

Research Article

An End-to-End Rumor Detection Model Based on Feature Aggregation

Aoshuang Ye ^{1,2} Lina Wang ^{1,2} Run Wang ^{1,2} Wenqi Wang,^{1,2} Jianpeng Ke,^{1,2} and Danlei Wang^{1,2}

¹Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan, Hubei, China

²School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei, China

Correspondence should be addressed to Lina Wang; lnwang@whu.edu.cn and Run Wang; wangrun@whu.edu.cn

Received 27 December 2020; Revised 10 April 2021; Accepted 23 April 2021; Published 12 May 2021

Academic Editor: Hocine Cherifi

Copyright © 2021 Aoshuang Ye et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The social network has become the primary medium of rumor propagation. Moreover, manual identification of rumors is extremely time-consuming and laborious. It is crucial to identify rumors automatically. Machine learning technology is widely implemented in the identification and detection of misinformation on social networks. However, the traditional machine learning methods profoundly rely on feature engineering and domain knowledge, and the learning ability of temporal features is insufficient. Furthermore, the features used by the deep learning method based on natural language processing are heavily limited. Therefore, it is of great significance and practical value to study the rumor detection method independent of feature engineering and effectively aggregate heterogeneous features to adapt to the complex and variable social network. In this paper, a deep neural network- (DNN-) based feature aggregation modeling method is proposed, which makes full use of the knowledge of propagation pattern feature and text content feature of social network event without feature engineering and domain knowledge. The experimental results show that the feature aggregation model has achieved 94.4% of accuracy as the best performance in recent works.

1. Introduction

With the development of social networks, the amount of information increases rapidly. However, the quality of information cannot be guaranteed. Misinformation and disinformation permeate almost every corner of social networks. Therefore, how to automatically evaluate the credibility and authenticity of social media information has high research and practical value.

Detecting and identifying rumor information is one of the most important research topics in information credibility evaluation and information content security. Social psychology defines rumor as unverified or intentionally false information [1]. The spread of rumors is harmful to daily life and social stability. It may cause unexpected losses to the public and society and significantly impact public safety [2]; for example, in February 2020, a rumor about “Shuanghuanlian is the cure of COVID-19” was spread in the

Chinese social networks Weibo. The rumor led to crowds taking to the streets all night to buy Shuanghuanlian, leading to a potential risk of infection. The rapid spread of lockdown rumors in 2020 is also an indication of the destructive power of rumors.

Furthermore, many research studies like Yu et al. [3], Ma et al. [4], and Ruchansky et al. [5] implement deep learning such as convolutional neural networks (CNNs) achieved impressive progress. Nonetheless, the limitations of existing automated rumor detection methods are evident [6]. Traditional methods based on statistical learning depend heavily on feature engineering. Both data-driven feature selection methods and manual feature extraction methods based on domain knowledge are time-consuming and laborious. There are unavoidable deviations challenging to adapt to the complex and variable modern social network scene. Moreover, the deep learning method plays an innovative role in cyberspace security [7].

Nevertheless, the feature type exploited by the previous end-to-end learning models is limited. The abundant feature information can not be used effectively, which limits the effect of the model. Therefore, it is of great significance and practical value to make up for the defects of the existing rumor detection methods and study the modeling method that not only does not depend on feature engineering and domain knowledge but also has the ability to aggregate different types of features.

To overcome the shortcomings of existing rumor detection methods, this work studies the temporal feature modeling method for propagation pattern and the end-to-end model for aggregating text-content features and propagation pattern features. According to previous research, context-based text features and propagation pattern features have been proven to be useful in rumor detection. The knowledge contained in the two types of features is independent. Therefore, we try to find an effective way to combine text-content features and temporal features, which achieves the better performance of rumor detection than the single feature-dependent model.

The contributions of this paper are as follows:

- (i) We study the propagation pattern of social events that do not depend on feature engineering and domain knowledge, which overcome the limitation that the propagation pattern features are difficult to be structured as input for general machine learning models. Our work proves that the propagation pattern features can effectively detect rumors by using convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
- (ii) We design a feature aggregation model based on DNN to exploit the aggregated feature combined by propagation pattern feature and text content feature. This work makes full use of the abundant information in different types of features and solves the limitation that a single model in the traditional machine learning method is challenging to deal with the heterogeneous information.
- (iii) By setting the same deadline metrics for training data and test data, better performance on early detection of social network rumor is achieved. Furthermore, the adverse effect of the different mathematical distribution between training data and test data on the prediction is solved.

Experiments show that the proposed end-to-end rumor detection model based on feature aggregation can effectively identify the rumor on social networks. The accuracy of rumor detection is as high as 94.4%, which is the best in the existing works. In the early detection of rumor, the average accuracy on corresponding time node is higher than 90%.

The rest of this paper is organized as follows. We introduce the relevant works and background knowledge in Section 2; the modeling method of propagation pattern feature is presented in Section 3; Section 4 discusses the rumor detection model based on feature aggregation; we present the experiments and corresponding analysis in Section 5; and Section 6 concludes this paper.

2. Related Work

Guo et al. [8] conclude that deep learning-based methods try to obtain the high-level representation of false information. The representation of feature directly influences the performance on the classification model. Recently, the well-developed learning-based methods of rumor detection on current social network are mainly supervised learning. Moreover, feature fusion-based methods concentrate on combining different features to achieve better representation of data. This provides us that feature fusion is also potential to be implemented in deep learning-based methods to enhance the performance. There are four types of features in learning-based systems as follows: content-based features, propagation-based features, user-based features, and other-based features.

