User-Aspect Multi-View Learning with Attention for Rumor Detection

Xueqin Chen, Fan Zhou, *Member, IEEE,* Goce Trajcevski, *Member, IEEE,* and Marcello Bonsangue

***Abstract*—The emergence of online social media (OSM) has facilitated fast information diffusion – however, it also provides ideal platforms for spreading rumors and fake news that could bring about the detrimental impact and consequences on politics, economics, and public health, etc. Researchers, enterprises, and even governments have made great efforts to detect misinformation promptly and accurately. Traditional solutions either examine complicated hand-crafted features (e.g., textual and structural 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 discovery of diffusion patterns with various neural networks. Despite the progress made by these methods, they still 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, 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 articles, and fuse the learned features thorough hierarchical attention mechanisms. Our experiments conducted on real-world Twitter 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 on user-aspect features and individual user's contributions.**

***Index Terms*—rumor detection, multi-view learning, information diffusion, dynamic routing, graph neural networks**

I. 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, 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]–[8].

A series of existing studies have focused on automatically detecting rumors, mainly falling into three categories:

(1) *Hand-crafted features based approaches* mainly focus on identifying and incorporating complicated manual features for rumor detection, such as: lexical features [9]–[11], syntactic features, [9], [12], [13], visual features [14], [15], user features [9], [16], [17], and network features [10], [18]. 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], [14], [19] 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 makes 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 [20], [21], visual features [22], and structural features [23] from the rumor contents and

Fan Zhou is the corresponding author.

This work was supported by Sichuan Regional Innovation Cooperation Project (Grant No. 2020YFQ0018), National Natural Science Foundation of China (Grant No.61602097), and NSF grants CNS 1646107.

Xueqin Chen is with University of Electronic Science and Technology of China and Leiden University. E-mail:nedchen0728@gamil.com

Fan Zhou is with University of Electronic Science and Technology of China. E-mail: fan.zhou@uestc.edu.cn

Goce Trajcevski is with Iowa State University. E-mail: gocet25@iastate.edu

Marcello Bonsangue is with Leiden University. E-mail: m.m.bonsangue@liacs.leidenuniv.nl

[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.

propagation. Despite the significant progresses, current deep learning based methods still confront several limitations:

(L1) *Overemphasis of the comment features*: Most of the current works focus more on the contents of comments to identify the rumor articles [21], [24], which is inadequate for early-stage detection as most users tend to retweet the source tweet with very few (if any) comments.

(L2) *Lack of modeling hierarchical diffusion*: Existing studies either pay attention to the micro-level of rumor diffusion, – e.g., capturing local structure correlations among propagated users and the sequential propagation patterns [25]; or focus on a macro-level spread – e.g., use graph representation techniques such as graph neural networks to learn the global structure of rumor diffusion [24]. However, few studies have unified the two-level structure knowledge for consistent propagation patterns learning in a unified framework.

(L3) *Absence of systematic user-aspect information fusion*: Users are significant contributors to the spread of rumors, and the recent research usually aggregates users' profile information (or only their engagements) to infer the types of news articles and tweets [9], [26]–[28]. These approaches model the rumors diffusion at an event level without properly taking into account of the users' behavior and roles in the process of rumor propagation. In contrast, recent research [29] reveals that humans are the principal "culprits" in spreading the false news mainly because people often prefer fake news rather than true information.

(4) *Indistinguishable importance of features and users*: Different features play different roles in rumor detection at different phases of the propagation. For example, as the spread of information increases, the role of structural information and temporal information on detecting rumors is different [26], [30]. Also, users may either unconsciously forward some unproven rumors, or deliberately propagate the fake news with different importance in the information spread [16]. Understanding the importance of features and individual users would help in better detecting rumors – which has not been investigated enough in the previous works.

To address the limitations L1–L4 above, we propose the *User-aspect Multi-view Learning with Attention for Rumor Detection* (UMLARD)[2] – a novel framework for rumor detection, which incorporates different views of the users engaged in the diffusion process to predict the credibility of a given post. UMLARD extracts the information from the hierarchical diffusion process of a given post (i.e., the full diffusion network and the local diffusion path); user profiles; and source tweet content for rumor detection. As depicted in Fig. 1, it uses an attention-based layer, a multi-layer diffusion graph convolutional network, and a time-decay LSTM to learn the high-level representations from the different views of users – i.e., the profile-view, structural-view and temporal-view. To understand the importance of each view and the role of the user, we design a view-wise attention network and a capsule attention network, which fuse both view-level and user-level features to better identify which types of features are important in rumor propagation.

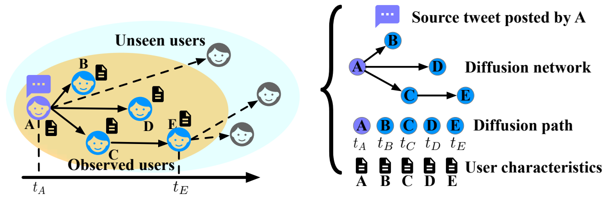


Fig. 1. An example of the extracted information from a tweet diffusion.

Our main contributions towards rumor detection problem provide:

- **Hierarchical structure learning & User-aspect feature extraction**: To the best of our knowledge, this is the first work to represent the information diffusion hierarchically and extract the related features from user-aspect – i.e., user structural features, user temporal features, and user profile features.
- **Efficient embedding methods**: UMLARD leverages deep learning to learn latent semantics in an end-to-end manner, avoiding intensive and complex feature engineering. It utilizes different embedding methods for different inputs: (1) an attention-based layer assigning different importance to features in user profiles; (2) an improved GCN-based network to learn structural information from the diffusion tree while considering the direction of the information dissemination; and (3) a time-decay LSTM considers the influence of users that decays with the time.
- **Hybrid multi-view fusion**: We design a hybrid multi-view fusion mechanism to unify the knowledge from different perspectives, which consists of two components: (1) view-wise attention layer for fusing features from different views; and (2) capsule attention layer to choose the most related users.
- **Explainable prediction results**: UMLARD can explain 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 different perspectives of the user 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.
- **Evaluations on real-world datasets**: We conduct extensive evaluations on two benchmark Twitter 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.

