences in languages, clients' and users' geographic distributions, etc.

2.2. Credibility propagation-based approaches

Inspired by the work of truth discovery that aims to find truth with conflicting information, this line of approaches consists of two main steps, i.e., (1) credibility network construction and (2) credibility propagation. The underlying assumption of these approaches is that the credibility of news is highly related to the reliability of relevant social media posts, and both homogeneous and heterogeneous credibility networks can be built for the propagation process. Homogeneous credibility networks consist of a single type of entities, such as posts and events. In contrast, heterogeneous credibility networks involve different types of entities, such as posts, sub-events, and events. Gupta et al. [12] first introduced a PageRank-like credibility propagation algorithm by encoding users' credibility and tweets' implications on a user-tweet-event information network. Inspired by the idea of linking entities altogether and leveraging inter-entity connections for credibility propagation, Jin et al. [17] proposed a three-layer hierarchical credibility network, which includes news aspects and utilizes a graph optimization framework to infer event credibility. The work in [1] found that relations between messages on microblogs (i.e. support and oppose) are crucial for evaluating the truthfulness of news events, and built a homogeneous credibility network among tweets to guide the process of credibility evaluation. While comparing with direct classification on the individual entity, credibility propagation-based approaches may leverage the inter-entity relations for robust detection results. However, the performance of these methods strongly relies on the constructed credibility network.

2.3. Deep learning-based approaches

The recent success of deep learning in NLP (Natural Language Processing) and CV (Computer Vision) communities spurs a lot of deep rumor detection methods. These models have shown improved performance over traditional approaches due to their enhanced ability to automatically representation learning. Most existing deep learning-based approaches are content-aware that mainly focused on extracting textual features [18,29] and visual features [20,40] from news content, user comments, and images, etc. Ma et al. [18] first proposed a recurrent neural network (RNN)-based model to learn temporal and textual features from news content via modeling the posts in an event as time series data. Shu et al. [29] proposed a co-attention network to exploit both news content and user comments for rumor detection while discovering explainable sentences. Jin et al. [20] presented a model to extract the visual, textual, and social context features, which are fused by the attention mechanism. Moreover, researchers also employed other deep learning techniques, such as multi-task learning [41], adversarial learning [42], and knowledge enhancement [43], to learn more robust content-aware features for rumor detection. However, rumors are intentionally written by mimicking real news [44], which makes content-aware methods hard to further improve detection performance due to the lack of necessary domain knowledge.

Recently, a few works exploited diffusion patterns in news spreading for rumor detection, e.g., temporal features [21,27,45] and structural features [19,25–27]. For example, Liu et al. [21] presented a time series classifier with RNN and CNN to predict whether a given news story is fake at an early stage, taking common user characteristics and propagation paths into consideration. Song et al. [45] proposed a temporal propagation-based model that can distinguish rumors from true news through modeling dynamic evolution patterns of news. As for the structural features, Ma et al. [19] presented a tree-structured RNN to catch the hidden representations from both propagation structures and

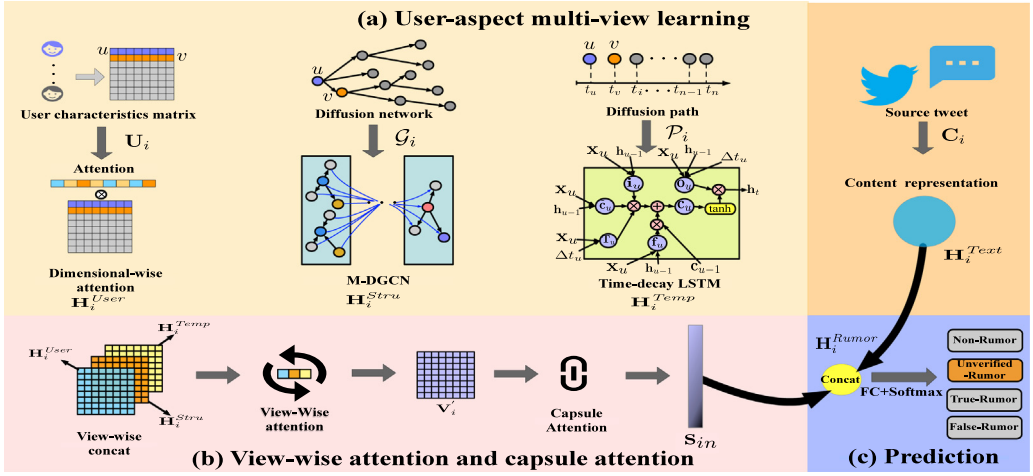


Fig. 1. An overview of UMLARD. (a) The inputs of UMLARD are the observed diffusion network, the diffusion path, the user characteristic matrix, and the content of the source tweet. It uses a dimensional-wise attention layer, a multi-layer diffusion graph convolutional network (M-DGCN), and a time-decay LSTM to learn the latent representations from the three kinds of inputs, respectively. (b) It learns to discriminate the role of three-views and the importance of users in identifying misinformation. (c) Finally, we concatenate the learned features with text content to perform classification.

text contents. Inspired by the success of graph neural networks in information cascades modeling [46,47], Bian et al. [26] proposed a graph convolutional network (GCN) [48]-based model that can learn global structural relationships of rumor dispersion. He et al. [49] improved the work [26] by using event augmentation and contrastive learning.

Recently, more studies considered both structural and temporal features for rumor detection. Chen et al. [27] introduced a hierarchical diffusion modeling model by extracting both temporal features and propagation structures from the microscopic diffusion and macroscopic diffusion jointly. In addition, some researchers realized that users play significant roles in rumor spreading. For example, Chen et al. [50] extracted social homophily, influence, and susceptibility of users from the user interaction network for rumor detection. Dou et al. [51] proposed a user preference-aware rumor detection model to learn user endogenous preference and exogenous context from users' historical posts and reply network, respectively.

While the above methods achieved enhanced performance over the content-aware approaches, they still suffer from some limitations, e.g., learning rumor representation at the event-level rather than user-level, and the learned high-level representations are entangled with input features, as well as inefficient feature fusion, which are systematically addressed in our proposed UMLARD model.

3. UMLARD: Preliminaries and methodology

In this section, we first introduce the preliminaries and basic notations, and then formalize the problem studied in this paper. Subsequently, we present the details of the proposed UMLARD framework.

As illustrated in Fig. 1, UMLARD consists of three main components: (1) *Representation learning layer* that simultaneously extracts user-aspect features from the profile-view, structural-view, and temporal-view, while embedding the source tweet content into low-dimensional space; (2) *Hierarchical fusion layer* that fuses the learned representation at both view-level and user-level; and (3) *Rumor detection layer* that makes use of a fully connected layer to predict the labels of tweets, based on the learned user-aspect knowledge and tweet content.

Table 1
List of notations.

Symbol	Description
$\mathcal{M}, M_i$	a set of tweets/posts and a specific tweet/post.
$\mathcal{G}_i, U_i, E_i$	diffusion network, user set and edge set of tweet M_i.
$\mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i$	diffusion path, user characteristic matrix, source tweet content vector of tweet M_i.
$T, t_*, \lvert U_i\rvert$	the observation window, time-stamp for each user and the number of users in tweet M_i.
$\mathbf{p}_*, \mathbf{g}_*, \mathbf{e}^s_*, \mathbf{e}^d_*$	the user profile vector, pre-trained node embedding, static embedding and dynamic embedding of each user.
$d_{user}, d_{stru}, d_{temp}, d_{word}, d_{view}$	the hidden size of the profile-view, structural-view, temporal-view, word embedding and multi-view layer.
$\mathbf{H}_i^{User}, \mathbf{H}_i^{Stru}, \mathbf{H}_i^{Temp}, \mathbf{H}_i^{Text}$	the representations of the profile-view, structural-view, temporal-view and content feature, respectively.
$\mathbf{V}_i^{'}, \mathbf{s}_{in}$	the representation after view-wise attention and capsule attention for tweet M_i.
$\mathbf{H}_i^{Rumor}$	the final representation of tweet M_i.
$\hat{Y}/\hat{\mathbf{y}}_*, Y/\mathbf{y}_*$	the predicted label and the ground truth.

3.1. Preliminaries and problem definition

Suppose we have a set of tweets $\mathcal{M} = \{M_i, i \in [1, |\mathcal{M}|]\}$, where each tweet M_i is a quadruplet representing the corresponding diffusion process and the users enrolled: $M_i = \{\mathcal{G}_i, \mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i\}$, where $\mathcal{G}_i, \mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i$ are diffusion network, diffusion path, user characteristic matrix and the content vector of source tweet, respectively. We now describe each component of M_i.

Diffusion Network. *A diffusion network for tweet M_i is a graph $\mathcal{G}_i = \{U_i, E_i\}$, where U_i is the set of nodes and $E_i \subset U_i \times U_i$ is a set of edges. A node $u_{ij} \in U_i$ represents a user, and a directed edge $u_{ij} \rightarrow u_{ik} \in E_i$ represents the relationship that u_{ik} retweeted the tweet received from u_{ij}. Note that, $\mathcal{G}_i$ is directed acyclic graph.*

Diffusion Path. *A diffusion path of tweet M_i is defined as a multivariate time series $\mathcal{P}_i = \{(u_{i1}, t_{i1}), \ldots, (u_{i|U_i|}, t_{i|U_i|})\}$, where $t_{i1} \leq t_{i2} \leq \cdots < t_{i|U_i|}$. Each pair (u_{ij}, t_{ij}) indicates that the user u_{ij} retweets the source tweet at time t_{ij}. In the case that $t_{ij} = t_{im} (j \neq m)$, the order in the sequence of $\mathcal{P}_i$ is determined based on the ordering*

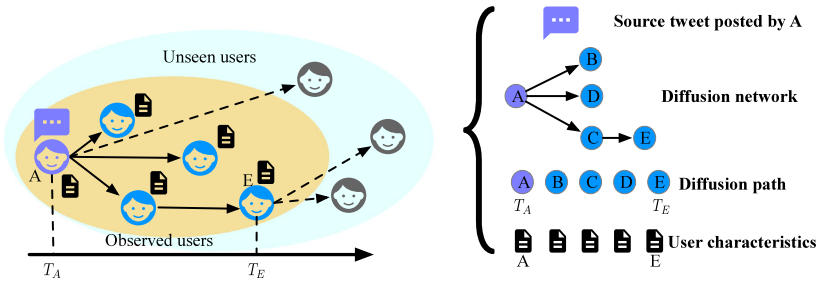


Fig. 2. An example of the extracted information from a tweet diffusion.

of the user IDs (which are assumed unique). The first user u_{i1} denotes the source user (i.e., the one who initiated the tweet at t_{i1}), and the rest of the users $u_{ij}, j \in [2, |U_i|]$ are users participating in spreading the information.

The concepts of diffusion network and diffusion path are illustrated in the right-hand side portion of Fig. 2 for the corresponding example. Even though both diffusion graph and diffusion path are abstracted from the diffusion thread of tweets, they are independent and different. Specifically, the diffusion graph reflects the direction of message passing between users, while the diffusion path reflects the time and sequential information of user engagement. Fig. 2 illustrates two important components, which will be explain in more details.