2.1. Features in Learning-Based Rumor Detection

2.1.1. Content-Based Features. Ratkiewicz et al. [9] utilized content-based features such as hashtags, mentions, URLs, and phrases with topological and crowd-sourced features to construct the political misinformation early detection model. Qazvinan et al. [10] showed the experiment result that content-based features outperform in the precision of predicting rumors than network-based and microblog-specific memes. Vedova et al. [11] combine content features and social-context features. Takahashi et al. [12] have found the difference in vocabulary distribution between rumor and nonrumor events. They applied this as a content-based feature in rumor detection. Zhang et al. [13] proposed an automatic rumor detection method based on the combination of traditional shallow features and new proposed implicit features of the message, such as topic popularity, internal and external consistency, sentiment polarity, and match degree of messages.

Similarly, Zhao et al. [14] proposed a rumor detection model based on the decision tree. They tried to find signature text phrases used by a few people to express skepticism about factual claims and are rarely used to express anything else. They used those as features in rumor detection. However, because social networks contain tons of information, extracting content-based features requires excessive time and effort. Moreover, there are unavoidable biases and data dependencies. It is difficult to extract deep-seated underlying features in complex and dynamic social situations.

Generally, the content-based features are the characteristics of the post itself, including timestamp, word count, and URL [15]. It provides a promising feature aspect to construct the rumor detection system.

2.1.2. Propagation-Based Features. The propagation-based features concentrate on the topological structure and credibility propagation [8]. Mendoza et al. [16] explored the behavior of Twitter users and analyzed how rumor propagated through the Twitter network. The results show that the propagation of rumors differs from the truth, and rumors tend to be questioned more than news. Yang et al. [17]

proposed a model incorporating both CNN and RNN for early detection of fake news on social media via classifying news propagation paths. Nir et al. [18] leveraged Weisfeiler–Lehman graph kernels to extract topological information. Bian et al. [19] explored both propagation and dispersion features of rumors with bidirectional graph convolutional networks (Bi-GCNs). Kwon et al. [20] discovered temporal characteristics of rumors on Twitter and demonstrated that rumors likely have fluctuations over time. The researchers fitted the time-series features using random forest.

Besides studying the overall properties and the properties of individual messages, Ma et al. [21] also studied the changes or the trends of these properties along the lifecycle of the rumor information and proposed a time series model to capture the variation in the wide spectrum of social context information, which achieved excellent improvement in rumor detection. Castillo et al. [22] and Yang et al. [23] used decision-tree and support vector machines (SVMs) to model the complete lifecycle of events, respectively. Wu et al. [24] proposed an automatic detection method of rumors on Sina microblog by constructing a graph-kernel based hybrid SVM classifier that captures the high-order propagation patterns in addition to semantic features such as topics and sentiments [25].

The propagation-based models have certain learning abilities, but the features make it difficult to describe the propagation feature of the entire event. It is also hard to structure the complex propagation features, such as the rumor diffusion topology, which makes it impossible to model them directly.

2.1.3. User-Based Features. Castillo et al. [22] exploited registration age, number of users posted messages, number of followers, the scale of moments, and other attributes of users to detect rumors. Some other attributes of users have been used as features for rumor detection in the works by Al-Khalifa et al. [26] and Gupta et al. [27]. Zhang et al. [13] introduced individual features of propagation, such as retweeted opinion influence, and match the degree of messages. The user-based feature is used to model individuals and assess the credibility of every message, which results in the high cost of data collection. Liang et al. [28] define that mostly user-based features contain the following attributes: count of followers, number of followers, personal description, user gender, user avatar type, registration time, and name type.

2.1.4. Other Features. Yang et al. [23] introduced the information from the user client and the location the events took place as features to build a detection model. Sun et al. [29] extracted multimedia features from pictures in messages to identify rumors. Wang et al. [30] introduced the sentimental analysis as an extra feature into time series division and word representations to obtain better performance. Basically, the other features include multimedia and timespan.

2.2. Rumor Detection Based on Deep Learning. As discussed above, the traditional machine learning models for rumor detection are usually based on the manually extracted features or simply use regular expressions to detect misinformation. This strategy requires much expertise, and feature engineering is crucial in this approach. Moreover, the conventional methods mainly concentrate on feature engineering, which fails to cover potential features in new scenarios. It has difficulty in shaping elaborate high-level interactions among significant features.

In order to detect critical features of rumors in social media and retain the time-sequence character of rumor propagation, using an end-to-end DNN is a more practical choice. DNNs, such as convolutional neural network and recurrent neural network, serially receive input sequences and gradually extract features in multilayer training. In recent years, researchers have begun to apply deep learning techniques for rumor detection and achieved remarkable results.

Ma et al. [31], for the first time, applied the end-to-end model to rumor detection. The researchers proposed a recurrent neural network to learn the hidden representations that capture the variation in contextual information of relevant posts over time. The experiments showed that the RNN-based method detects rumors more quickly and accurately than other methods. Similarly, Chen et al. [32] introduced the attention mechanism to RNN. Yu et al. [3] proposed a rumor detection method based on CNN, which extracts key features scattered among an input sequence and shapes high-level interactions among significant features. Their work overcomes the deficiencies that the RNN-based method is not qualified for practical early detection of rumor and poses a bias to the latest input.

The existing end-to-end methods have overcome deficiencies of manual feature extraction and take advantage of semantic and temporal characters of content-based text features. However, the type of features used in these models is limited. The methods focus only on content-based information. The individual characters of each event will not be utilized in previous models, which would lead to failure of rumor detection in specific scenarios when text features are hard to be obtained and processed.

To overcome the defects of the existing rumor detection methods, we attempt to develop an effective end-to-end detection model that is independent of feature engineering and has the ability to aggregate different types of features.