Organization: In the rest of this paper, Section II reviews the related work. In Section III, we formalize the problem and presents the necessary background, and then discuss the details of UMLARD. The results of the experimental evaluations quantifying the benefits of our approach are presented in Section IV. We conclude the article and outline directions for future work in Section V.

[2] We release the code of our model for other users to reproduce and compared to our method: https://github.com/ChenNed/UMLARD.

II. 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 [31], marketing [32], healthcare [15], 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 [29]. Existing methods for rumor detection generally fall into three categories, i.e., feature-based, credibility-based, and deep learning-based approaches.

A. Hand-crafted Features-based Approaches

Most of earlier works extract 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 [9]–[11], syntactic features [9], [12], [13], topic features [33], visual statistical features [14], [15], and visual content features [34]; 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 [9], [16], [17], propagation features [10], [30], [33], and temporal features [10], [26]. 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. [11] discover two types of language patterns in rumors, i.e., inquiry and correction patterns, and detect the patterns of rumor messages through supervised feature selection on a set of labeled messages. Wu et al. [33] define a set of topic features to summaries semantics and train 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 [15] and emotions [9]. As for visual content features, Jin et al. [34] find 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. [9] first identify the credibility of tweets in Twitter, and Kwon et al. [10] extends it by proposing 15 structural features extracted from the diffusion network and the user friendship network. In the work [26], 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. [27] 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.

B. 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. [14] 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. [19] 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.

C. Deep Learning-based approaches

The recent success of deep learning in NLP (Natural Language Processing) and CV (Computer Vision) communities spurs a few deep rumor detection methods. These models have shown improved performance over traditional approaches due to their enhanced ability to automatically representation learning. Ma et al. [20] use recurrent neural networks (RNN) to model the rumors as time-series data, and learn temporal and textual features from the raw data. Later, they improved [20] by building a tree-structured RNN to catch the hidden representations from both propagation structures and text contents [30]. Jin et al. [22] proposed a model which extracts the visual, textual, and social context features, and fuses them by attention mechanism. Yu et al. [35] introduced a convolutional neural network (CNN)-based approach to handling the issues in RNN-based methods, i.e., RNN is not qualified to the early-stage detection with limited inputting data and has a bias towards the latest elements of the input sequence. There are also plenty of methods that combine the merits of CNN and RNN for rumor detection. For example, Liu et al. [23] built a time series classifier with both RNN and CNN to predict whether a given news story is fake at an early stage, which takes common user characteristics and propagation paths into consideration.

Recently, graph neural networks have emerged as the standard models for graph learning and spurred many works to learn the structural information diffusion patterns [24], [36]–[38]. For example, Bian et al. [24] propose a graph convolutional network (GCN [39])-based model that can learn global structural relationships of rumor dispersion. Similarly, Lu et al., [28] improved the work of [23] by calculating the similarity between users and used a graph-aware attention network for rumor detection. Khattar et al. [40] proposed a multimodal variational autoencoder method that learns a shared representation of texts and images for rumor detection. Moreover, a few works focus on multiple propagation-related learning simultaneously [37], [41], e.g., Ma et al. [41] proposed two multi-task architectures based on RNNs, which jointly trains the task of stance classification and rumor detection. While achieving enhanced performance over feature-based and credibility-based approaches, current deep learning-based methods are event-driven and are still highly dependent on the content features of online articles. In contrast, our method focus on learning the user-aspect features since users are the main contributors in the spread of information [36], [42]. Besides, our model provides explainable results neglected in most previous works.

III. UMLARD: Model, Approach and Properties

In this section, after introducing the preliminaries and basic notation, we formalizing the problem and with detailed discussions of the main components of UMLARD.

As illustrated in Fig. 2, 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) *Hybrid 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 , based on the learned user-aspect knowledge and tweet content.

A. 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.

TABLE I
List of Notations

Symbol	Description
$\mathcal{M}$	a set of tweets/posts.
M_i	a specific tweet/post.
$\mathcal{G}_i$	diffusion network of tweet M_i.
$\mathcal{P}_i$	diffusion path of tweet M_i.
$\mathbf{U}_i$	user characteristic matrix of tweet M_i.
$\mathbf{C}_i$	source tweet content vector of tweet M_i.
U_i, E_i	user set and edge set of tweet M_i.
$T, t_*, \|U_i\|$	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.
$\hat{\mathbf{Y}}/\hat{\mathbf{y}}_*, \mathbf{Y}/\mathbf{y}_*$	the predicted label and the ground truth.

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}.*

We note that the diffusion networks considered in our paper are directed acyclic graphs.

Diffusion Path. *A diffusion path of tweet M_i is defined as a multivariate time series $\mathcal{P}_i = \{(u_{i1}, t_{i1}), (u_{i2}, t_{i2}), \cdots, (u_{i|U_i|}, t_{i|U_i|})\}$, where $t_{i1} \leq t_{i2} \leq \ldots < t_{i|U_i|}$. 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 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. 1 for the corresponding example. The figure also illustrates two more important components, which we describe next.