User Characteristic Matrix. *Each user $u_{ij} \in U_i$ is associated with a user vector $\mathbf{p}_{ij} \in \mathbb{R}^{d_{user}}$, which is extracted from users' profiles — e.g., screen name, description, etc. We concatenate the user vectors for all users that share the given tweet to form the user characteristic matrix $\mathbf{U}_i \in \mathbb{R}^{|U_i| \times d_{user}}$, in which each row corresponds to a user and the users are ranked in chronological order according to the respective retweet times.*

Tweet Content. *For a tweet M_i, the text content $\mathbf{C}_i$ is considered to be a sequence of words — i.e., $\mathbf{C}_i = [\mathbf{w}_{i1}, \mathbf{w}_{i2}, \ldots, \mathbf{w}_{iL}] \in \mathbb{R}^{L \times d_{word}}$, where L is the number of words in source tweet.*

We note that each word is represented by a d_{word}-dimension vector using a particular word embedding technique, e.g., word2vec.

We summarize the (definitions of the) symbols used in the paper in Table 1. We note that, in the sequel, whenever there is no ambiguity, we may omit the double-subscript from the notation (i.e., whenever we are unambiguously working with one specific tweet M_i, we may drop i from the sequences denoting users, time-stamps, etc.).

We now formally define the rumor detection problem that we study as follows:

Definition 1 (*Rumor Detection.*)**.** Given a tweet $M_i = \{\mathcal{G}_i, \mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i\}$ within an observation window T, our rumor detection goal is to learn a function f from labeled claims, i.e., $f(\hat{\mathbf{y}}_i | \mathcal{G}_i, \mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i; T)$, where the predicted result $\hat{\mathbf{y}}_i$ takes one of the four finer-grained classes: non-rumor, false rumor, true rumor, and unverified rumor (as introduced in [28]).

3.2. Learning users profile-view

User profiles have been demonstrated to be strong indicators when detecting fake news [14,15]. The user profile characteristics are either explicit (e.g., username and geolocations) or implicit (e.g., gender and age). However, accessing the implicit features may not always be feasible due to the privacy concerns of many OSMs. Therefore, we consider the following eight explicit features, grouped in two major categories, which can be typically accessed in most OSMs:

- **Profile-Related features** include five basic user description fields: the screen name that the user identify herself; the user's self description; the attribute indicating whether the account has been verified by the platform; the geographical location of the user; and the UTC time that the user account was created on the social platform.
- **Influence-Related features** include three attributes describing user activities and social relations: the number of posts issued by the user, the number of followers, and the mutual follower-ship.

For each user u_j in a tweet M_i, we concatenate the profile characteristics into one feature vector, and then form the user characteristic matrix $\mathbf{U}_i \in \mathbb{R}^{|U_i| \times d_{user}}$ by concatenating all user vectors for the users involved in spreading the tweet.

To provide explanations on which characteristics are useful for rumor detection, we design a dimensional-wise attention layer to assign weights to each dimension of user profiles. Its aim is to learn how to discriminate the importance of different characteristics. First, we expand $\mathbf{U}_i$ as a sequence of 1-dimensional "channels" for the features, i.e., $\mathbf{U}_i \in \mathbb{R}^{|U_i| \times 1 \times d_{user}}$, where $|U_i|$, 1 and d_{user} can be regarded as the height, width and channel of an image (similarly to the channels for each of the primitive colors – red, green and blue – in image processing). Then, we use a global average pooling (GAP) to aggregate the global information into a dimensional-wise descriptor $\mathbf{z} \in \mathbb{R}^{d_{user}}$, where $\mathbf{z} = \frac{1}{|U_i| \times 1} \sum_{h=1, w=1}^{|U_i|, 1} \mathbf{U}_i(h, w)$. To capture the dimensional-wise dependencies, we employ two fully connected layers with non-linearity — i.e., dimensionality-reduction layer and dimensionality-increasing layer:

$$\begin{aligned} \mathbf{f}_{red} &= \tanh(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1), \\ \mathbf{f}_{inc} &= \text{softmax}(\mathbf{W}_2 \mathbf{f}_{red} + \mathbf{b}_2), \end{aligned} \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{\frac{d_{user}}{r} \times d_{user}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{user} \times \frac{d_{user}}{r}}$ are parameter matrices, $\mathbf{b}_1 \in \mathbb{R}^{\frac{d_{user}}{r}}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_{user}}$ are biases, and r is the reduction ratio. Thus, the final output of the user profile-view becomes:

$$\mathbf{H}_i^{User} = \mathbf{U}_i \mathbf{f}_{inc} + \mathbf{U}_i, \tag{2}$$

where $\mathbf{H}_i^{User} \in \mathbb{R}^{|U_i| \times d_{user}}$, $\mathbf{f}_{inc}$ denotes the attention score allocating different importance to each dimension of $\mathbf{U}_i$ through the multiplication operation, i.e., $\mathbf{U}_i \mathbf{f}_{inc}$. The operation of plus $\mathbf{U}_i$ is borrowed from idea of skip connections [52].

The objective of dimensional-wise attention layer is to obtain a new user characteristic matrix through correlation training between the user profile's different characteristics by assigning different dimensions of the matrix with the different weights during training the model. In general, the contributing characteristics would be strengthened. Since the trivial characteristics should be weakened, we can also reduce the noise brought by non-critical characteristics, thereby improving the accuracy of the detection task. This effect is especially valuable for early-stage rumor detection. For example, when the number of participating users and

the corresponding profiles are limited, it is particularly important to encourage the fundamental characteristics to explain rumor identification decisions. We will provide visual explanations in Section 4.

3.3. Learning users structural-view

The structural information of users who participate in spreading a tweet is extracted from the diffusion graph, which aims to capture the degree of connection, similarity, distance, and even community, etc., between users [44]. Inspired by the recent successes of network representation learning methods in processing graph-structured data [48,53–55], we define a multi-layer diffusion graph convolutional network (M-DGCN) as user structural-view encoder, in which the propagation rule of diffusion convolutional network is defined as:

$$\mathbf{H}^{(l+1)} = \sigma((\theta_O(\mathbf{D}_O^{-1}\mathbf{A}) + \theta_I(\mathbf{D}_I^{-1}\mathbf{A}^T))\mathbf{H}^{(l)}), \tag{3}$$

where θ_O and θ_I are filter parameters; $\mathbf{D}_O^{-1}\mathbf{A}$ and $\mathbf{D}_I^{-1}\mathbf{A}^T$ are transition matrices of the forward diffusion process and the reverse one, respectively – $\mathbf{D}_O$ and $\mathbf{D}_I$ represent out-degree diagonal matrix and in-degree diagonal matrix, respectively; $\sigma(\cdot)$ denotes activation function, i.e., ReLU$(\cdot)$ here; $\mathbf{H}^{(l)} \in \mathbb{R}^{|U_i|\times F}$ is the matrix of activation in the lth layer – $|U_i|$ is the number of users in the diffusion network and F is the dimension of the output. The difference between our M-DGCN and previous graph convolutional network [48,53] is that the Chebyshev kernel in M-DGCN is equal to 1, whereas we stack a couple of such layers to aggregate the information from the distant nodes rather than the K-localized convolutions. In this layer, the initial input $\mathbf{H}^{(0)}$ is obtained from a pre-trained network embedding layer which maps a user u_j to its D-dimensional representation $\mathbf{g}_j \in \mathbb{R}^D$, which allows the varying-size diffusion networks learning.

In order to reduce over-fitting for diffusion convolutional network, we employed a recently developed technique *DropEdge* (cf. [56]) for robust structural-view learning. That is, we randomly drop edges from the input diffusion trees to generate different copies with a certain ratio in each training epoch. More specifically, suppose the total number of edges in the diffusion tree is $|E_i|$ and the dropping rate is r_{drop}. The adjacency matrix after dropout is computed as $\hat{\mathbf{A}} = \mathbf{A} - \mathbf{A}_{drop}$, where $\mathbf{A}_{drop}$ is the matrix constructed using $|E_i| \times r_{drop}$ edges randomly sampled from the original edge set E_i. After the diffusion convolutional layer, the diffusion network $\mathcal{G}_i$ is represented as a vector matrix $\mathbf{H}_i^{Stru} \in \mathbb{R}^{|U_i|\times d_{stru}}$.

The structural-view $\mathbf{H}_i^{Stru}$ learned through M-DGCN represents the role of a node (i.e., a user) in the information spreading. M-DGCN not only models the propagation direction of information between spreaders but also aggregates high-order structural details, including the cascade virality, spreading patterns, etc., which may facilitate the rumor identification. We note that in [28] it has been demonstrated that the rumors have similar propagation patterns.

3.4. Learning users temporal-view

Users' engagement time and the sequential patterns of retweets also play an essential role in detecting rumors [18,19,21]. We capture this view of users based on the diffusion path. Each user in the diffusion path would be assigned two types of embeddings: a static-embedding and a dynamic-embedding.

- **Static-embedding** refers to the relative position j ($1 \le j \le |U_i|$) for each user u_j in the sequence. We encode this information based on the chronological order of retweet times, and the users with the same retweet time will have the same position embedding. Inspired by the self-attention [57], we obtain the static-embedding $\mathbf{e}_j^s$ using a positional-encoding technique based on sine and cosine functions of frequencies:

$$\mathbf{PE}(j)_{2d} = \sin(j/10000^{2d/d_e}),$$
$$\mathbf{PE}(j)_{2d+1} = \cos(j/10000^{2d/d_e})$$

  where d_e is an adjustable dimension and $1 \le d \le d_e/2$ denotes the dimension index in $\mathbf{e}_j^s$. The basic idea of this choice is to allow the model attending the relative position of the users. For details of this formula, refer to [57].
- **Dynamic-embedding** initializes user representations as one-hot vector $\mathbf{q} \in \mathbb{R}^N$, where N denotes the total number of users in the dataset. All users are associated with a specific embedding matrix $\mathbf{E} \in \mathbb{R}^{N\times d_e}$, where d_e is an adjustable dimension. Matrix $\mathbf{E}$ converts each user u_j into a unique representation vector as $\mathbf{e}_j^d = \mathbf{qE}$, $\mathbf{e}_j^d \in \mathbb{R}^{d_e}$. In this way, the user embedding matrix $\mathbf{E}$ can be learned during training, supervised by the downstream task, i.e., rumor detection in this work.

Subsequently, we use an RNN model (e.g., LSTM [58]) to learn the temporal dependence of the diffusion. However, the influence of retweet users will diminish over time, and the "vanilla LSTM" is not capable of capturing this time-decay effect of information diffusion. To address this issue, we introduce a time-gate inspired by [59] into the LSTM.