3. Temporal Propagation Pattern Modeling

In this section, we discuss the modeling method of temporal propagation features. Firstly, we analyze the propagation pattern by counting the statistics in each layer of the rumor propagation cycle. Furthermore, we define the propagation pattern feature and the method of implementation. Then, we introduce the nonlinear partition method to solve the long-tail problem, which results in better differentiation of data. Finally, we detail the process of constructing the convolutional neural network and the recurrent neural network,

respectively. Also, we present verification of the validity of temporal propagation feature in rumor detection.

3.1. Propagation Pattern Analysis. The growth of the number of nodes in the propagation graph is an important feature in communication on social networks. In addition, the change in the topological structure of the propagation graph is also a vital feature to describe the process of information dissemination. Research in [16] analyzed the network topology of forwarding behavior in the tweet and pointed out the difference between rumor and nonrumor in propagation pattern.

In traditional machine learning, the sample is described by eigenvector, and the topological structure of the graph is difficult to be used as the learner input. We analyze the growth characteristics of the propagation graph topology of rumor and nonrumor information in social networks and transform it into multiple vectors with a high degree of discrimination. Compared with the nonrumor samples, the rumor samples tend to have more propagation layers and more complex topological structures. Therefore, we first give the quantitative method of the structural growth trend of rumor and nonrumor in the message propagation cycle.

The propagation of the message on a social network can be regarded as a directed acyclic graph (DAG) with a unique root node, and each node can be divided into different layers according to its position: the nodes whose parent node is the root node are in the first layer, their child nodes are in the second layer, and so on. In each time interval, the number of new nodes in each layer of the propagation graph is significantly different. The time series trend of the number of new nodes reflects the growing trend of the propagation to a certain extent.

We describe the message propagation in the social network as a set $E = \{E_i\}$ of event, and any event in the event set is a set $E_i = \{(m_{ij}, p_{ij}, t_{ij}, l_{ij})\}$ of a series of event-related messages (e.g., Weibo and tweets). Each message has a timestamp t_{ij} , indicating its release time and its source p_{ij} ; that is, the message m_{ij} is forwarded from the message p_{ij} . In the propagation topology, p_{ij} is the parent node of m_{ij} and l_{ij} is the layer of node m_{ij} .

Let the release time of the earliest message of event E_i is TimeFirst_i , and the latest message release time is TimeLast_i . The propagating period of event E_i is divided into N equal time intervals. The following formulas describe the linear time interval calculation process of each message m_{ij} :

$$\text{Interval}(E_i) = \left\lfloor \frac{\text{TimeLast}_i - \text{TimeFirst}_i}{N} \right\rfloor, \quad (1)$$

$$\text{TimeStamp}(m_{ij}) = \left\lfloor \frac{t_{ij} - \text{TimeFirst}_i}{\text{Interval}(E_i)} \right\rfloor, \quad (2)$$

where $\text{Interval}(\cdot)$ indicates that the event is divided into N equal length and $\text{TimeStamp}(\cdot)$ indicates the time interval index of the message. Tables 1 and 2 show the statistics of the number of nodes at the end of the propagation cycle of rumor and nonrumor samples, respectively. We adopt the dataset

provided in literature [31] as the experimental data. For Sina Weibo, the dataset collected a series of identified rumors from the Sina Community Management Center. For a specific event, through the application programming interface of Weibo to get the related original messages and all forwarding, comment messages. The dataset used in the following experiments contains 2313 rumor samples and 2351 nonrumor samples.

The data in Tables 1 and 2 show that, in the process of message propagation, most of the nodes are concentrated in the first four layers, and most of the samples are propagated in no more than four layers. Therefore, we focus on the temporal characteristics of the newly added nodes in the first four layers in the information propagation graph.

Based on the above analysis, the temporal topological features of the event E_i are expressed as

$$\text{SeriesTop}(E_i) = (ST_i^1, ST_i^2, ST_i^3, ST_i^4), \quad (3)$$

$$ST_i^L = (st_{i,0}^L, st_{i,1}^L, \dots, st_{i,n-1}^L)^T, \quad (4)$$

$$st_{ij}^L = |\{m_{ik} | \text{TimeStamp}(m_{ik}) = j \& l_{ik} = L\}|, \quad (5)$$

where $\text{SeriesTop}(\cdot)$ represents the time series topology of the event, ST_i^L represents the time series volume in the L th layer, and n is the feature length.

We randomly select four rumor samples and four nonrumor samples and present the temporal variation in the number of new nodes in each layer, as shown in Figure 1.

In Figure 1, the four subgraphs in the first line represent the distribution curves of the newly added nodes of a certain rumor event, and the subgraphs in the second line represent the random nonrumor event. The abscissa in each subgraph represents the time of propagation, and the ordinate represents the number of new nodes for each time period.

As shown in Figure 1, compared with nonrumor events, rumor events exhibit a richer hierarchy, usually at more than layer 3, while the propagating levels of nonrumor events are generally below layer 2. Secondly, it can be found from Figure 1 that the growth trend of nodes in each layer divided by linear equal-time interval appears as obvious long tail phenomenon. In order to solve this problem, a nonlinear partition method of the interval is proposed in this paper; that is, the timestamp of each node is mapped to logarithmic space according to the logarithm of the time interval. Through this measure, the interval between the backward intervals in the propagation cycle becomes longer. After adjustment of formulas (1) and (2), the following formulas are obtained:

$$\text{Interval}(E_i) = \left\lfloor \frac{\log(\text{TimeLast}_i - \text{TimeFirst}_i - 1)}{N} \right\rfloor, \quad (6)$$

$$\text{TimeStamp}(m_{ij}) = \left\lfloor \frac{\log(t_{ij} - \text{TimeFirst}_i + 1)}{\text{Interval}(E_i)} \right\rfloor. \quad (7)$$

We divide the rumor and nonrumor samples in Figure 1 into logarithmic time with a base number of 10. The length

TABLE 1: Statistics of the number of nodes in each layer in the rumor propagation cycle.