User Characteristic Matrix. *Each user $u_{ij} \in U_i$ is associated with a user vector $\boldsymbol{p}_{ij} \in \mathbb{R}^{d_{user}}$, 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 $\boldsymbol{U}_i \in \mathbb{R}^{|U_i| \times d_{user}}$, 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 $\boldsymbol{C}_i$ is considered to be a sequence of words – i.e., $\boldsymbol{C}_i = [\boldsymbol{w}_{i1}, \boldsymbol{w}_{i2}, \ldots, \boldsymbol{w}_{iL}] \in \mathbb{R}^{L \times d_{word}}$, where L is the number of words in source tweet.*

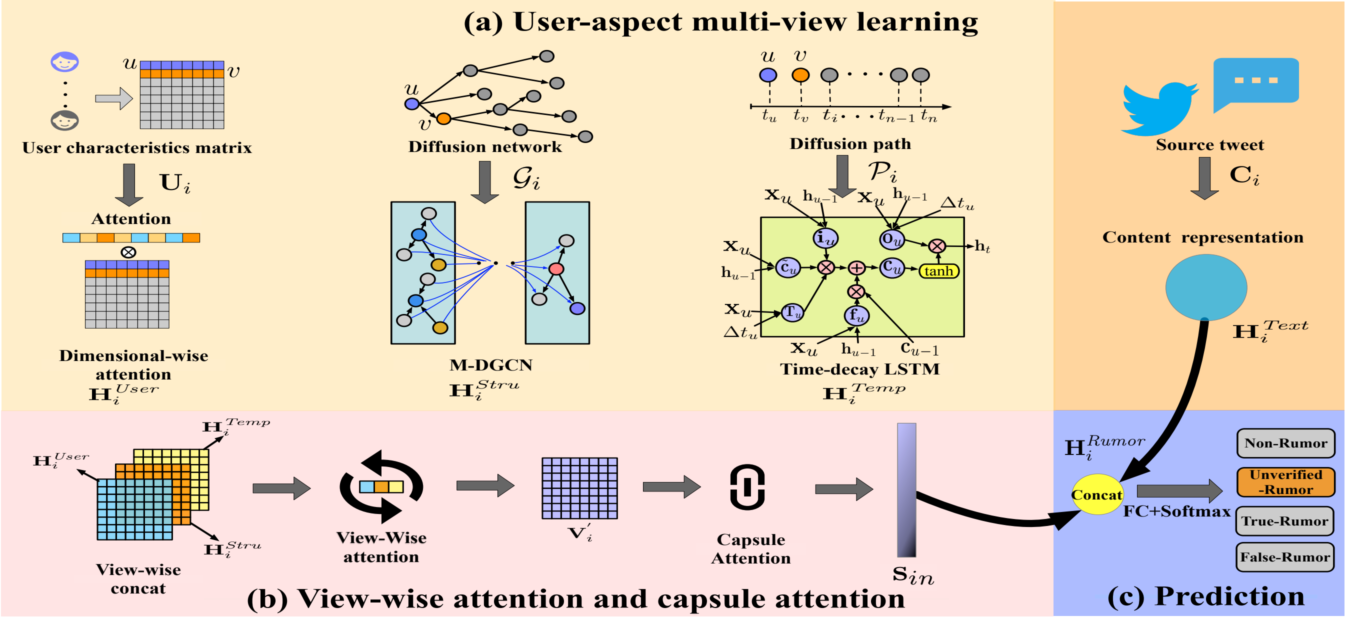


Fig. 2. An overview of UMLARD. (a) The inputs of UMLARD are the observed diffusion network, the diffusion path, the user characteristic matrix, and the content of the source tweet. It uses a dimensional-wise attention layer, a multi-layer diffusion graph convolutional network (M-DGCN), and a time-decay 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.

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 I. 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* $M_i = \{\mathcal{G}_i, \mathcal{P}_i, \boldsymbol{U}_i, \boldsymbol{C}_i\}$ *within an observation window* T*, our rumor detection goal is to learn a function* f *from labeled claims, i.e.,* $f(\hat{\boldsymbol{y}}_i|\mathcal{G}_i, \mathcal{P}_i, \boldsymbol{U}_i, \boldsymbol{C}_i; T)$*, where the predicted result* $\hat{\boldsymbol{y}}_i$ *takes one of the four finer-grained classes: non-rumor, false rumor, true rumor, and unverified rumor (as introduced in [30]).*

B. Learning Users Profile-View

User profiles have been demonstrated to be strong indicators when detecting fake news [16], [17]. 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 (cf. Table. III):

- **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 regard 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:

$$\begin{aligned} \mathbf{f}^1 &= \tanh(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1), \\ \mathbf{f}^2 &= \text{softmax}(\mathbf{W}_2 \mathbf{f}^1 + \mathbf{b}_2), \end{aligned} \tag{1}$$

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:

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

where $\mathbf{H}_i^{User} \in \mathbb{R}^{|U_i| \times d_{user}}$.

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 Sec. IV.

C. Learning Users Structural-View

The structural information of users who participate in spreading a tweet can be considered as an information cascade. Inspired by the recent successes of network representation learning methods in processing graph-structured data [36], [43], 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 l-th 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 diffusion networks [36], [43] 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 it's 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. [44]) 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 [30] it has been demonstrated that the rumors have similar propagation patterns.

D. Learning Temporal-View of Users

Users' engagement time and the sequential patterns of retweets also play an essential role in detecting rumors [20], [21], [25]. To capture this view of users, we sample the diffusion path from the multivariate time series based on the retweet time. 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 [45], we obtain the static-embedding $\mathbf{e}_j^s$ using a positional-encoding technique based on sine and cosine functions of frequencies:

$$\mathbf{PE}(j)_{2d} = \sin(j/10000^{2d/d_e}), \tag{4}$$

$$\mathbf{PE}(j)_{2d+1} = \cos(j/10000^{2d/d_e}), \tag{5}$$

  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 [45].
- **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 $\mathbf{e}_j^d = \mathbf{q}\mathbf{E}, \mathbf{e}_j^d \in \mathbb{R}^{d_e}$. 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 [46]) 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 [47] 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:

$$\begin{aligned} \mathbf{x}_j &= \mathbf{e}_j^s + \mathbf{e}_j^d, \\ \mathbf{i}_j &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_j + \mathbf{U}_{hi}\mathbf{h}_{j-1} + \mathbf{b}_i\right), \\ \mathbf{f}_j &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_j + \mathbf{U}_{hf}\mathbf{h}_{j-1} + \mathbf{b}_f\right), \\ \mathbf{T}_j &= \sigma\left(\mathbf{W}_{xT}\mathbf{x}_j + \tanh\left(\mathbf{W}_{tt}\Delta t_j\right) + \mathbf{b}_T\right), \\ \mathbf{o}_j &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_j + \mathbf{U}_{ho}\mathbf{h}_{j-1} + \mathbf{W}_{to}\Delta t_j + \mathbf{b}_o\right), \\ \widetilde{\mathbf{c}}_j &= \tanh\left(\mathbf{W}_{xz}\mathbf{x}_j + \mathbf{U}_{hz}\mathbf{h}_{j-1} + \mathbf{b}_z\right), \end{aligned} \tag{6}$$

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:

$$\mathbf{c}_j = \mathbf{f}_j \odot \mathbf{c}_{j-1} + \mathbf{i}_j \odot \mathbf{T}_j \odot \widetilde{\mathbf{c}}_j, \tag{7}$$

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

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh\left(\mathbf{c}_j\right), \tag{8}$$

Finally, the representation vector for the temporal-view is $\mathbf{H}_i^{Temp} = \left\{\mathbf{h}_1^{Temp}, \mathbf{h}_2^{Temp}, \cdots, \mathbf{h}_{|U_i|}^{Temp}\right\}$, 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.

E. View-Wise Attention for View-Level Feature Fusion

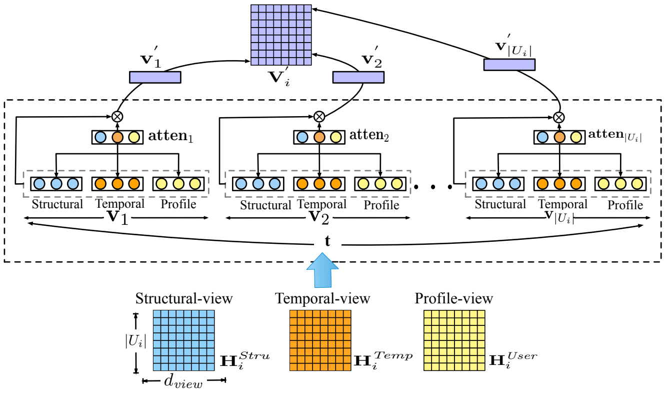


Fig. 3. Illustration of view-wise attention.

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 [22], [48], [49], we present a method to capture the differences 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, we first normalize the dimensions of the three views' vectors as d_{view}. 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 user-level feature set. The attention $\mathbf{atten}_j$ for each row $\mathbf{v}_j$ can be computed as:

$$\mathbf{v}_j^{tran} = \tanh\left(\mathbf{w}_j \cdot \mathbf{v}_j\right), \tag{9}$$

$$\mathbf{atten}_j = \text{softmax}\left(\mathbf{w}_j^{tran} \cdot \mathbf{v}_j^{tran}\right), \tag{10}$$

where $\mathbf{w}_j \in \mathbb{R}^{d_{view} \times d_{view}}$, $\mathbf{w}_j^{tran} \in \mathbb{R}^{d_{view}}$, and $\mathbf{atten}_j \in \mathbb{R}^{1 \times 3}$. The fused multi-view feature vector $\mathbf{v}_j^{'}$ for each user can be calculated as $\mathbf{v}_j^{'} = \mathbf{v}_j \cdot \mathbf{atten}_j^T$, where $\mathbf{v}_j^{'} \in \mathbb{R}^{d_{view}}$. Finally, we get the fused multi-view feature vector matrix as $\mathbf{V}_i^{'} = \left\{\mathbf{v}_1^{'}, \mathbf{v}_2^{'}, \cdots, \mathbf{v}_{|U_i|}^{'}\right\}$, where $\mathbf{V}_i^{'} \in \mathbb{R}^{|U_i| \times d_{view}}$.

F. Capsule Attention for User-level Feature Fusion

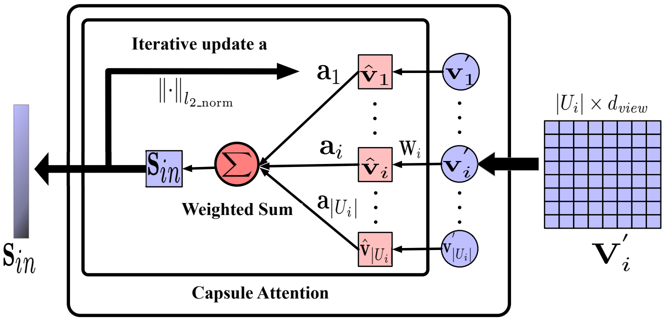


Fig. 4. Capsule Fusion Layer.

Most of existing works [24], [25], [28] 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 [50], [51]. The Capsule network was first proposed in [50] 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 n-th iteration, the upper capsule $\mathbf{s}_{in}$ is calculated by:

$$\mathbf{s}_{in} = \sum_{j}^{|U_i|} \mathbf{a}_j \hat{\mathbf{v}}_j, \hat{\mathbf{v}}_j = \mathbf{W}\mathbf{v}_j^{'}, \tag{11}$$

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. $\mathbf{W} \in \mathbb{R}^{d_{view} \times d_a}$ 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 L_2 regularizer $\widetilde{\mathbf{s}}_{in} = \|\mathbf{s}_{in}\|_{l_{2\_\text{norm}}} = \mathbf{s}_{in} / \|\mathbf{s}_{in}\|$ in $\mathbf{s}_{in}$ to overcome the information loss caused by the original CapsAtt [51].