The time-gate not only controls the influence of $\mathbf{x}_j$ – the combination of static and dynamic embeddings – on the current step, but also caches the time interval between consecutive retweets to model the time-decay effect. Specifically, a time-decay LSTM unit takes: $\mathbf{x}_j$, previous hidden state $\mathbf{h}_{j-1}$, and time interval Δ_{t_j} as inputs — and outputs the current hidden state $\mathbf{h}_j$ using:

$$\begin{aligned}
\mathbf{x}_j &= \mathbf{e}_j^s + \mathbf{e}_j^d, \\
\mathbf{i}_j &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_j + \mathbf{U}_{hi}\mathbf{h}_{j-1} + \mathbf{b}_i\right), \\
\mathbf{f}_j &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_j + \mathbf{U}_{hf}\mathbf{h}_{j-1} + \mathbf{b}_f\right), \\
\mathbf{T}_j &= \sigma\left(\mathbf{W}_{xT}\mathbf{x}_j + \tanh\left(\mathbf{W}_{tt}\Delta t_j\right) + \mathbf{b}_T\right), \\
\mathbf{o}_j &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_j + \mathbf{U}_{ho}\mathbf{h}_{j-1} + \mathbf{W}_{to}\Delta t_j + \mathbf{b}_o\right), \\
\widetilde{\mathbf{c}}_j &= \tanh\left(\mathbf{W}_{xz}\mathbf{x}_j + \mathbf{U}_{hz}\mathbf{h}_{j-1} + \mathbf{b}_z\right),
\end{aligned} \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function; $\mathbf{i}_j, \mathbf{f}_j, \mathbf{T}_j, \mathbf{o}_j, \widetilde{\mathbf{c}}_j, \mathbf{b}_*$ are the input gate, forget gate, time gate, output gate, new candidate vector and bias vector, respectively. The matrices $\mathbf{W}_{x*} \in \mathbb{R}^{d_e\times d_{temp}}$, $\mathbf{W}_{t*} \in \mathbb{R}^{1\times d_{temp}}$ and $\mathbf{U}_{h*} \in \mathbb{R}^{d_h\times d_{temp}}$ represent the different gate parameters. In particular, the memory cell $\mathbf{c}_j$ is updated by replacing the existing memory unit with a new cell $\mathbf{c}_j$ as:

$$\mathbf{c}_j = \mathbf{f}_j \odot \mathbf{c}_{j-1} + \mathbf{i}_j \odot \mathbf{T}_j \odot \widetilde{\mathbf{c}}_j, \tag{5}$$

where $\odot$ denotes the element-wise multiplication. The hidden state is then updated by:

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh\left(\mathbf{c}_j\right), \tag{6}$$

Finally, the representation vector for the temporal-view is $\mathbf{H}_i^{Temp} = \left\{\mathbf{h}_1^{Temp}, \mathbf{h}_2^{Temp}, \ldots, \mathbf{h}_{|U_i|}^{Temp}\right\}$, where $\mathbf{H}_i^{Temp} \in \mathbb{R}^{|U_i|\times d_{temp}}$. Note that the temporal-view of the user obtained by the time-decay LSTM reflects each user's influence on the subsequent participators in the message diffusion.

3.5. View-wise attention for view-level feature fusion

After obtaining the latent representation for each view, we need to fuse the multi-view features. Rather than directly concatenating different aspects, as often done in the existing solutions [20,60,61], we present a method to capture the differences

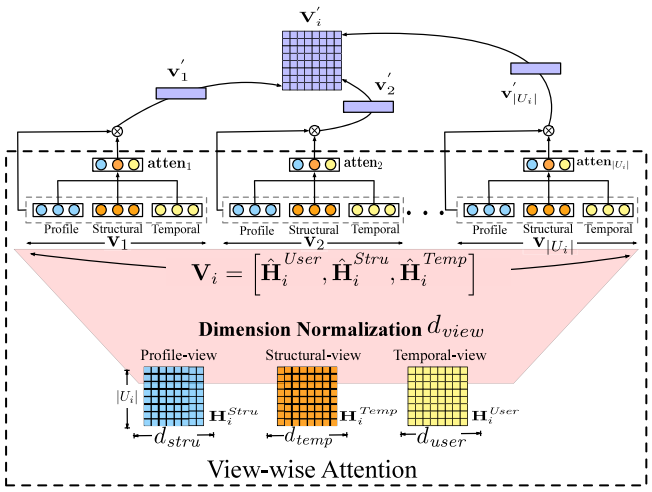


Fig. 3. Illustration of view-wise attention.

between different views. The primary motivation stems from the observation that various views are not equally relevant in the task of rumor identification. Towards that, we propose a view-wise attention layer to prioritize the fundamental views for each user. As depicted in Fig. 3, the view-wise attention layer takes profile-view, structural-view, and temporal-view as input and generates the attention score for each view at the user-level. Specifically, it first normalizes the dimensions of the three views' vectors as d_{view} via a fully connected layer. Let $\mathbf{V}_i = \left[\hat{\mathbf{H}}_i^{User}, \hat{\mathbf{H}}_i^{Stru}, \hat{\mathbf{H}}_i^{Temp}\right]$ denote the feature set after dimension normalization. Each vector $\mathbf{v}_j = \left[\hat{\mathbf{h}}_j^{User}, \hat{\mathbf{h}}_j^{Stru}, \hat{\mathbf{h}}_j^{Temp}\right] \in \mathbf{V}_i$ represents a view feature set for a specific user j engaged in spreading tweet M_i. Then, the view-wise attention layer calculates the attention score $\mathbf{atten}_j \in \mathbb{R}^{1\times 3}$ for each view of the user-level feature set $\mathbf{v}_j \in \mathbb{R}^{d_{view}\times 3}$ as:

$$\bar{\mathbf{v}}_j = \tanh\left(\mathbf{W}_v \cdot \mathbf{v}_j\right), \tag{7}$$

$$\mathbf{atten}_j = \mathrm{softmax}\left(\mathbf{w}_v^T \cdot \bar{\mathbf{v}}_j\right), \tag{8}$$

where $\mathbf{W}_v \in \mathbb{R}^{d_{view}\times d_{view}}$, $\mathbf{w}_v \in \mathbb{R}^{d_{view}}$ are learnable projection parameters during training, $\bar{\mathbf{v}}_j = \left[\bar{\mathbf{h}}_j^{User}, \bar{\mathbf{h}}_j^{Stru}, \bar{\mathbf{h}}_j^{Temp}\right]$. Here, the view-wise attention layer first computes the hidden representation of $\mathbf{v}_j$ through multiplying it with $\mathbf{W}_v$ to get $\bar{\mathbf{v}}_j$, which is implemented with a fully connected layer without bias. It measures the weight of a view as the similarity of $\bar{\mathbf{h}}_j^*$ ($* \in \{User, Stru, Temp\}$) with a view-level context vector $\mathbf{w}_v$ and finally obtains a normalized weight through a softmax function. Each entry of $\mathbf{atten}_j$ represents an importance score for a specific view of user j.

Finally, the fused multi-view feature vector $\mathbf{v}_j'$ for user j can be calculated as:

$$\mathbf{v}_j' = \mathbf{atten}_j \cdot \mathbf{v}_j, \tag{9}$$

where $\mathbf{v}_j' \in \mathbb{R}^{d_{view}}$. The fused multi-view feature vector for each user forms the multi-view matrix, denoted as $\mathbf{V}_i' = \left\{\mathbf{v}_1', \mathbf{v}_2', \ldots, \mathbf{v}_{|U_i|}'\right\}$, where $\mathbf{V}_i' \in \mathbb{R}^{|U_i|\times d_{view}}$.

3.6. Capsule attention for user-level feature fusion

Most of existing works [21,25,26] would directly use $\mathbf{V}_i'$ for rumor detection. However, that does not properly discriminate different users, contrary to the fact that different users in a tweet propagation network may contribute differently to classifying the tweet. In our UMLARD, we introduce a capsule attention layer inspired by the recent success of capsule networks [62–64]. The Capsule network was first proposed in [62] and the main idea is to replace the scalar-output feature detectors in traditional neural networks with vector-output capsules, and train the model by the dynamic routing algorithm. It can be regarded as a parallel attention mechanism that allows each underlying capsule to attend to higher capsules at different importance.

In UMLARD, the capsule attention chooses the most related underlying vectors dynamically to form the only upper capsule via an unsupervised routing-by-agreement mechanism, which also avoids the intensive computation raised by a huge amount of parameters used in multi-layer attention. More precisely, in the nth iteration, the upper capsule $\mathbf{s}_{in}$ is calculated by:

$$\mathbf{s}_{in} = \sum_j^{|U_i|} \mathbf{a}_j \hat{\mathbf{v}}_j, \hat{\mathbf{v}}_j = \mathbf{W}\mathbf{v}_j', \tag{10}$$

where the coupling coefficient $\mathbf{a}_j$ indicates the contributions of a user capsule to the upper capsule — namely, the attention score of each user. $\mathbf{W} \in \mathbb{R}^{d_{view}\times d_{caps}}$ is the transform matrix that guarantees the feature representation ability of the center vector after clustering, and identifies the order of input features. Note that before the last iteration we add a normalization $\widetilde{\mathbf{s}}_{in} = \mathbf{s}_{in}/\|\mathbf{s}_{in}\|$ in $\mathbf{s}_{in}$ to overcome the information loss caused by the original CapsAtt [63].

The coupling coefficient $\mathbf{a}_j \in \mathbb{R}^{|U_i|\times 1}$ is determined by a "routing softmax" whose initial logit is denoted as $\mathbf{b}_j$, where $\mathbf{b}_j$ is the log prior probability that the jth user capsule should be coupled to the upper capsule $\mathbf{s}_{in}$. The coefficient is calculated by:

$$\mathbf{a}_j = \frac{\exp(\mathbf{b}_j)}{\sum_k^{|U_i|}\exp(\mathbf{b}_k)}, \tag{11}$$

The log prior is initialized with zero and then updated by adding agreements between the user capsule and the upper capsule:

$$\mathbf{b}_j = \mathbf{b}_j + \hat{\mathbf{v}}_j \cdot \widetilde{\mathbf{s}}_{in}, \tag{12}$$

These agreements are added to log priors after each routing, i.e., the output capsule $\mathbf{s}_{in}$ represents the feature matrix after correlation learning, which can be easily coupled into the model for downstream tasks, in our case the rumor detection.

3.7. Tweet content representation

Tweet content is one of the most important features in rumor detection [7–9], and has been extensively studied in the literature [11,18,25,26,65], where various NLP techniques have been exploited for learning informative signals from the textual content. Though content learning is not the main work of this article, we describe a simple CNN layer for text representation learning from the input of word embedding matrix for completeness. A single CNN layer is denoted as:

$$\mathbf{h}_m = \sigma(\mathbf{W} * \mathbf{w}_{m:m+d-1}), \tag{13}$$

where $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{L-d+1}\}$ is the extracted feature map, and $\mathbf{W} \in \mathbb{R}^{d\times d_{word}}$ is the convolutional kernel with d as size of the receptive field, and σ as non-linearity. Then max-pooling operation is used over the feature map to generate the output representation $\hat{\mathbf{H}}$. In our work, we use multiple CNN layers with different receptive field to obtain multiple features, and then concatenate all outputs to form the tweet content representation $\mathbf{H}_i^{Text}$.

3.8. Training objective

Finally, we concatenate content representation $\mathbf{H}_i^{Text}$ and capsule attention $\mathbf{s}_{in}$ to merge the information as:

$$\mathbf{H}_i^{Rumor} = \text{concat}(\mathbf{H}_i^{Text}, \mathbf{s}_{in}) \tag{14}$$

which is subsequently used for predicting the label $\hat{\mathbf{y}}_i$ of tweet M_i via a fully connected layer and the softmax function:

$$\hat{\mathbf{y}}_i = \text{softmax}\left(\text{FC}\left(\mathbf{H}_i^{Rumor}\right)\right). \tag{15}$$

We train all the parameters by minimizing the *cross-entropy* of the predictions $\hat{\mathbf{Y}}$ and the ground truth labels $\mathbf{Y}$ as:

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\sum_{i \in |\mathcal{M}|} \mathbf{y}_i \log \hat{\mathbf{y}}_i + \lambda \|\Theta\|_2^2, \tag{16}$$

where $\|\Theta\|_2^2$ is the L_2 regularizer over all the model parameters Θ, and λ is the trade-off coefficient. In this work, we use *RAdam* [66] as optimizer. The whole training process of UMLARD is outlined in Algorithm 1.