Layer	Statistic			
	Mean	Standard deviation	Maximum	Nonzero samples
1	496.26	2123.14	39608	2312
2	228.63	825.19	17537	2245
2	81.41	273.17	4825	2163
4	40.98	176.10	4938	1923
5	22.57	98.34	1997	1589
6	12.87	56.81	1136	1268

TABLE 2: Statistics of the number of nodes in each layer in the nonrumor propagation cycle.

Layer	Statistic			
	Mean	Standard deviation	Maximum	Nonzero samples
1	549.24	1351.79	29560	2351
2	110.94	519.59	19377	2099
3	23.34	91.10	2478	1486
4	8.61	41.89	940	972
5	4.26	26.53	605	613
6	2.92	32.07	1190	419

of vector is chosen to be 100, and the growth curve of temporal volume is shown in Figure 2.

In Figure 2, the abscissa denotes the number of nodes, and the ordinate represents the logarithmic time transformed from propagation time. It can be seen from Figure 2 that the data transformed by logarithmic time have no longer long-tail phenomenon. The variation in each stage is well reflected in the whole propagation cycle. Moreover, by comparing rumor events with nonrumor events, it is found that the temporal volume features of rumor samples are more volatile than that of nonrumor samples, and the layers of nonrumor samples are more homogeneous. The growth curve reflects the distribution of temporal volume of social events, and there is clear discrimination between the growth curves of rumor and nonrumor events. Therefore, we can exploit it as the input for the end-to-end model. The detailed modeling method will be discussed in Section 3.2.

3.2. Model Selection. In this section, we will discuss the modeling methods of temporal propagation features and temporal topological features, which will inspire the following research of the feature aggregation model. There are two basic requirements that need to be met for the selected model:

- (i) The logarithmic temporal feature is used as the input of the model. Thus, the learning model should have strong temporal sensitivity and does not require additional feature engineering.
- (ii) The model should be supervised, and the extracted high-level features can be represented as low-dimensional vectors.

According to the analysis of temporal features in Section 3.1, there is clear discrimination between rumor and nonrumor events on the distribution of growth curves. We implement

CNN and RNN to model the temporal features for the following advantages:

- (i) Compared with the traditional machine learning model, the DNN is more suitable for dealing with the sequence features; for example, RNN is suitable for processing the feature vector sequence. Similarly, CNN is suitable for dealing with the feature matrix. Besides, it is also efficient in terms of representation.
- (ii) DNN has shown significant successes in many areas, especially in semantic feature modeling. In the process of rumor propagation, the above analysis proves that the propagation features are relatively smooth in time-series and have rich semantic form and contour characteristics.
- (iii) As a typical representation learning method, the DNN can transform the original input into intermediate features, such as the feature map of each layer in CNN and the hidden layer vector in RNN. The intermediate feature is the result of supervised input data feature extraction in the training process of the DNN. Compared with the original feature, the dimension of intermediate feature is lower, and it has strong statistical characteristics for the sample label. The intermediate feature satisfies the low-dimensional vector form of the deep feature and is an important part of feature aggregation modeling.

3.2.1. Model Construction Based on CNN. According to the analysis in Section 3.1, each propagation event is transformed into a feature vector, which represents the propagation volume of the event in each time period after logarithmic mapping. The length of the feature vector is the optional hyperparameter of the model.