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 j-th user capsule should be coupled to the upper capsule $\mathbf{s}_{in}$. The coefficient is calculated by:

$$\mathbf{a}_j = \frac{\exp(\mathbf{b}_j)}{\sum_k^{|U_i|} \exp(\mathbf{b}_k)}, \tag{12}$$

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

$$\mathbf{b}_j = \mathbf{b}_j + \hat{\mathbf{v}}_j \cdot \widetilde{\mathbf{s}}_{in}, \tag{13}$$

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.

G. Tweet Content Representation

Tweet content is one of the most important features in rumor detection [9]–[11], and has been extensively studied in the literature [13], [20], [24], [28], [52], 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:

$$\mathbf{h}_m = \sigma(\mathbf{W} * \mathbf{w}_{m:m+d-1}), \tag{14}$$

where $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{L-d+1}\}$ 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 filed, 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 $\mathbf{H}_i^{Text}$.

H. Training Objective

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

$$\mathbf{H}_i^{Rumor} = \text{concat}(\mathbf{H}_i^{Text}, \mathbf{s}_{in}) \tag{15}$$

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:

$$\hat{\mathbf{y}}_i = \text{softmax}\left(\text{FC}\left(\mathbf{H}_i^{Rumor}\right)\right). \tag{16}$$

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

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = -\sum_{i \in |\mathcal{M}|} \mathbf{y}_i \log \hat{\mathbf{y}}_i + \lambda \|\Theta\|_2^2, \tag{17}$$

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* [53] as optimizer.