Algorithm 1 Training of UMLARD.

Input: A set of tweets $\mathcal{M} = \{M_i\}_{i=1}^{|\mathcal{M}|}$, each tweet $M_i = \{\mathcal{G}_i, \mathcal{P}_i, \mathbf{U}_i, \mathbf{C}_i\}$, and batch size B.
Output: Predicted labels $\hat{\mathbf{Y}}$ for all tweets.

repeat
  for M_i in a batch **do**
    Profile-view learning: $\mathbf{H}_i^{User} \leftarrow \mathbf{U}_i$ via Eq. (1) and Eq. (2);
    Structural-view learning $\mathbf{H}_i^{Stru} \leftarrow \mathcal{G}_i$ via Eq. (3);
    Temporal-view Learning $\mathbf{H}_i^{Temp} \leftarrow \mathcal{P}_i$ via Eq. (4) - Eq. (6);
    Content representation: $\mathbf{H}_i^{Text} \leftarrow \mathbf{C}_i$ via Eq. (13);
    Normalize dimensions:
    $\mathbf{V}_i = [\hat{\mathbf{H}}_i^{User}, \hat{\mathbf{H}}_i^{Stru}, \hat{\mathbf{H}}_i^{Temp}] \leftarrow [\mathbf{H}_i^{User}, \mathbf{H}_i^{Stru}, \mathbf{H}_i^{Temp}]$;
    View-wise attention learning: $\mathbf{V}_i^{'} \leftarrow \mathbf{V}_i$ via Eq. (7) to Eq. (9);
    Capsule attention learning: $\mathbf{s}_{in} \leftarrow \mathbf{V}_i^{'}$ via Eq. (10);
    Merge $\mathbf{H}_i^{Text}$ and $\mathbf{s}_{in}$ via Eq. (14);
    Estimate the probability $\hat{\mathbf{y}}_i$ via Eq. (15);
    Compute loss $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$, via Eq. (16);
    Update parameters using RAdam.
  end for
until convergence;

3.9. Computational complexity

We finalize this section with a discussion of the computational complexity of UMLARD, analyzed in two categories.

— *Complexity of multi-view representation learning* is influenced by four main components:

(1) As for **profile-view** that only uses dimensional-wise attention to allocate varying weights to each dimension, the computational complexity stems from the two fully connected layers, i.e., $\mathcal{O}(2d_{user}^2/r)$. Because the dimension of user characteristic d_{user} is very small, this computational cost is typically negligible.

(2) We use a multi-layer diffusion convolutional network for the **structural-view** learning (cf. Eq. (3)), which can be decomposed into two parts with the same time complexity, i.e., $\mathbf{D}_l^{-1}\mathbf{A}$ and $\mathbf{D}_0^{-1}\mathbf{A}^T$. Since the two matrices are very sparse, the time complexity is $\mathcal{O}(|E_i|)$, i.e., linear with the number of edges. Specifically, in a two-layer M-DGCN, the computational complexity is $\mathcal{O}(|E_i|DF_1F_2)$, where D, F_1 and F_2 are the input feature size, and the hidden size for the first and the last M-DGCN layer, respectively.

(3) The **temporal-view** is learned through a time-decay LSTM. The computational complexity of original LSTM per time step is $\mathcal{O}(1)$ due to LSTM is local in space and time [58]. Compared with LSTM, the only difference of our time-decay LSTM is an extra time-gate that controls the influential decreasing with time. This operator introduced extra parameters that requires $4(d_e d_{temp} + d_{temp}^2 + d_{temp}) + d_e d_{temp} + 3d_{temp}$ complexity. Besides, the dynamic embedding in UMLARD needs $N \times d_e$ parameters.

(4) For the **source tweet** representation learning, the CNN layers have the time complexity of $\mathcal{O}(\sum_{l=1}^{L}(M_l^2 K_l^2 C_{l-1} C_l))$, where L is the total number of CNN layers; K_l C_{l-1}, C_l are kernel size, input channel number and output channel number for lth layer; output size is $M_l = (X_l - K_l)/Stride + 1$ and X_l is the input feature size. Overall, this component requires $\sum_{l=1}^{L}(K_l^2 C_{l-1} C_l)$ parameters.

— *Complexity of fusion layers.* In the hierarchical fusion layers, the time and space complexities of both view-wise attention and capsule attention are related to the input and output dimensions of the latent variables. In view-wise attention, it introduces $d_{view} \times d_{view} + |U_i| \times d_{view}$ parameters. As for the capsule attention layer, the parameter size is $d_{view} \times d_{caps}$, where d_{view} and d_{caps} represent view size and capsule size, respectively.

4. Experimental observations

We now present the findings from our experimental evaluations. We compare the performance of our UMLARD with the state-of-art baselines on rumor detection, and we also investigate the effects of different components by comparing several variants of UMLARD.

Specifically, we would aim at providing quantitative characterization of the following research-related questions:

- **RQ1**: How does UMLARD perform on rumor detection compare with the state-of-the-art baselines?
- **RQ2**: What is the effect of each component of UMLARD?
- **RQ3**: Can UMLARD detect rumors in early stages of their propagation?
- **RQ4**: Can UMLARD explain the model behavior and the predicted results?

4.1. Experimental settings

Following is the description of the main aspects of our experimental setup.

(1) Datasets: We conduct our experiments on three real-world datasets: *Twitter15*, *Twitter16* [28] and *Weibo* [18]. Twitter and Weibo are popular social media sites in U.S. and China. In each dataset, a group of widespread source tweets along with their propagation threads with time stamps are provided. We construct propagation paths and diffusion networks from the propagation threads, which are also used for user temporal-aspect embedding and user structural-aspect embedding.

In Twitter datasets, each source tweet is annotated with one of the four class labels, i.e., *non-rumor*, *false-rumor*, *true-rumor*, and *unverified-rumor*, while the Weibo dataset contains binary labels: *false-rumor*, *true-rumor* — the labeling rules follow the method in [18]. The statistics of the three datasets are shown in Table 2.

Due to the constraints of the Twitter service terms, the original datasets do not contain user profile information. We crawl all the related user profiles via *Twitter API*,[2] based on the provided user IDs. From the crawled user profiles, we extract eight user characteristics: (1) *The length of user name* (2) *The user account created time*, (3) *The length of description*, (4) *The followers count*, (5) *The friends ccount*, (6) *The statuses count*, (7) *Is verified*, and (8) *Is geo-enabled*. In contrast, as for the Weibo dataset, we directly extract these eight characteristics from the JSON files in the original dataset.

Following previous works [19,21,25], we randomly choose 70% data for training, 10% data for validation, and the remaining data for testing.

(2) Baselines: We compare UMLARD with following state-of-the-art rumor detection baseline models:

[2] https://dev.twitter.com/rest/public.

- **DTC** [7]: A decision tree-based classification model that combines manually engineered characteristics of tweets to compute the information credibility.
- **SVM-RBF** [24]: A support vector machine (SVM) based model that uses radius basis function (RBF) as the kernel and leverages the handcrafted features of posts for rumor detection.
- **SVM-TS** [23]: A linear SVM-based time series model that captures the variation of a wide spectrum of social context information over time through converting the continuous-time stream into fixed time intervals.
- **GRU** [67]: A variant of the RNN with the gated recurrent units that has been employed in [18] to learn the sequential cascading effect of tweets with high-level feature representations extracted from relevant posts over time.
- **TD-RvNN** [19]: A tree-structured model based on RNN for rumor detection, which embeds hidden indicative signals in the tree-structures and explores the importance of tweet content for rumor detection.
- **PPC_RNN+CNN (PPC)** [21]: A model for early-stage rumor detection through classifying news propagation paths with RNN and CNN, which learns the rumor representations through the characteristics of users and source tweets.
- **PLAN** [68]: A hierarchical token- and post-level attention model for rumor detection, which models pairwise interactions between tweets via the self-attention mechanism.
- **Bi-GCN** [26]: A GCN-based model exploiting the bi-directional propagation structures and text contents (i.e., source tweet and comments) for rumor detection. We also provide a variant of Bi-GCN, denoted as Bi-GCN-U, which uses user profile characteristics to replace the comment features.
- **STS-NN** [69]: A rumor detection model based on spatial–temporal neural networks. It treats the spatial structure and temporal structures as a whole to learn a fine-grained rumor representation.
- **GCAN** [25]: A co-attention network that detects true and false rumors based on the content of the source tweet and its propagation-based users. For fair comparison, we also provide a variant of GCAN, named GCAN-G, which uses the diffusion graph to replace the user similarity graph.
- **RDEA** [49]: A self-supervised rumor detection model. On the basis of Bi-GCN [26], RDEA improves the rumor representations and alleviates limited data issues through event augmentation and contrastive learning.

(3) Implementation details: We implement DTC with Weka,[3] SVM-based models with scikit-learn,[4] and other neural network-based models with Tensorflow.[5] All baselines follow the parameter settings in the original papers. For UMLARD, the learning rate is initialized at 0.001 and gradually decreases as the training proceeds. We use word2vec to initialize the word embeddings with $d_{word} = 300$ dimensions, and the convolution kernel size is set to [3, 4, 5], and per size with 100 kernels. The embedding size for structural view d_{stru} and temporal view d_{temp} of users are both set to 64; the view size d_{view} is also set to 64, as is the capsule size; and the iteration number varies between 2 and 4. The batch size is 64; and the rate of dropout in the main neural networks is 0.5; the dropout rate in DropEdge is 0.2. The training process is iterated upon for 200 epochs, but would be stopped earlier if the validation loss does not decrease after 10 epochs.

[3] https://www.cs.waikato.ac.nz/ml/weka/.

[4] https://scikit-learn.org/.

[5] https://www.tensorflow.org/.

Table 2
Statistics of the datasets.

Statistic	Twitter15	Twitter16	Weibo
# source tweets	1482	809	4664
# users	477,009	286,657	2,746,818
# non-rumors	370	199	–
# false-rumors	369	205	2313
# true-rumors	372	207	2351
# unverified-rumors	371	198	–
Max. # retweets	2989	3058	59,318
Min. # retweets	55	73	10
Avg. # retweets	398	422	816
Avg. # time length	1268 h	828 h	1811 h

(4) Evaluation metrics: We use accuracy (ACC) and F-measure (F1) as the evaluation protocols to measure the models' performance. Specifically, ACC measures the proportion of correctly classified tweets, while F1 is the harmonic mean of the precision and recall values averaged across four classes. As for the Weibo dataset, we also report the precision and recall results.

4.2. Overall performance (RQ1)

Tables 3 and 4 reports the performance comparison among UMLARD and baselines on three datasets, from which we have the following observations:

O1: Feature-based approaches such as SVM-TS, SVM-RBF, and DTC perform poorly. These methods use hand-crafted features based on the overall statistics of tweets, but are not sufficient to capture the generalizable features associated with tweets and the process of information diffusion. Notably, SVM-RBF performs worse than the other two methods on two Twitter datasets. However, it achieves the second-best performance among the feature-based modes on Weibo dataset, because it selects the features based on Weibo that are hard to be generalized to other social platforms such as Twitter. SVM-TS achieves relatively better performance because it utilizes an extensive set of features and primarily focuses on retweets' temporal traits.

O2: Deep learning-based models perform significantly better than feature-based methods. As the first work exploiting RNN for efficient rumor detection, GRU only relies on temporal-linguistics of the repost sequence while ignoring other useful information such as diffusion structures and user profiles. TD-RvNN and PPC_RNN+CNN outperform GRU, which indicates the effectiveness of modeling the propagation structure and temporal information in rumor detection. The performance of PLAN slightly exceeds TD-RvNN and PPC_RNN+CNN, because it still mainly focuses on textual information and ignores structural features of rumor propagation.