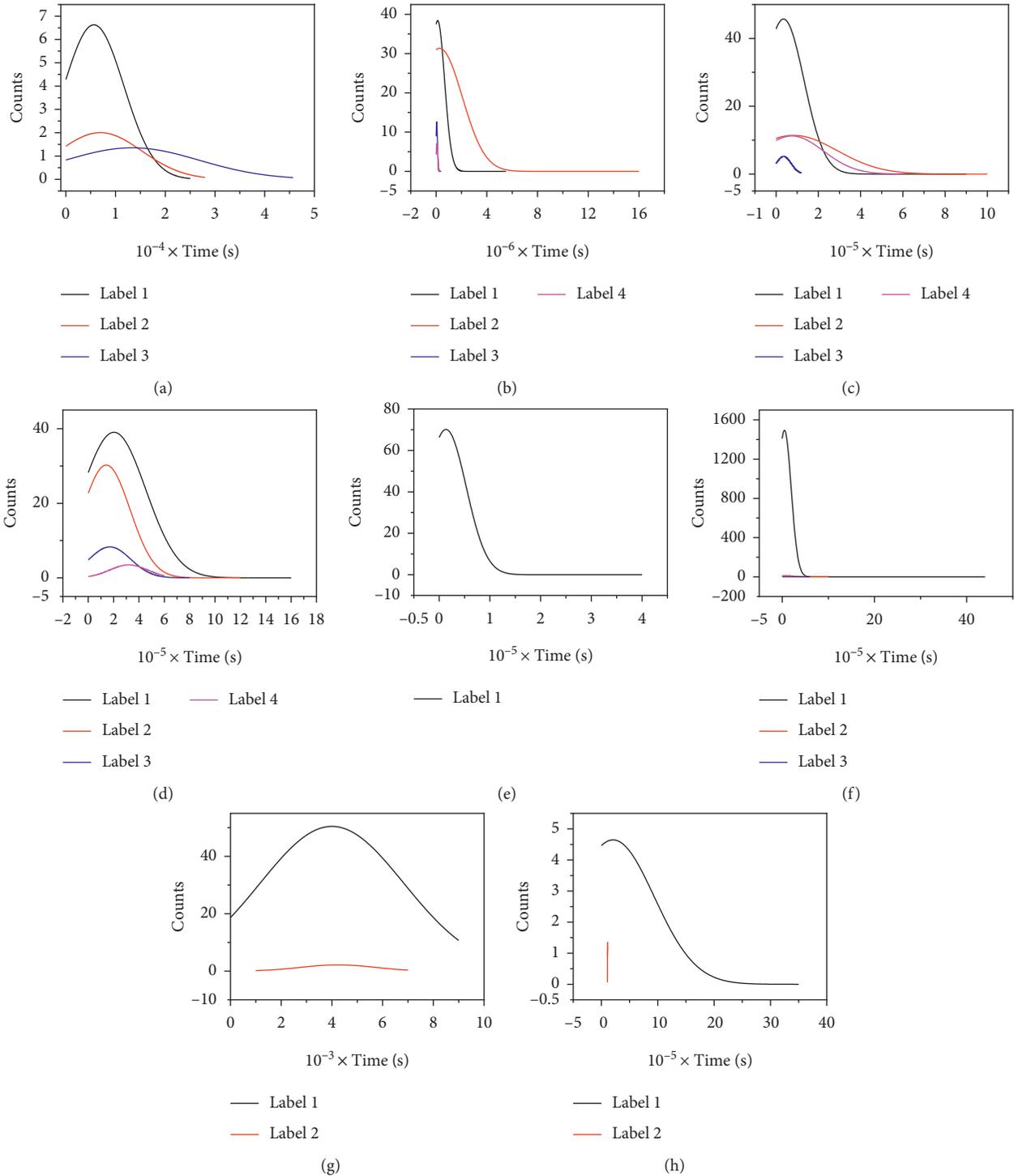


FIGURE 1: Linear time distribution curve of network events.

The feature vector is a sequential combination of the time-series features, and it is a high-dimensional vector that is sensitive to the sequence of features. The feature vector of each sample can be regarded as a specific waveform. The waveform reflects the temporal distribution of the propagation volume of rumor and nonrumor events. CNN has contour sensitivity and is good at dealing with local features, so two-dimensional CNN is used to model these features.

Figure 3 shows the framework of the proposed one-dimensional CNN model, which can be divided into three submodules from the bottom up: (1) data structuring; (2) feature extraction; (3) classification.

(1) *Data Structuring*. The module maps all the relevant messages in each sample to logarithmic time intervals according to the first four topological layers in the