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.(6) - Eq.(8);
        Content representation: $\mathbf{H}_i^{Text} \leftarrow \mathbf{C}_i$ via Eq.(14);
        Nomalize 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.(9);
        Capsule attention learning: $\mathbf{s}_{in} \leftarrow \mathbf{V}_i^{'}$ via Eq.(11);
        Merge $\mathbf{H}_i^{Text}$ and $\mathbf{s}_{in}$ via Eq.(15);
        Estimate the probability $\hat{\mathbf{y}}_i$ via Eq.(16);
        Compute loss $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}_i})$, via Eq.(17);
        Update parameters using RAdam.
    end for
until convergence;
```

I. 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., $\mathbf{D}_I^{-1}\mathbf{A}$ and $\mathbf{D}_O^{-1}\mathbf{A}^T$. 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 [46]. 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 l-th layer; output size is $M_l = (X_l - K_l)/\mathit{Stride} + 1$ 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 hybrid 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.

IV. Experiments

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

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

- **RQ1**: How does UMLARD perform on rumor detection compared 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?

TABLE II
Statistics of the datasets.

Statistic	Twitter15	Twitter16
# source tweets	1,482	809
# users	477,009	286,657
# non-rumors	370	199
# false-rumors	369	205
# true-rumors	372	207
# unverified-rumors	371	198
Max. # retweets	2989	3058
Min. # retweets	55	73
Avg. # retweets	398	422
Avg. # time length	1,268 Hours	828 Hours

A. Experimental Settings

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

1) Datasets: We conducted our experiments on two real-world datasets[3]: *Twitter15* and *Twitter16*, which were collected in [30] from one of the most popular social media, Twitter. In each dataset, a group of widespread source tweets along with their propagation threads with time stamps are provided. We constructed propagation paths and diffusion networks from the propagation threads, which are also used for user temporal-aspect embedding and user structural-aspect embedding.

Each source tweet is annotated with one of the four class labels, i.e., *non-rumor*, *false-rumor*, *true-rumor*, and *unverified-rumor* – the labeling rules follow the method in [20]. The statistics of the two datasets are shown in Table II. Fig. 5(a) shows that the message propagation speed is gradually saturated in 24 hours. Fig. 5(b) and 5(c) show the propagation speed for different types of messages within 1 hour on the two datasets, respectively.

TABLE III
Summary of user characteristics.

No.	Characteristic	Data Type
1	LENGTH OF USER NAME	Integer
2	USER COUNT CREATED TIME	Integer
3	LENGTH OF DESCRIPTION	Integer
4	FOLLOWERS COUNT	Integer
5	FRIENDS COUNT	Integer
6	STATUSES COUNT	Integer
7	IS VERIFIED	Binary
8	IS GEO-ENABLED	Binary

Due to the constraints of the Twitter service terms, the original datasets do not contain user profile information. We crawled all the related user profiles via *Twitter API*[4] based on the provided user IDs. From the crawled user profiles, we extract eight user characteristics, as shown in Table III.

Following previous works [21], [24], [28], we randomly choose 70% data for training, 10% data for validation, and the remaining data for testing.

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

- **DTC** [9]: A decision tree-based classification model that combines manually engineered characteristics of tweets to compute the information credibility.
- **SVM-RBF** [27]: 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** [26]: A linear SVM-based timesseries 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** [54]: A variant of the RNN with the gated recurrent units that has been employed in [20] to learn the sequential cascading effect of tweets with high-level feature representations extracted from relevant posts over time.
- **TD-RvNN** [21]: 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** [25]: 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.
- **Bi-GCN** [24]: A GCN-based model exploiting the bi-directional propagation structures and text contents for rumor detection.
- **GCAN** [28]: A co-attention network that detects true and false rumors based on the content of the source tweet and its propagation-based users.

3) Implementation details: We implemented DTC with Weka[5], SVM-based models with scikit-learn[6], and other neural network-based models with Tensorflow[7]. All baselines follow the parameter settings in the original papers.

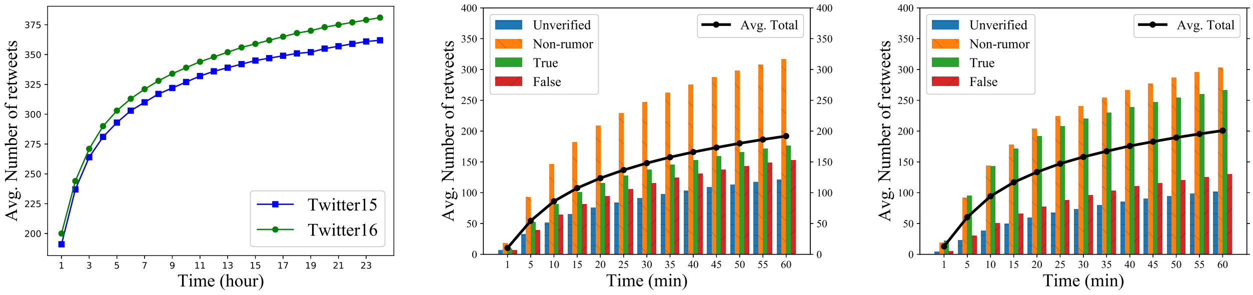
[3] https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0

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

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

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

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

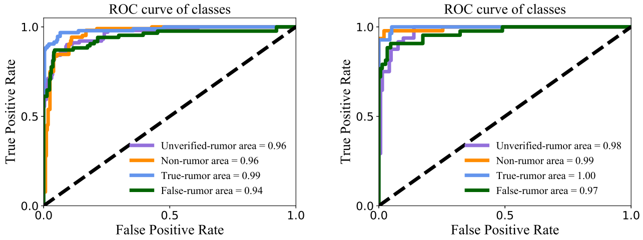


(a) Avg. 24 hours (b) Twitter15 (c) Twitter16

Fig. 5. Datasets analysis. (a) The scale of the information cascades in Twitter16 is slightly larger than in Twitter15. (b) and (c) The number of rumors on Twitter is definitely increasing.

TABLE IV
OVERALL PERFORMANCE COMPARISON OF RUMOR DETECTION ON TWITTER15 AND TWITTER16. "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 $p < 0.001$ COMPARED TO THE BEST BASELINE METHOD.

Datasets	Metric		SVM-TS	SVM-RBF	DTC	GRU	TD-RvNN	PPC_RNN+CNN	Bi-GCN	GCAN	UMLARD
Twitter15	ACC.		0.544	0.318	0.454	0.646	0.723	0.697	0.829	0.808	**0.857***
	UR	F1	0.483	0.218	0.415	0.608	0.654	0.689	0.752	0.69	**0.835***
	NR	F1	0.796	0.225	0.733	0.592	0.682	0.760	0.772	**0.93**	0.84*
	TR	F1	0.404	0.455	0.317	0.792	0.821	0.696	0.885	0.812	**0.906***
	FR	F1	0.472	0.082	0.355	0.574	0.758	0.645	0.847	0.758	**0.848***
Twitter16	ACC.		0.574	0.321	0.465	0.633	0.737	0.702	0.837	0.765	**0.901***
	UR	F1	0.526	0.419	0.403	0.686	0.708	0.608	0.818	0.784	**0.822***
	NR	F1	0.755	0.037	0.643	0.593	0.662	0.711	0.772	0.848	**0.965***
	TR	F1	0.571	0.423	0.419	0.772	0.835	0.816	0.885	0.678	**0.960***
	FR	F1	0.420	0.085	0.393	0.489	0.743	0.664	0.847	0.754	**0.855***



(a) Twitter15 (b) Twitter16

Fig. 6. ROC curve comparison for each information type. Area under curve of ROC (AUC) is presented after the legend.