O3: Bi-GCN, GCAN, STN-SS, and RDEA have considered structural or temporal information, and thus outperform other baselines. In particular, Bi-GCN constructs the diffusion tree based on user replies, i.e., the retweets with comments, which may not reflect the whole structure of rumor dispersion. In contrast, GCAN models the structural information from the user similarity matrix rather than propagation network. Therefore, according to the results, Bi-GCN performs much better than GCAN, because it takes the comments information into consideration. Besides, the bi-directional GCN is more effective in learning propagation structures than vanilla GCN used in GCAN. Although STS-NN extracts both structural and temporal features for rumor detection, STS-NN still performs worse than Bi-GCN and GCAN, because it fails to discriminate the spatial structures and the temporal patterns. RDEA improves the performance of Bi-GCAN via introducing contrastive learning and event augmentations, which alleviate the influence of limited data issue. However, this

Table 3
Overall performance comparison of rumor detection on Twitter15 and Twitter16 (the observation window is set to the previous 40 retweets). "UR": unverified-rumor; "NR": non-rumor; "TR": true-rumor; "FR": false-rumor. The best method is shown in **bold**, and the second best is shown as underlined. A paired t-test is performed and $*$ indicates a statistical significance $p < 0.05$ compared to the best baseline method (RDEA).

Model	Twitter15					Twitter16				
	ACC.	F1				ACC.	F1			
		UR	NR	TR	FR		UR	NR	TR	FR
DTC	0.454	0.415	0.733	0.317	0.355	0.465	0.403	0.643	0.419	0.393
SVM-RBF	0.318	0.218	0.225	0.455	0.082	0.321	0.419	0.037	0.423	0.085
SVM-TS	0.544	0.483	0.796	0.404	0.472	0.574	0.526	0.755	0.571	0.420
GRU	0.646	0.608	0.592	0.792	0.574	0.633	0.686	0.593	0.772	0.489
TD-RvNN	0.723	0.654	0.682	0.821	0.758	0.737	0.708	0.662	0.835	0.743
PPC	0.697	0.689	0.760	0.696	0.645	0.702	0.608	0.711	0.816	0.664
PLAN	0.787	0.775	0.7754	0.768	0.807	0.799	0.779	0.754	0.836	0.821
Bi-GCN	0.829	0.752	0.772	0.885	0.847	0.837	0.818	0.772	0.885	0.847
Bi-GCN-U	0.778	0.764	0.741	0.853	0.752	0.786	0.733	0.783	0.875	0.767
GCAN	0.808	0.690	**0.930**	0.812	0.758	0.765	0.784	0.848	0.678	0.754
GCAN-G	0.750	0.731	0.754	0.823	0.678	0.721	0.642	0.690	0.799	0.732
STS-NN	0.808	0.779	0.786	0.860	0.808	0.829	0.838	0.775	0.899	0.809
RDEA	0.835	0.819	0.786	0.887	0.837	0.848	**0.868**	0.729	0.922	0.823
UMLARD	**0.857***	**0.835***	0.840*	**0.906***	**0.848***	**0.901***	0.822*	**0.965***	**0.960***	**0.855***

Table 4
Overall performance comparison of rumor detection on Weibo (the observation window is set to the previous 40 retweets). "TR": true-rumor; "FR": false-rumor. The best method is shown in **bold**, and the second best is shown as underlined. A paired t-test is performed and $**$ indicates a statistical significance $p < 0.01$ compared to the best baseline method (RDEA).

Model	Weibo						
	ACC	TR			FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1
DTC	0.731	0.715	0.747	0.730	0.747	0.715	0.731
SVM-RBF	0.741	0.738	0.747	0.742	0.745	0.735	0.740
SVM-TS	0.780	0.801	0.753	0.780	0.762	0.808	0.784
GRU	0.762	0.803	0.715	0.757	0.728	0.809	0.767
TD-RvNN	0.832	0.832	0.812	0.821	0.821	0.861	0.841
PPC	0.845	0.870	0.810	0.839	0.810	0.883	0.844
PLAN	0.857	0.829	0.904	0.857	0.893	0.805	0.835
Bi-GCN	0.891	0.892	0.892	0.890	0.891	0.891	0.890
Bi-GCN-U	0.864	0.896	0.818	0.860	0.830	0.910	0.868
GCAN	0.880	0.911	0.861	0.885	0.866	0.929	0.896
GCAN-G	0.831	0.815	0.824	0.819	0.847	0.815	0.831
STS-NN	0.875	0.881	0.866	0.865	0.851	0.872	0.852
RDEA	0.911	0.902	0.923	0.907	**0.913**	0.899	0.901
UMLARD	**0.928****	**0.942****	**0.965****	**0.924****	0.894**	**0.944****	**0.928****

method still faces the same problem as Bi-GCN, i.e., reply network is not enough to represent the full information diffusion process. Through comparing UMLARD with Bi-GCN-U and GCAN-G, we find that the performance of Bi-GCN-U and GCAN-G drops significantly. This result indicates that these methods heavily depend on the input features and are ineffective in extracting diffusion patterns as our method.

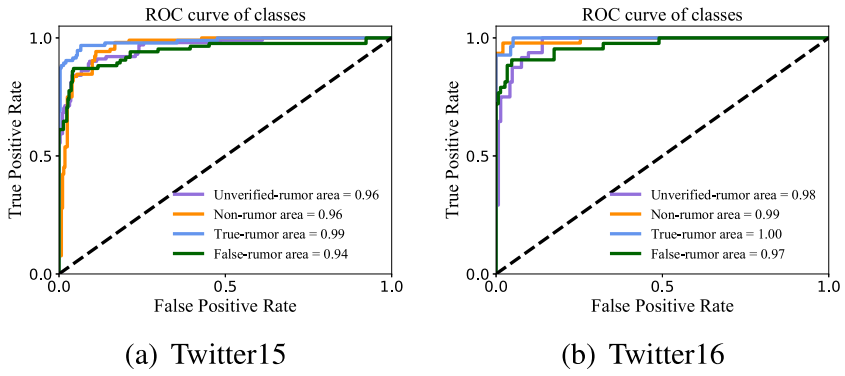
O4: UMLARD consistently outperforms all other baselines across all datasets. Compare to the best baseline RDEA, UMLARD models rumor diffusion from multi-view perspective that allows the model to discriminate the importance of features and users in spreading the tweets. These results also validate one of our primary motivations, i.e., various features play different roles in spreading the rumors, and users are the main contributor to the misinformation propagation.

Finally, we scrutinize the performance of UMLARD on discriminating against the individual type of information on Twitter15 and Twitter16. Fig. 4 plots the ROC curves of the model performance on four different kinds of tweets. We find that our model achieves the best identification results on true-rumors, which indicates that the characteristics of true-rumors are more distinctive from other types of messages. This result also implies that our model is more expressive on a binary classification task that only needs to classify tweets as rumors or truths (cf. the results on Weibo in Table 4). In practice, however, unverified-rumors and false-rumors are noisy signals that require careful treatment, which is a promising way of further improving the detection accuracy.

4.3. Ablation experiments (RQ2)

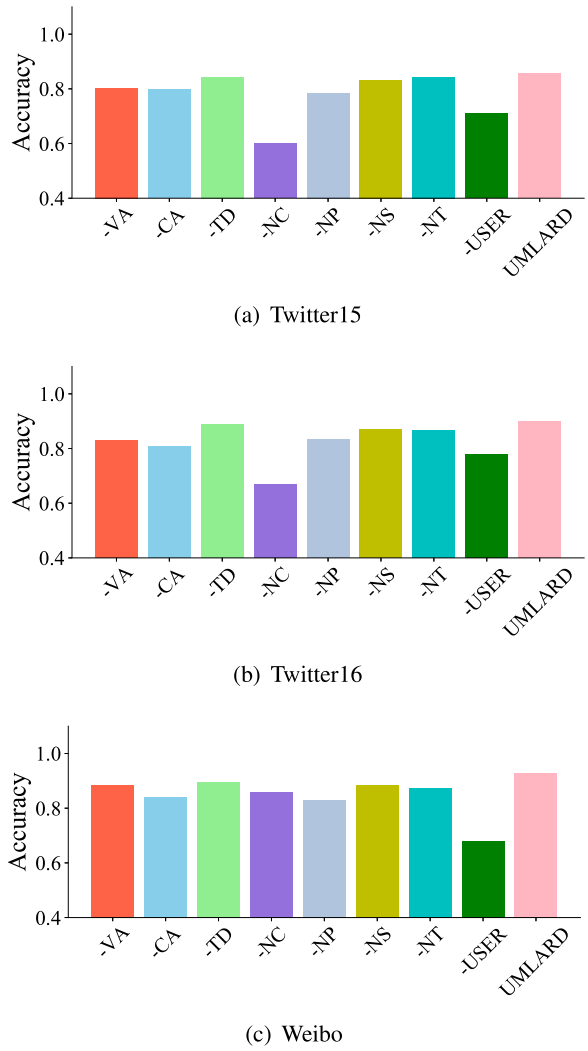
In this section, we conduct an ablation study to explore the effect of each component in UMLARD. Towards that, we derive the following variants of UMLARD:

- **-VA**: In -VA, ignores the different importance of different views, i.e., it removes the view-wise attention layer.
- **-CA**: In -CA, replaces the capsule attention layer with a fully connected layer.
- **-TD**: In -TD, neglects the time decay effect of retweet behaviors which is replaced by a vanilla LSTM [58] to learn sequential retweet behavior.
- **-NC**: In -NC, removes the content feature of the source tweet but keeps the temporal, profile, and structural features.
- **-NP**: In -NP, disregards the profile features of users but retains temporal, structural, and content features.
- **-NS**: In -NS, ignores the structural features of users but keeps temporal, profile, and content features.



(a) Twitter15 (b) Twitter16

Fig. 4. ROC curve comparison for each information type. Area under curve of ROC (AUC) is presented after the legend.



(a) Twitter15

(b) Twitter16

(c) Weibo

Fig. 5. Ablation study of UMLARD. Two attention mechanisms can significantly improve the detection performance by distinguishing the importance of features and users. Tweet content and profile information are two most informative features on rumor detection.

- **-NT**: In -NT, ignores the temporal features of users but keep structural, profile, and content features.
- **-USER**: In -USER, ignores the user-aspect features (i.e., temporal, structural, and profile) that only retains the content feature.

Fig. 5 illustrates the performance of the variants, where we can observe that:

(1) The content of tweet (**-NC** and **-USER**) is still the most critical signal of discriminating rumors among various features. Without it, the model performance would significantly drop, as observed in many previous works [19,70]. However, only based on the content feature is insufficient to develop an effective rumor detection model that can identify different types of rumors with high accuracy.

(2) Profile information (**-NP**) is another reliable indicator to detect the rumors because it is a straightforward but useful method to identify the users that spread the misinformation intentionally [14,15].

(3) Though both structural (**-NS**) and temporal information (**-NT**) are informative, they are not as important as contents of tweets and user profiles. This result also explains why the methods proposed in [21], and [26] do not show comparable performance as ours — the former mainly focuses on modeling the temporal information of retweets, whereas the latter one relies on graph neural networks to exploit the diffusion structures. We also conduct additional experiments to demonstrate the importance of structural and temporal features once the input contains enough information, especially in a binary classification task (e.g., Weibo). The results are shown in Fig. 6. We find that as for Twitter datasets, the detection performance based on structural features grows slightly but is still not good enough as the type of rumors is fine-grained, making it challenging to learn discriminative structural features. As for the Weibo dataset, both structural and temporal features are helpful for rumor detection even in a short time.