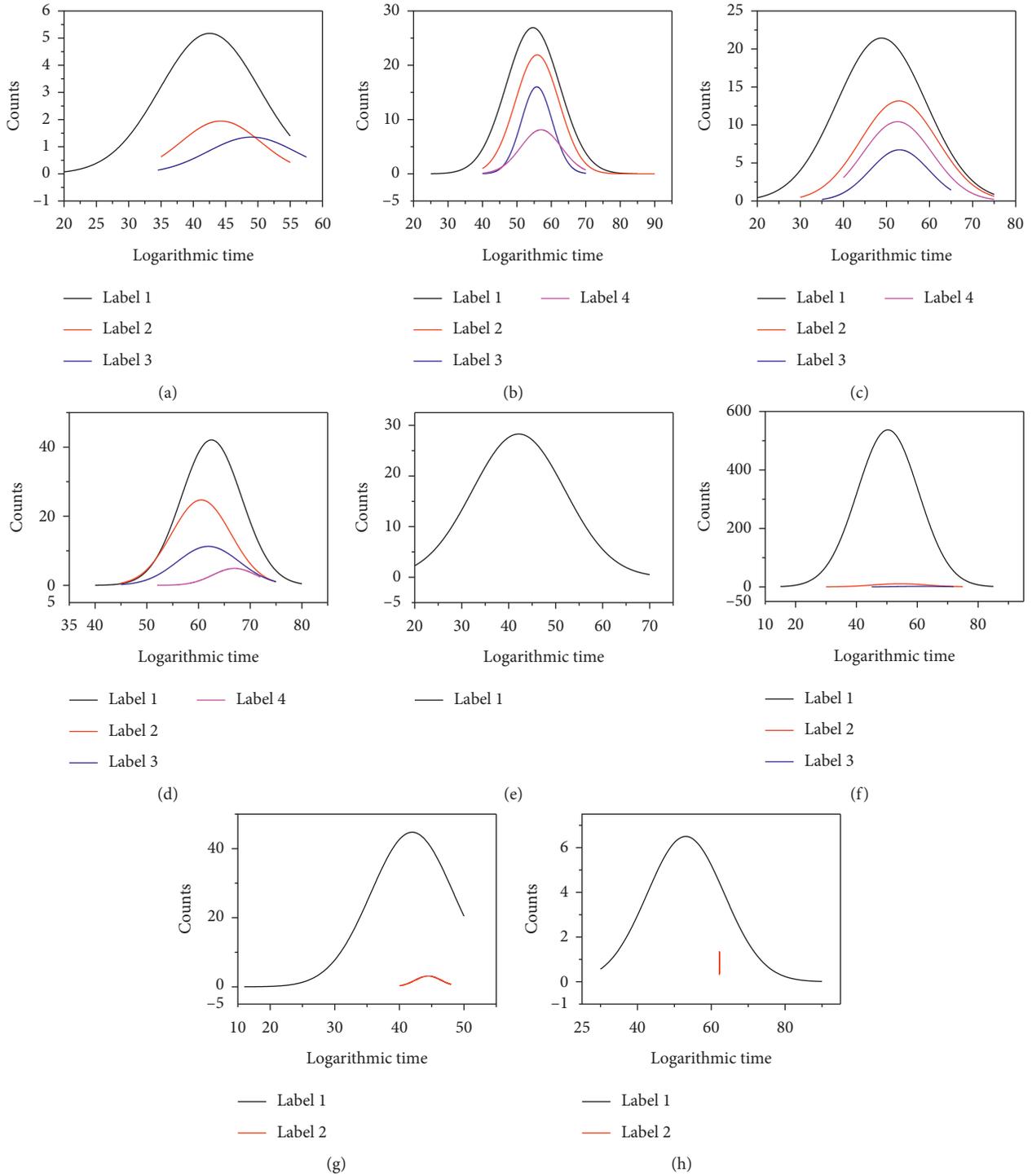


FIGURE 2: Logarithmic time distribution curve of network events.

propagation graph and the released timestamp. The number of intervals is N , and the number of messages in each interval is counted sequentially. Finally, each sample is transformed into a feature matrix composed of four rows.

(2) *Feature Extraction.* The feature extraction module includes two sets of convolutional layer, pooling layer, and activation layer based on ReLU function. The convolutional layers of the model use two-dimensional convolutional

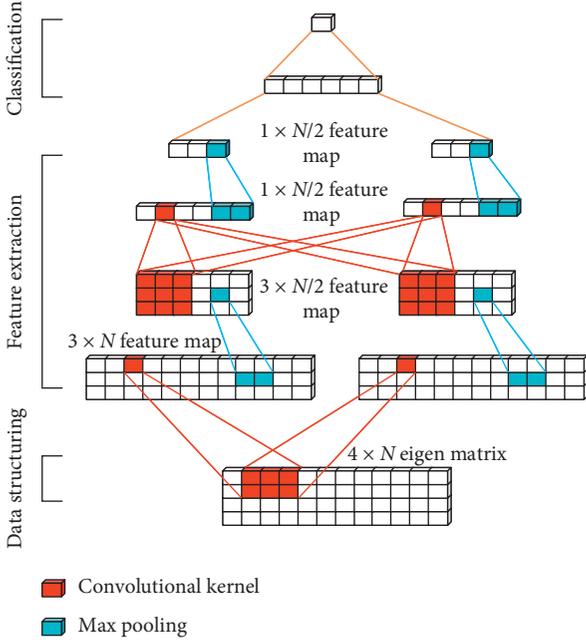


FIGURE 3: Framework of the temporal model based on CNN.

kernels to process the eigenmatrix, and the receptive field of the two groups is different in size. The pooling layer applies a max operation to subsample the output using the maximum value from each of a cluster of neurons at the prior layer.

The first group contains 8 convolutional kernels of 2×3 size. Zero-padding is applied for each row of the feature matrix but not for the column. Therefore, after filtering on the eigenmatrix of size $4 \times N$, 8 feature maps of size $3 \times N$ are obtained. The size of feature maps is converted to $3 \times (N/2)$ after doing the first max-pooling; The second group contains 16 convolutional kernels of 3×3 . We still apply zero-padding for each row of the eigenmatrix. The 16 feature maps of $1 \times (N/2)$ from the first layer are transformed into 16 one-dimensional feature maps with the length of $(N/4)$ after the maxpooling in the second layer. Finally, the model will generate a one-dimensional intermediate eigenvector of length $4N$ by connecting these feature maps.

(3) *Classification*. Because rumor detection is a binary-classification task, there is only one neuron in the output layer of the model. The intermediate feature vectors are connected to the output layer through a fully connected layer. The output value is mapped to the real number between 0 and 1 by using the Sigmoid activation function, and the result represents the classification confidence.

3.2.2. *Model Construction Based on RNN*. The temporal topological characteristics of social events are described as multiple fixed-length vectors, which represent the growth trend of nodes at each layer in the event propagation, and the topological features of the first four layers of the propagation graph are selected for modeling.

The inputs of the RNN model are four feature vectors of length N . In order to make full use of the advantages of

automatic feature extraction and strong sensitivity to time-series structural data, we must overcome the catastrophic forgetting in the network. RNN model tends to forget the earlier feature information in the long input sequences. Long short-term memory (LSTM) RNN and gated recurrent unit (GRU) RNN alleviate this problem. However, for the long sequence scene, the output is more affected by the later features of the sequence. Therefore, if the four feature vectors are connected directly, the innermost and outermost features of the propagation graph would have the greatest influence on prediction.

Because the feature vectors are independent and have complete characteristics of the temporal topology, we propose that the input sequences of the RNN model can be constructed by dividing each vector separately and then using the method of time series splicing.

Figure 4 presents the proposed RNN's framework. Similar to the framework of the CNN model, the RNN model's framework can be divided into three submodules from the bottom up: (1) data structuring; (2) feature extraction; (3) classification.

(1) *Data Structuring*. All the relevant messages in each sample are mapped to logarithmic time intervals according to the first four layers of the topology in the propagation graph and the released timestamp. The interval number is N . The number of messages in each interval is counted sequentially, and four feature vectors of length N are obtained. The topological feature of event E_i in social network propagation is shown in formulas (3) and (4).