For UMLARD, the learning rate is initialized at 0.001 and gradually decreases as the training proceeds. We initialized 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 the capsule size; and the iteration number varies between 2 and 4. The batch size is 64; the dropping rate in DropEdge is 0.2; and the rate of dropout in the main neural networks is 0.5. The training process is iterated upon for 200 epochs, but would be stopped earlier if the validation loss does not decrease after 10 epochs.

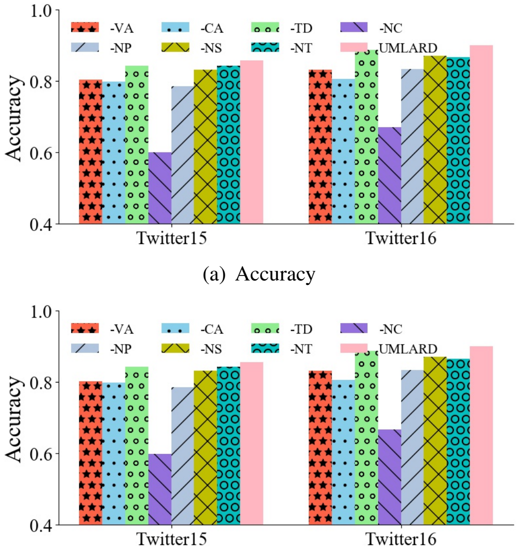
4) Evaluation metrics: We use accuracy (ACC) and F-measure (F1) as the evaluation protocols to measure the models' performance. Specifically, ACC measures the proportion of correctly classified tweets, while F1 is the harmonic mean of the precision and recall values averaged across four classes.

B. Overall Performance (RQ1)

Table IV reports the performance comparison among UMLARD and baselines on Twitter15 and Twitter16 datasets, from which we have the following observations:

O1: Feature-based approaches such as SVM-TS, SVM-RBF, and DTC perform poorly. These methods used 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, because it selects the features based on Weibo (a Chinese microblog platform), which are hard to be generalized to other social platforms such as Twitter-based ones used here. SVM-TS achieved 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. Both Bi-GCN and GCAN consider structural information, and their performance indeed exceeds other baseline methods. In particular, Bi-GCN constructs the structural tree based on the replies, i.e., the retweets with comments, which can not reflect the whole structure of rumor dispersion. In contrast, GCAN models the structural information form the user similarity matrix, which can not capture the full process of rumor diffusion. According to the results, Bi-GCN performs much better than GCAN, because it takes the comments information into consideration. Besides, the bi-direction GCN is more effective in learning propagation structures than vanilla GCN used in GCAN.



(a) Accuracy

(b) Macro-F1

Fig. 7. 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.

O3: UMLARD consistently outperforms all the other baselines on both datasets. Compared to the best baseline method Bi-GCN, UMLARD models rumor diffusion from multiple users' perspectives and discriminate the importance of features and users in spreading the tweets. These results demonstrate the validity of the observation that was one of the primary motivations of this work – i.e., that different 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. Fig. 6 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. 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.

C. Ablation Experiments (RQ2)

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

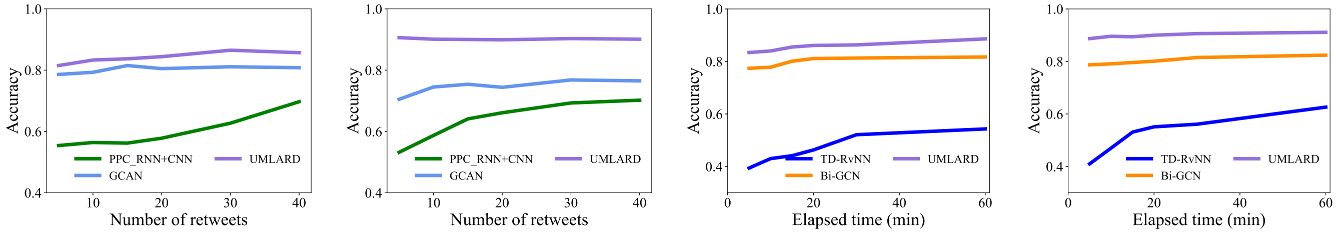
- **-VA**: In -VA, we do not consider the different importance for different views, i.e., we remove the view-wise attention layer in UMLARD.
- **-CA**: In -CA, we replace the capsule attention layer with a fully connected layer.
- **-TD**: In -TD, we do not consider the time decay effect of retweet behaviors. Instead, we use vanilla LSTM [46] to learn sequential retweet behavior.
- **-NC**: In -NC, we do not use the content feature of source tweet but keep the temporal, profile, and structural feature.
- **-NP**: In -NP, we do not consider the profile features of users but keep temporal, structural, and content feature.
- **-NS**: In -NS, we do not consider the structural features of users but keep temporal, profile, and content feature.
- **-NT**: In -NT, we do not consider the temporal features of users but keep structural, profile, and content feature.

Fig. 7 illustrates the performance of the variants, where we can observe that: (1) The content of tweet (**-NC**) 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 [21], [55]. (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 [16], [17]. (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 [25], and [24] 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. (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. (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 (cf. Fig. 5).

D. 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.

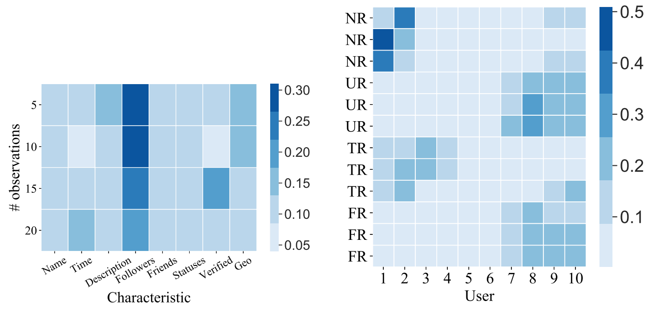
Fig. 8 shows the performance comparison on early-stage detection between our UMLARD and the baselines. Note that we omitted the feature-based methods and credibility-based approaches since they did not show comparable performance, especially on early rumor detection. We observe that UMLARD performs better, especially when there are only a few



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

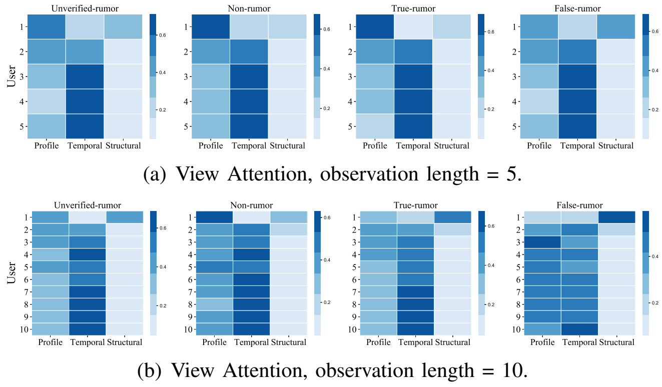
Fig. 8. Evaluations on early rumor detection. (a) and (b): PPC_RNN+CNN and GCAN are cascade length-based methods. (c) and (d): Tv-RvNN and Bi-GCN are built on the user comments that may not exist in early-stage retweets – hence, we observe their performance over time.

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 (cf. Fig. 5). 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 contrast, the other methods either rely on tweet content and profile information (e.g., PPC_RNN+CNN and GCAN) or learn structural and temporal information of cascades (e.g., TD-RvNN and Bi-GCN), which, however, are insufficient for early-stage misinformation detection.



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

Fig. 9. 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. 10. Visualization of the different user-aspect importance.

E. 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. 9 shows the importance of user-profiles and users themselves – the higher the value, the more important the feature or the user. Fig. 9(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 [16], [16], [17], i.e., the users enrolled in spreading of rumors have fewer followers.

In Fig. 9(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* (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. Fig. 10(a) and Fig. 10(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 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. 9(b).

V. Conclusions and Future 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 we considered a simple content representation approach in UMLARD, that 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 [52]. 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 [56]. 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 [57] 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 of further examination.

References

[1] Z. Jin, J. Cao, Y. Zhang, and J. Luo, "News verification by exploiting conflicting social viewpoints in microblogs," in *AAAI*, 2016.