In order to demonstrate our findings in Fig. 6, we conduct statistical analysis of the datasets and plot the temporal and structural propagation patterns in Figs. 8–11. We find that the differences in temporal patterns are more obvious compared with the structural patterns. In addition, the differences between true and false rumors in Weibo are more significant than the discrepancy between the fine-grained types of rumors in Twitter datasets.

(4) The two attention mechanisms proposed in this work, i.e., view-wise attention (**-VA**) and capsule attention (**-CA**), play a crucial role on identifying the misinformation — the importance of which even exceed temporal features and diffusion patterns. This result also suggests that distinguishing the significance of different views of users can improve classification performance. Similarly, different users play different roles in spreading misinformation, e.g., users may intentionally mislead others or unknowingly retweet doubtful news. However, examining users' purposes is beyond the scope of this work and is left as our future work.

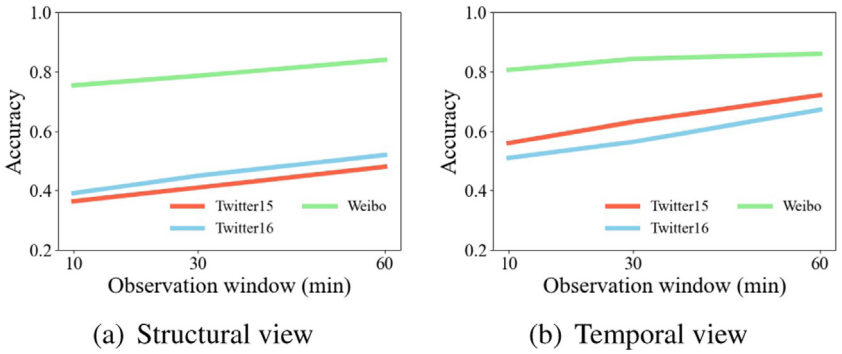


Fig. 6. Study on structural and temporal view. We only use structural or temporal views to detect rumors in 10, 30, and 60 mins. The results demonstrate the importance of both views in rumor detection.

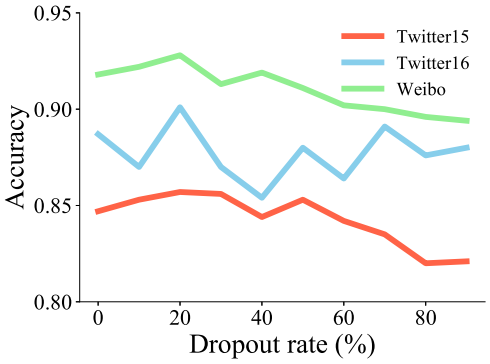


Fig. 7. DropEdge Study of UMLARD.

(5) Finally, the discrepancy between UMLARD and **-TD** indicates the gain of modeling time decay in retweet cascades. In other words, both real information and false information will significantly reduce their influence over time.

To further investigate the content-aspect effect, we examine the influence of different word embedding methods. Specifically, we use the state-of-the-art Bert-based pretraining model [71] to replace the word2vec and then compare the performance on accuracy and macro-F1. In our work, we choose BERT-Base,[6] which was trained on a large text corpus (e.g., Wikipedia). The results are shown in Fig. 12. We can observe that the Bert-based UMLARD is surprisingly incomparable. This happens due to the characteristics of tweet text, which are *short*, *sparse*, *sporadic* and written *casually*. Therefore, the Bert-based pretraining techniques that are usually trained on large-scale language corpus are difficult to directly used for short-text tasks such as Twitter content embedding. This conjecture is in accordance with some recent observations on [72].

Furthermore, at the end of this section, we conduct an extra experiment to demonstrate the effectiveness of the "DropEdge" technical used in the data prepossessing. The dropout rate is set from 0 to 0.9, and the experimental results are shown in Fig. 7. We find that slightly dropping the edges in the diffusion graph would improve the model performance.

4.4. Performance on early detection (RQ3)

Another important goal of rumor detection is to detect misinformation as early as possible and stop its spread in a timely fashion. Now we investigate the performance of models on identifying rumors at early-stage. Here, we consider two metrics for gauging the observation windows of information spread, i.e., the previous 40 retweets and the propagation in the first hour. In this section, the experiments of early detection are conducted on Twitter15 and Twitter16.

Fig. 13 shows the performance comparison on early-stage detection between our UMLARD and the baselines. Note that we omit the feature-based methods and credibility-based approaches since they did not show comparable performance, especially on early rumor detection. We observe that UMLARD performs better, especially when there are only a few observations. UMLARD needs a short time to identify the misinformation because it fuses the multi-view knowledge of users. For example, understanding the role of a user in spreading information is vital since tweets' size, spread speed and patterns are different. Moreover, UMLARD is capable of discriminating the importance of features even with few observations, which means the interference caused by the trivial or useless features would be dampened during training the model. In all cases, their early detection accuracy grows at the early stage of propagation. However, we find that the performance of our model demonstrates obvious advantage as time goes on.

We also investigate the time-varying performance between the variants and the full UMLARD. As shown in Fig. 14, we find that the accuracy of all methods grows to saturation with increasing the number of retweets or time elapsed. Moreover, from Fig. 14(c) and 14(d), we can observe that the performance of -NP, -NT, and -NS is very close to the full UMLARD, because the models have acquired enough knowledge to detect rumors within a short observation time.

4.5. Interpretability analysis (RQ4)

The above experimental results have shown the superiority of the proposed hierarchical attentions. Namely, they can effectively discriminate the importance of multi-views of users and the roles of users in spreading the (mis)information. Here, we provide more in-depth insights into the two components by visualizing the hierarchical attention layers in UMLARD.

Fig. 15 shows the importance of user-profiles and users themselves — the higher the value, the more important the feature or the user. Fig. 15(a) plots the importance of eight user profile characteristics, where we vary the number of observed retweets between {5, 10, 15, 20}. We can observe that the follower counts is the most informative feature, followed by the register time, verified account, and geo-enabled features, consistent with the findings in [14,14,15], i.e., the users enrolled in spreading of rumors have fewer followers.

In Fig. 15(b), we investigate the role of the retweet users at the very beginning of the cascade. As shown, the earlier users are more important for detecting *non-rumors* (NR) and *true-rumors*

[6] https://github.com/google-research/bert.

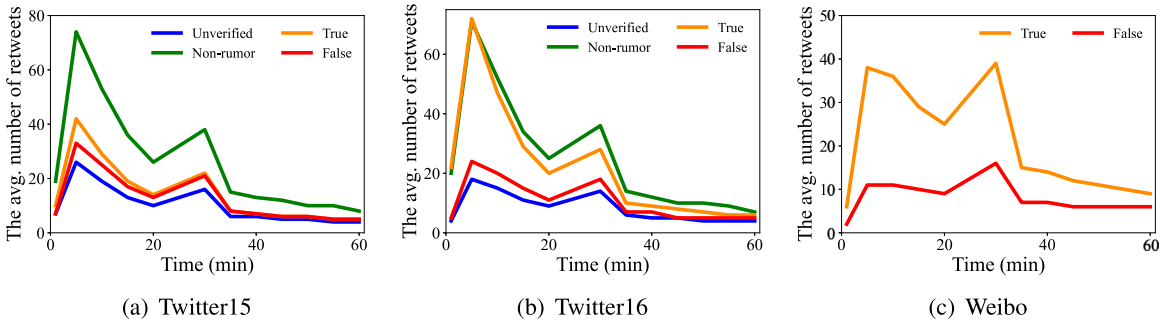


Fig. 8. The average number of retweets for different types of rumors at different timestamps.

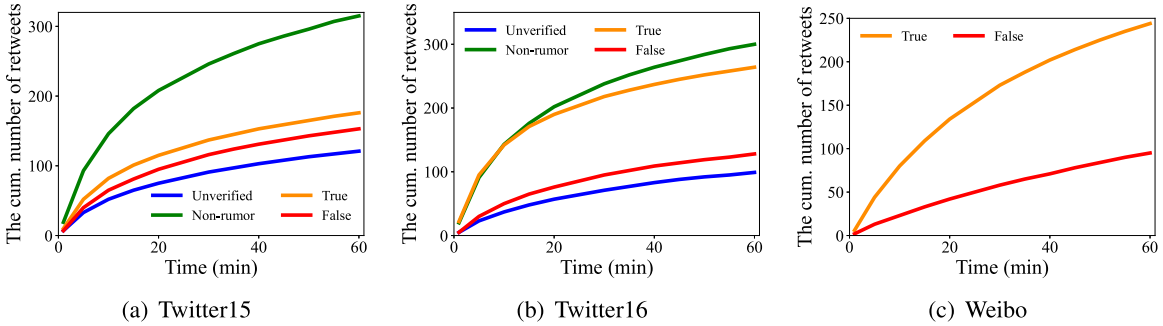


Fig. 9. The cumulative number of retweets for different types of rumors at different timestamps.

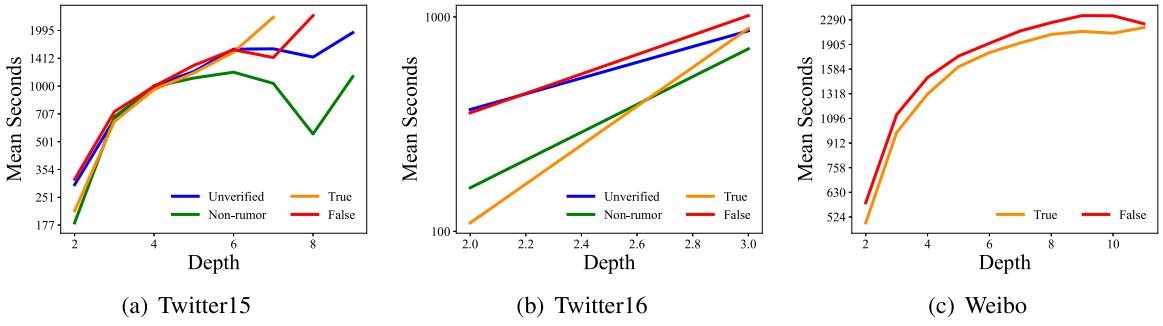


Fig. 10. The average time (in seconds) required to reach the same network depth. The observation window 60 min.

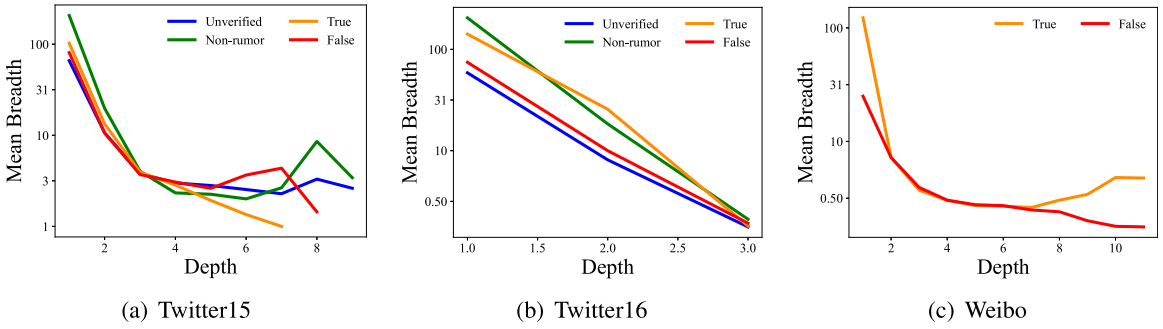


Fig. 11. The average network breadth for different types of rumors. The observation window is 60 min.