The input of the RNN model is a sequence of vectors. For the original feature contains multiple equal-length vectors, it is necessary to overcome the long-term forgetting problem of the model. In this paper, the input sequence of the RNN model is constructed by dividing the vectors separately and splicing them in time series, which is shown as follows:

$$\begin{aligned} \text{Input}(E_i) &= (X_0, X_1, \dots, X_{k-1}), \\ X_t &= x_t^1 \| x_t^2 \| x_t^3 \| x_t^4, \\ x_t^L &= (st_{i,k^*}^L, st_{i,k^*+1}^L, \dots, st_{i,k^*}^L(t+1)-1). \end{aligned} \quad (8)$$

The feature vectors ST_i^L representing the topological structure of each layer in event E_i are, respectively, divided into k segments by equal length. x_t^L represents the k th segment after the vector of the L th layer is segmented. $\text{Input}(\cdot)$ represents the input sequence of the event in RNN model, containing k equal-length vectors, where the vector X_t of order t consists of $x_t^1, x_t^2, x_t^3, x_t^4$.

The time series feature constructed by this method preserves the temporal property of the original features without significantly increasing the length of the sequence.

(2) *Feature Extraction*. In this work, the bidirectional recurrent neural network (BiRNN) is used to learn time-series features. In the process of training, the feature sequences are calculated in the forward direction and the backward direction, respectively. The model processes the feature sequence step by step. The input of each step is the hidden state

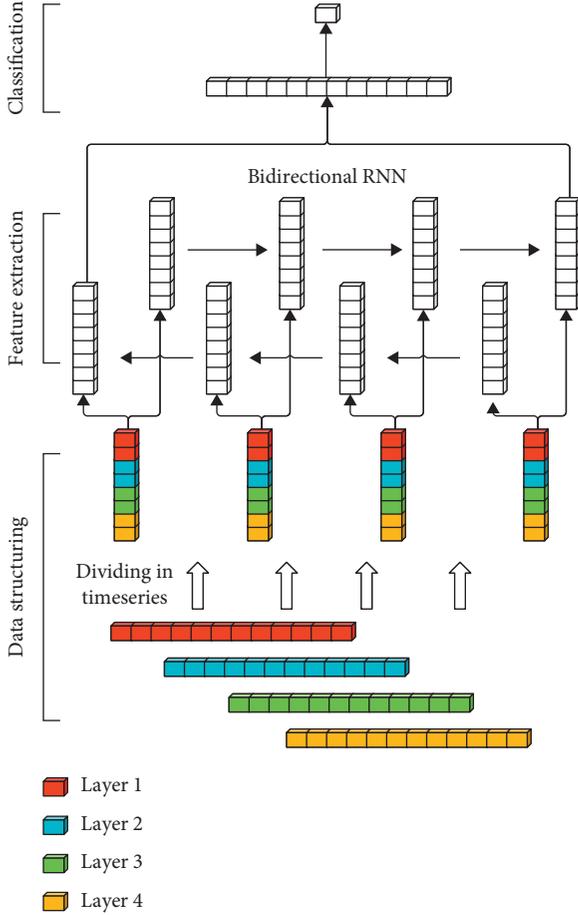


FIGURE 4: Framework of the temporal model based on RNN.

with L length of the output of the previous step and the current feature vector in the sequence. The intermediate feature is represented by the hidden state of the final output of forward and backward passes.

(3) *Classification*. The feature extraction module generates two intermediate vectors with length L , which contain the deep features of the original feature in the forward direction and backward direction, respectively. The two intermediate feature vectors are spliced and fully connected to the output layer. Similar to the CNN model proposed in Section 3.2.1, there is only one neuron in the output layer of the RNN model. The output value uses the Sigmoid function as the activation function, and the output value is mapped to the real number between 0 and 1 to express the classification confidence.

4. Feature Aggregation Model

In this section, we first discuss the framework of the aggregation model. Then, the structure method of text content feature is presented, as well as text feature-based submodels. Finally, we propose how to achieve a higher accuracy in the early detection of rumor than other works.

4.1. *Framework of Aggregation Model*. The type of features used by current end-to-end learning methods is limited, resulting in failing to effectively utilize the rich and easily acquired information outside the text. According to this work, it has been proved that the propagation pattern feature can be effectively used to identify the rumor and nonrumor on the social network. As the information contained in the text content feature and propagation pattern feature is independent, we try to study how to improve the accuracy of rumor detection by aggregating the two different types of features.

Firstly, the submodels of DNNs are constructed for text feature and propagation feature, respectively. Then, the top layers (fully connected layer) of these two submodels are removed. The intermediate feature vectors before the fully connected layer are spliced together and reconnected to a new full connection layer for feature aggregation.

Figure 5 shows an example of the framework of the feature aggregation model. In this example, the text content features are structured and input into the submodel based on RNN (the left submodel in Figure 5), and the propagation pattern features are input into the CNN model (the right submodel in Figure 5). The aggregation model combines the intermediate features generated from the submodels into one feature vector, which will be subclassified by a fully connected layer of a single neuron. Binary cross entropy (BCE) is used as loss function and denoted as L as follows:

$$L = y * \log \hat{y} + (1 - y) * \log(1 - \hat{y}). \quad (9)$$

Denote the neural network i generates the intermediate feature vector $F_i(x; \theta_i)$ from the original feature x . The weight parameter of top fully connected layer is w_i , and the bias parameter is b_i . The weight parameter of top fully connected layer in the feature aggregation model is w , and its corresponding bias parameter is b . For the original feature x , the following formula represents the prediction of the feature aggregation model:

$$\hat{y}' = \text{sigmoid}(W^T \cdot \text{concat}_i^n(F_i(x; \theta_i)) + b). \quad (10)$$