[2] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of economic perspectives*, 2017.

[3] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, "Adaptive hidden markov model with anomaly states for price manipulation detection," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 2, pp. 318–330, 2014.

[4] W. Ahmed, J. Vidal-Alaball, J. Downing, and F. López Seguí, "Covid-19 and the 5g conspiracy theory: Social network analysis of twitter data," *J Med Internet Res*, 2020.

[5] Y. Lin, X. Wang, F. Hao, Y. Jiang, Y. Wu, G. Min, D. He, S. Zhu, and W. Zhao, "Dynamic control of fraud information spreading in mobile social networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2019.

[6] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Game theoretic suppression of forged messages in online social networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2019.

[7] B. Chang, E. Chen, F. Zhu, Q. Liu, T. Xu, and Z. Wang, "Maximum a posteriori estimation for information source detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 6, pp. 2242–2256, 2020.

[8] L. Ding, P. Hu, Z. Guan, and T. Li, "An efficient hybrid control strategy for restraining rumor spreading," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2020.

[9] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *WWW*, 2011.

[10] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *ICDM*, 2013.

[11] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *WWW*, 2015.

[12] A. Hassan, V. Qazvinian, and D. Radev, "What's with the attitude?: identifying sentences with attitude in online discussions," in *EMNLP*, 2010.

[13] B. Ma, D. Lin, and D. Cao, "Content representation for microblog rumor detection," in *Advances in Computational Intelligence Systems*. Springer, 2017.

[14] M. Gupta, P. Zhao, and J. Han, "Evaluating event credibility on twitter," in *SDM*, 2012.

[15] Z. Zhang, Z. Zhang, and H. Li, "Predictors of the authenticity of internet health rumours," *Health Information & Libraries Journal*, 2015.

[16] K. Shu, S. Wang, and H. Liu, "Understanding user profiles on social media for fake news detection," in *MIPR*, 2018.

[17] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profiles for fake news detection," in *ASONAM*, 2019.

[18] Y. Yang, K. Niu, and Z. He, "Exploiting the topology property of social network for rumor detection," in *JCSSE*, 2015.

[19] Z. Jin, J. Cao, Y.-G. Jiang, and Y. Zhang, "News credibility evaluation on microblog with a hierarchical propagation model," in *ICDM*, 2014.

[20] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *IJCAI*, 2016.

[21] J. Ma, W. Gao, and K. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *ACL*, 2018.

[22] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *MM*, 2017.

[23] Y. Liu and Y. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *AAAI*, 2018.

[24] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *AAAI*, 2020.

[25] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *AAAI*, 2018.

[26] J. Ma, W. Gao, Z. Wei, Y. Lu, and K. Wong, "Detect rumors using time series of social context information on microblogging websites," in *CIKM*, 2015.

[27] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *SIGKDD*, 2012.

[28] Y.-J. Lu and C.-T. Li, "Gcan: Graph-aware co-attention networks for explainable fake news detection on social media," *arXiv preprint arXiv:2004.11648*, 2020.

[29] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, 2018.

[30] J. Ma, W. Gao, and K. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *ACL*, 2017.

[31] N. DiFonzo and P. Bordia, "Rumor and prediction: Making sense (but losing dollars) in the stock market," *Organizational Behavior and Human Decision Processes*, 1997.

[32] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *TWEB*, 2007.

[33] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *ICDE*, 2015.

[34] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE transactions on multimedia*, 2016.

[35] F. Yu, Q. Liu, S. Wu, L. Wang, T. Tan *et al.*, "A convolutional approach for misinformation identification," in *IJCAI*, 2017.

[36] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang, "Information diffusion prediction via recurrent cascades convolution," in *ICDE*, 2019.

[37] X. Chen, K. Zhang, F. Zhou, G. Trajcevski, T. Zhong, and F. Zhang, "Information cascades modeling via deep multi-task learning," in *SIGIR*, 2019.

[38] F. Zhou, X. Xu, K. Zhang, G. Trajcevski, and T. Zhong, "Variational information diffusion for probabilistic cascades prediction," in *INFOCOM*, 2020.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[40] D. Khattar, J. S. Goud, M. Gupta, and V. Varma, "Mvae: Multimodal variational autoencoder for fake news detection," in *WWW*, 2019.

[41] J. Ma, W. Gao, and K. Wong, "Detect rumor and stance jointly by neural multi-task learning," in *WWW*, 2018.

[42] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, "A survey of information cascade analysis: Models, predictions and recent advances," *arXiv preprint arXiv:2005.11041*, 2020.

[43] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *SIGKDD*, 2018, pp. 1320–1329.

[44] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *ICLR*, 2020.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to do next: Modeling user behaviors by time-lstm." in *IJCAI*, 2017.

[48] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "Eann: Event adversarial neural networks for multi-modal fake news detection," in *KDD*, 2018.

[49] S. Schwarz, A. Theóphilo, and A. Rocha, "Emet: Embeddings from multilingual-encoder transformer for fake news detection," in *ICASSP*, 2020.

[50] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *NIPS*, 2017.

[51] Y. Zhou, R. Ji, J. Su, X. Sun, and W. Chen, "Dynamic capsule attention for visual question answering," in *AAAI*, 2019.

[52] Z. Liu, C. Xiong, M. Sun, and Z. Liu, "Fine-grained fact verification with kernel graph attention network," in *ACL*, 2020, pp. 7342–7351.

[53] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[54] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS*, 2014.

[55] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," in *WSDM*, 2019.

[56] M. Cheng, S. Nazarian, and P. Bogdan, "VRoC: Variational Autoencoder-aided Multi-task Rumor Classifier Based on Text," in *WWW*, 2020, pp. 2892–2898.

[57] N. Rosenfeld, A. Szanto, and D. C. Parkes, "A Kernel of Truth: Determining Rumor Veracity on Twitter by Diffusion Pattern Alone," in *WWW*, 2020, pp. 1018–1028.