(TR). To the contrary, the latter participators are important for detecting *unverified-rumors* (UR) and *false-rumors* (FR). This phenomenon shows that authoritative users usually spread TRs and NRs at the beginning of spreading information. URs and FRs, after the false information spread a while, will see an influx of massive malicious users, who would pretend these tweets as real information.

We now discuss the impact of the different views of users in rumor detection. We randomly selected four different types of tweets in Twitter15 and plots the importance of different views. Figs. 16(a) and 16(b) show the results of previous 5 and 10 retweet users, respectively. Overall, we can see that the three views of each user in this tweet have different importance. Specifically, when there are few observations (e.g., only 5 retweet

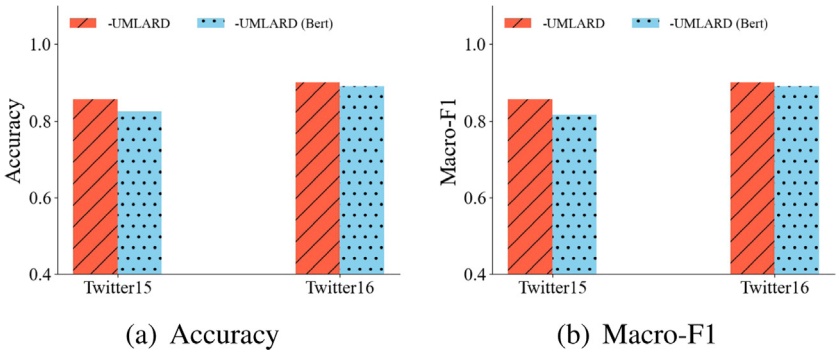
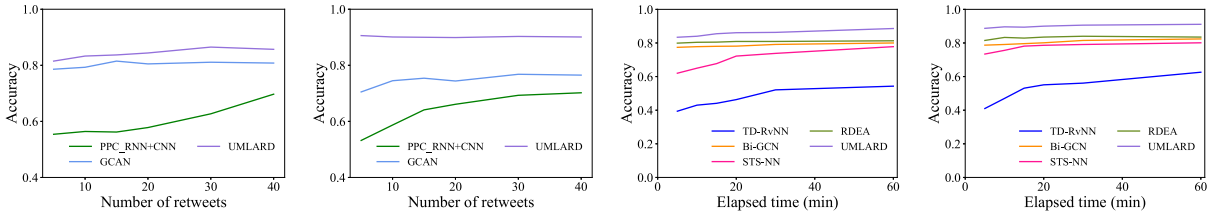
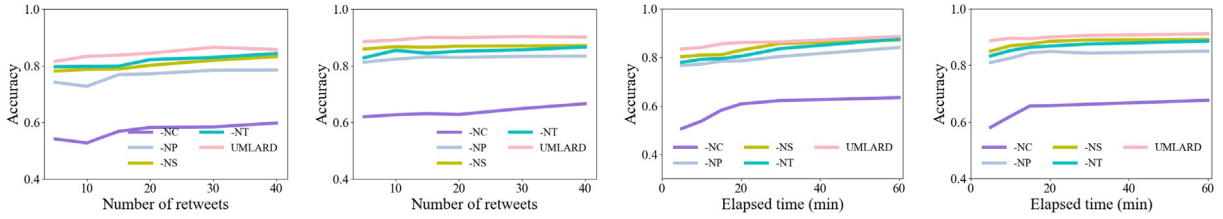


Fig. 12. Content-aspect study of UMLARD.



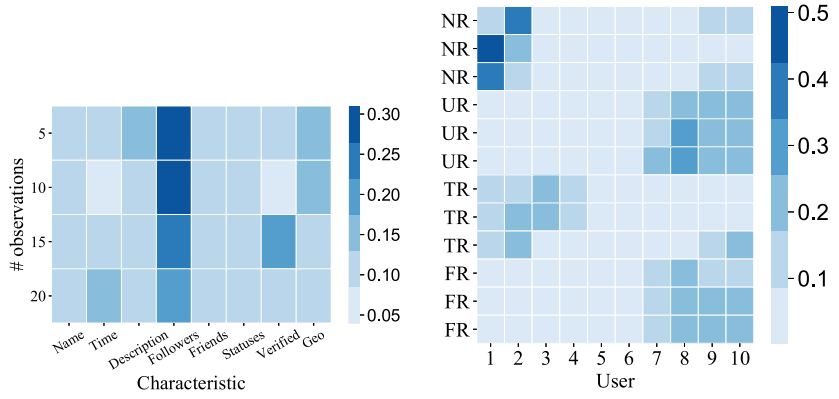
(a) Early 40 retweets (Twitter15). (b) Early 40 retweets (Twitter16). (c) Early 1-hour retweets (Twitter15). (d) Early 1-hour retweets (Twitter16).

Fig. 13. Evaluations on early rumor detection. (a) and (b): PPC_RNN+CNN and GCAN are cascade length-based methods. (c) and (d): Tv-RvNN, Bi-GCN, STS-NN and RDEA are built on the user comments that may not exist in early-stage retweets — hence, we observe their performance over time.



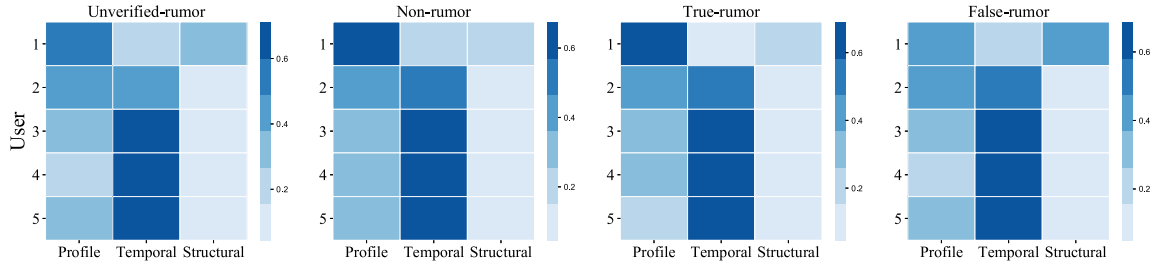
(a) Early 40 retweets (Twitter15). (b) Early 40 retweets (Twitter16). (c) Early 1-hour retweets (Twitter15). (d) Early 1-hour retweets (Twitter16).

Fig. 14. Evaluations on early rumor detection among variants of UMLARD. (a) and (b): The model performance using early 40 retweets. (c) and (d): The models are trained with early one hour observations.

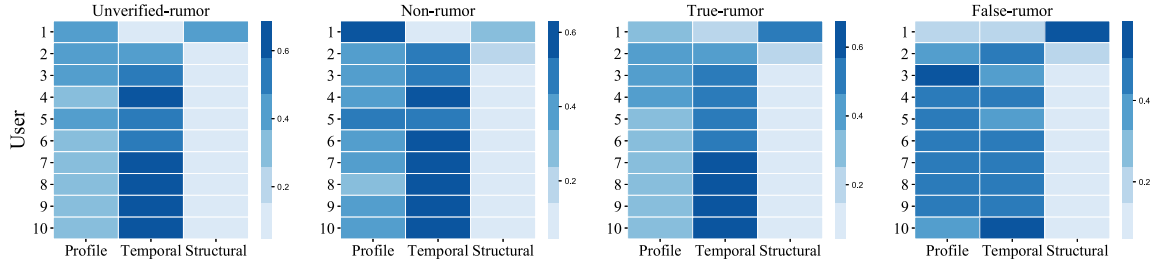


(a) User profile attention. (b) Capsule attention.

Fig. 15. Visualization of user profiles importance and the role of earlier spreaders (Twitter15).



(a) View Attention, observation length = 5.



(b) View Attention, observation length = 10.

Fig. 16. Visualization of the different user-aspect importance (Twitter15).

users), the profile view and the temporal view of the users dominate the rumor detection performance. As the number of retweet users increases, the structural information becomes more and more important. This result can be understood intuitively: In reality, at the very beginning the participants directly retweet the information from the source spreader, which leads to the similar propagation structures of information cascades. However, users are different from each other in profile and the time of retweeting, which are, consequently, the most important views for early-stage misinformation detection. Besides, by comparing different types of information, the non-rumor and the true-rumor have very similar weight distribution over different users' views, as observed in Fig. 15(b).

5. Conclusion

In this work, we presented UMLARD — a novel model for rumor detection which fuses multiple information contexts pertaining to users of social networks. Combining multiple views of users aspects and discriminating the importance of spreaders and user-aspect information, we successfully identified users' roles in different stages of rumor diffusion. UMLARD significantly outperforms previous methods in terms of misinformation classification and rapid rumor detection. Our approach is also notable in its strength of interpreting model behaviors and the predicted results. The experiments conducted on real Twitter datasets support the hypothesis that characteristics of user-profiles, aspects view of participants, as well as user's engagement time and tweets' diffusion patterns, can contribute to the misinformation prediction from the collective signals. Besides, our experimental results on early-detection discern several vital features of false information.

Although our method enriches the body of work on rumor detection via modeling systematic user-aspect information, we envision several future works. Recall that UMLARD only provides primary textual features of the source tweets for detecting and tracking the rumors. However, it is of interest for OSMs and policymakers to intervene in the spread of misinformation by checking the fact of the claims in the tweets. Therefore, taking more explicitly into account the verification is a promising way of improving prediction performance [65]. Besides, users' stance and intentions are critical in identifying the misinformation, which requires careful consideration as to why a particular user is involved in retweeting an article [73]. Finally, the structure of the information cascade provides the least informative signals in our model, which does not mean that the structural information is trivial in rumor detection. On the contrary, a recent study [74] suggests that the collective sharing pattern of the crowd may reveal underlying patterns of rumor spreading that is the same important as tweet content and user attributes, which is noteworthy for further examination.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Research supported in part by the National Natural Science Foundation of China (Grant No. 62072077 and 62176043) and the National Science Foundation SWIFT, USA grant 2030249.

References

[1] Z. Jin, J. Cao, Y. Zhang, J. Luo, News verification by exploiting conflicting social viewpoints in microblogs, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, in: AAAI, vol. 16, 2016, pp. 2972–2978.
[2] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, J. Econ. Perspect. (2017) 211–236.
[3] E. Hadavandi, H. Shavandi, A. Ghanbari, Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting, Knowl.-Based Syst. 23 (8) (2010) 800–808.
[4] W. Ahmed, J. Vidal-Alaball, J. Downing, F. López Seguí, COVID-19 and the 5G conspiracy theory: Social network analysis of Twitter data, J. Med. Int. Res. (2020) e19458.

[5] F. Xu, V.S. Sheng, M. Wang, Near real-time topic-driven rumor detection in source microblogs, Knowl.-Based Syst. 207 (2020) 106391.
[6] A.I.E. Hosni, K. Li, Minimizing the influence of rumors during breaking news events in online social networks, Knowl.-Based Syst. 193 (2020) 105452.
[7] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th International Conference on World Wide Web, in: WWW, vol. 11, 2011, pp. 675–684.
[8] S. Kwon, M. Cha, K. Jung, W. Chen, Y. Wang, Prominent features of rumor propagation in online social media, in: 2013 IEEE 13th International Conference on Data Mining, in: ICDM, vol. 13, 2013, pp. 1103–1108.
[9] Z. Zhao, P. Resnick, Q. Mei, Enquiring minds: Early detection of rumors in social media from enquiry posts, in: Proceedings of the 24th International Conference on World Wide Web, in: WWW, vol. 15, 2015, pp. 1395–1405.
[10] A. Hassan, V. Qazvinian, D. Radev, What's with the attitude?: identifying sentences with attitude in online discussions, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, in: EMNLP, vol. 10, 2010, pp. 1245–1255.
[11] B. Ma, D. Lin, D. Cao, Content representation for microblog rumor detection, in: Advances in Computational Intelligence Systems, 2017, pp. 245–251.
[12] M. Gupta, P. Zhao, J. Han, Evaluating event credibility on twitter, in: Proceedings of the 2012 SIAM International Conference on Data Mining, in: SDM, vol. 12, 2012, pp. 153–164.
[13] Z. Zhang, Z. Zhang, H. Li, Predictors of the authenticity of internet health rumours, Health Inf. Libr. J. (2015) 195–205.
[14] K. Shu, S. Wang, H. Liu, Understanding user profiles on social media for fake news detection, in: 2018 IEEE Conference on Multimedia Information Processing and Retrieval, in: MIPR, vol. 18, 2018, pp. 430–435.
[15] K. Shu, X. Zhou, S. Wang, R. Zafarani, H. Liu, The role of user profiles for fake news detection, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, in: ASONAM, vol. 19, 2019, pp. 436–439.
[16] Y. Yang, K. Niu, Z. He, Exploiting the topology property of social network for rumor detection, in: 2015 12th International Joint Conference on Computer Science and Software Engineering, in: JCSSE, vol. 15, 2015, pp. 41–46.
[17] Z. Jin, J. Cao, Y.-G. Jiang, Y. Zhang, News credibility evaluation on microblog with a hierarchical propagation model, in: 2014 IEEE International Conference on Data Mining, in: ICDM, vol. 14, 2014, pp. 230–239.
[18] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K. Wong, M. Cha, Detecting rumors from microblogs with recurrent neural networks, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, in: IJCAI, vol. 16, 2016, pp. 3818–3824.
[19] J. Ma, W. Gao, K. Wong, Rumor detection on twitter with tree-structured recursive neural networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, in: ACL, vol. 18, 2018, pp. 1980–1989.
[20] Z. Jin, J. Cao, H. Guo, Y. Zhang, J. Luo, Multimodal fusion with recurrent neural networks for rumor detection on microblogs, in: Proceedings of the 25th ACM International Conference on Multimedia, in: MM, vol. 17, 2017, pp. 795–816.
[21] Y. Liu, Y.-F.B. Wu, 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, in: AAAI, vol. 18, 2018.
[22] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.
[23] J. Ma, W. Gao, Z. Wei, Y. Lu, K. Wong, Detect rumors using time series of social context information on microblogging websites, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, in: CIKM, vol. 15, 2015, pp. 1751–1754.
[24] F. Yang, Y. Liu, X. Yu, M. Yang, Automatic detection of rumor on sina weibo, in: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, in: MDS, vol. 12, 2012.
[25] Y.-J. Lu, C.-T. Li, 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, in: ACL, vol. 20, 2020, pp. 505–514.
[26] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, J. Huang, Rumor detection on social media with bi-directional graph convolutional networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, in: AAAI, vol. 20, 2020, pp. 549–556.
[27] X. Chen, F. Zhou, F. Zhang, M. Bonsangue, Modeling microscopic and macroscopic information diffusion for rumor detection, Int. J. Intell. Syst. 36 (10) (2021) 5449–5471.
[28] J. Ma, W. Gao, K. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, in: ACL, vol. 17, 2017, pp. 708–717.
[29] K. Shu, L. Cui, S. Wang, D. Lee, H. Liu, DEFEND: Explainable fake news detection, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in: KDD, vol. 19, 2019, pp. 395–405.
[30] D. Zhao, Q. Gao, Y. Lu, D. Sun, Y. Cheng, Consistency and diversity neural network multi-view multi-label learning, Knowl.-Based Syst. 218 (2021) 106841.
[31] C. Zhu, P. Wang, L. Ma, R. Zhou, L. Wei, Global and local multi-view multi-label learning with incomplete views and labels, Neural Comput. Appl. 32 (18) (2020) 15007–15028.
[32] K. Jia, J. Lin, M. Tan, D. Tao, Deep multi-view learning using neuron-wise correlation-maximizing regularizers, IEEE Trans. Image Process. 28 (10) (2019) 5121–5134.
[33] J. Tang, Y. Tian, D. Liu, G. Kou, Coupling privileged kernel method for multi-view learning, Inform. Sci. 481 (2019) 110–127.
[34] J. Tang, W. Xu, J. Li, Y. Tian, S. Xu, Multi-view learning methods with the LINEX loss for pattern classification, Knowl.-Based Syst. 228 (2021) 107285.
[35] C. Wang, X. Chen, B. Chen, F. Nie, B. Wang, Z. Ming, Learning unsupervised node representation from multi-view network, Inform. Sci. 579 (2021) 700–716.
[36] N. DiFonzo, P. Bordia, Rumor and prediction: Making sense (but losing dollars) in the stock market, Organ. Behav. Hum. Decis. Process. (1997) 329–353.
[37] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, ACM Trans. Web 1 (1) (2007) 5–es.
[38] K. Wu, S. Yang, K.Q. Zhu, False rumors detection on sina weibo by propagation structures, in: 2015 IEEE 31st International Conference on Data Engineering, in: ICDE, vol. 15, 2015, pp. 651–662.
[39] Z. Jin, J. Cao, Y. Zhang, J. Zhou, Q. Tian, Novel visual and statistical image features for microblogs news verification, IEEE Trans. Multimed. (2016) 598–608.
[40] D. Khattar, J.S. Goud, M. Gupta, V. Varma, Mvae: Multimodal variational autoencoder for fake news detection, in: The World Wide Web Conference, in: WWW, vol. 19, 2019, pp. 2915–2921.
[41] J. Ma, W. Gao, K. Wong, Detect rumor and stance jointly by neural multi-task learning, in: Companion Proceedings of the the Web Conference 2018, in: WWW, vol. 18, 2018, pp. 585–593.
[42] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, X. Zhang, Rumor detection on social media with graph structured adversarial learning, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, in: IJCAI, vol. 20, 2020, pp. 1417–1423.
[43] L. Cui, H. Seo, M. Tabar, F. Ma, S. Wang, D. Lee, Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in: KDD, vol. 20, 2020, pp. 492–502.
[44] K. Shu, D. Mahudeswaran, S. Wang, H. Liu, Hierarchical propagation networks for fake news detection: Investigation and exploitation, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 14, 2020, pp. 626–637.
[45] C. Song, K. Shu, B. Wu, Temporally evolving graph neural network for fake news detection, Inf. Process. Manage. 58 (6) (2021) 102712.
[46] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, F. Zhang, Information diffusion prediction via recurrent cascades convolution, in: 2019 IEEE 35th International Conference on Data Engineering, in: ICDE, vol. 19, 2019, pp. 770–781.
[47] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, F. Zhang, Information cascades modeling via deep multi-task learning, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, vol. 19, 2019, pp. 885–888.
[48] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, in: ICLR, vol. 17, 2017.
[49] Z. He, C. Li, F. Zhou, Y. Yang, Rumor detection on social media with event augmentations, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, vol. 21, 2021, pp. 2020–2024.
[50] X. Chen, F. Zhou, F. Zhang, M. Bonsangue, Catch me if you can: A participant-level rumor detection framework via fine-grained user representation learning, Inf. Process. Manage. 58 (5) (2021) 102678.
[51] Y. Dou, K. Shu, C. Xia, P.S. Yu, L. Sun, User preference-aware fake news detection, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR, vol. 21, 2021, pp. 2051–2055.
[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, in: CVPR, vol. 16, 2016, pp. 770–778.
[53] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, in: NIPS, vol. 16, 2016, pp. 3844–3852.

[54] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations, in: ICLR, vol. 18, 2018.
[55] Y. Xie, C. Yao, M. Gong, C. Chen, A. Qin, Graph convolutional networks with multi-level coarsening for graph classification, Knowl.-Based Syst. 194 (2020) 105578.
[56] Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: International Conference on Learning Representations, in: ICLR, vol. 20, 2020.
[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st Conference on Neural Information Processing Systems, in: NIPS, vol. 17, 2017.
[58] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[59] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, D. Cai, What to do next: Modeling user behaviors by time-LSTM, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, in: IJCAI, vol. 17, 2017, pp. 3602–3608.
[60] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, J. Gao, Eann: Event adversarial neural networks for multi-modal fake news detection, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in: KDD, vol. 18, 2018, pp. 849–857.
[61] S. Schwarz, A. Theóphilo, A. Rocha, Emet: Embeddings from multilingual-encoder transformer for fake news detection, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, in: ICASSP, vol. 20, 2020, pp. 2777–2781.
[62] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: Proceedings of the 31st Conference on Neural Information Processing Systems, in: NIPS, vol. 17, 2017.
[63] Y. Zhou, R. Ji, J. Su, X. Sun, W. Chen, Dynamic capsule attention for visual question answering, in: Proceedings of the AAAI Conference on Artificial Intelligence, in: AAAI, vol. 19, 2019, pp. 9324–9331.
[64] X. Liu, Q. Chen, Y. Liu, J. Siebert, B. Hu, X. Wu, B. Tang, Decomposing word embedding with the capsule network, Knowl.-Based Syst. 212 (2021) 106611.
[65] Z. Liu, C. Xiong, M. Sun, Z. Liu, Fine-grained fact verification with kernel graph attention network, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, in: ACL, vol. 20, 2020, pp. 7342–7351.
[66] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, 2019, arXiv preprint arXiv:1908.03265.
[67] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[68] L.M.S. Khoo, H.L. Chieu, Z. Qian, J. Jiang, Interpretable rumor detection in microblogs by attending to user interactions, in: Proceedings of the AAAI Conference on Artificial Intelligence, in: AAAI, vol. 20, 2020, pp. 8783–8790.
[69] Q. Huang, C. Zhou, J. Wu, L. Liu, B. Wang, Deep spatial–temporal structure learning for rumor detection on Twitter, Neural Comput. Appl. (2020) 1–11.
[70] K. Shu, S. Wang, H. Liu, Beyond news contents: The role of social context for fake news detection, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, in: WSDM, vol. 19, 2019, pp. 312–320.
[71] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, in: NAACL, vol. 19, 2019, pp. 4171–4186.
[72] D.Q. Nguyen, T. Vu, A. Tuan Nguyen, BERTweet: A pre-trained language model for english tweets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, in: EMNLP, vol. 20, 2020, pp. 9–14.
[73] M. Cheng, S. Nazarian, P. Bogdan, Vroc: Variational autoencoder-aided multi-task rumor classifier based on text, in: Proceedings of the Web Conference 2020, in: WWW, vol. 20, 2020, pp. 2892–2898.
[74] N. Rosenfeld, A. Szanto, D.C. Parkes, A kernel of truth: Determining rumor veracity on Twitter by diffusion pattern alone, in: Proceedings of the Web Conference 2020, in: WWW, vol. 20, 2020, pp. 1018–1028.