Because the intermediate feature vector is not fixed, the errors exist in the feature extraction, combination, and classification process, which are still involved in the backpropagation of the submodel and provide the gradient for parameter updating. The following formulas calculate the gradient of the submodel parameters during the error backpropagation of the feature aggregation model:

$$\frac{\partial L}{\partial F} = F \cdot \left(y - \text{sigmoid} \left(\sum_i^n \beta_i \cdot F_i \right) \right), \quad (11)$$

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial F} \cdot \frac{\partial F}{\partial \theta} = F \cdot \left(y - \text{sigmoid} \left(\sum_i^n \beta_i \cdot F_i \right) \right) \cdot \frac{\partial F}{\partial \theta}, \quad (12)$$

where $\beta = (w; b)$ and the intermediate feature vector gradients of each submodel are influenced by each other, as

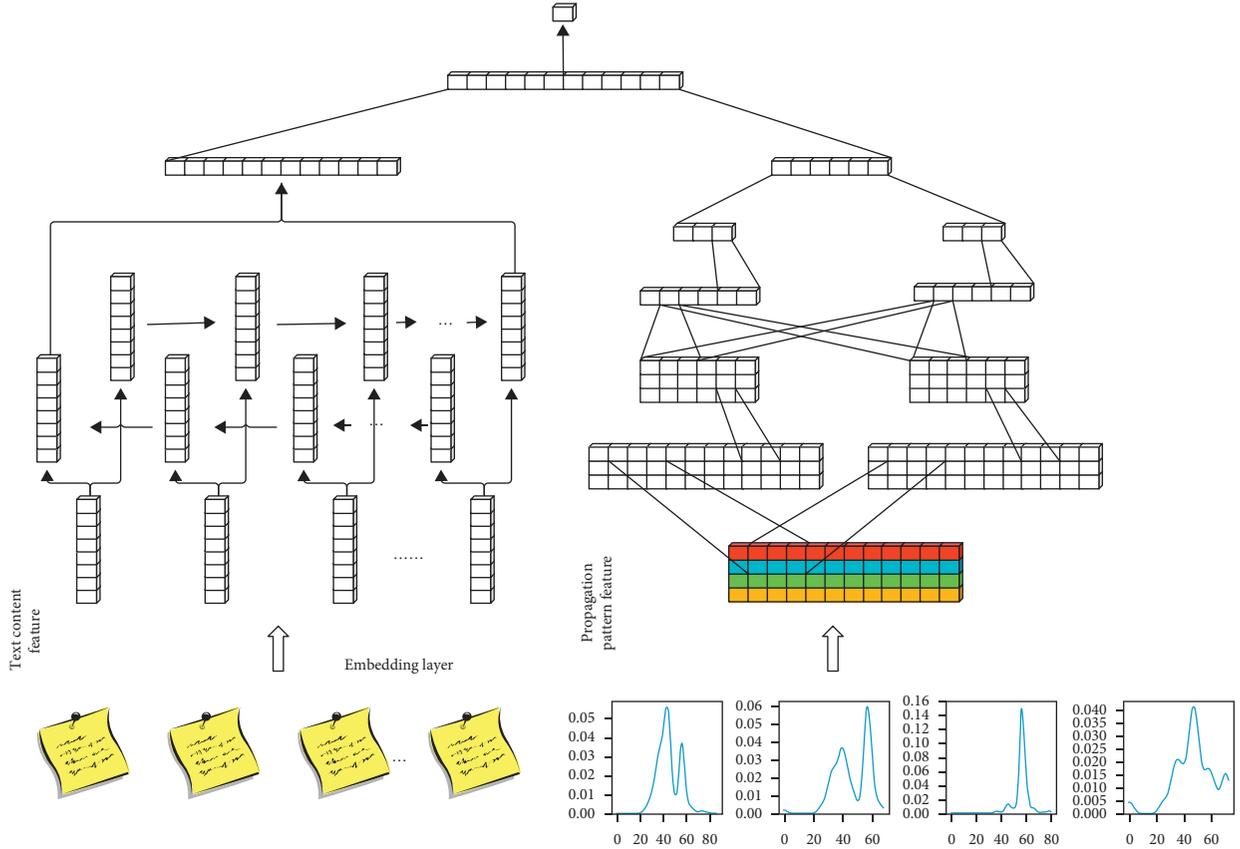


FIGURE 5: Example of feature aggregation model framework.

shown in formulas (11) and (12). Thus, they are effectively complemented in the supervised feature extraction process.

It needs to note that the type of neural networks used to construct submodels can be changed. We will choose the DNN that performs best in the current dataset to build the submodel handling certain features; for example, if CNN is more suitable for dealing with text content feature than RNN in the used dataset, the RNN-based submodel in Figure 5 would be replaced by a CNN-based submodel. The details are discussed in Section 5.3.

The aggregation model combines the models with different structures and constructs a complete neural network to learn heterogeneous features, which makes full use of the knowledge of text feature and propagation feature and the advantages of submodels with different structures.

4.2. Text Content Feature Structure Method. In this section, the structured method for text content feature is discussed. As the input to the RNN submodel of the aggregation model, the quality of text content feature considerably affects the performance of the aggregation model. Nevertheless, the existing rumor detection methods for text content are based on natural language processing, which applies different structures and vectorization methods. The essence of these methods is the low-dimensional embedding of original text information, and these methods focus on different attributes

of text information, resulting in inevitable reconstruction errors and deviations.

Based on word vector and paragraph vector, previous studies have proposed structured approaches of temporal text feature. Chen et al. [32] designed an RNN model to structure text features. They grouped the messages in the event propagation on the social network at equal intervals. The information extracted from each group is used as a unit in the input sequence of RNN. Due to the uneven times of message releasing, the partial groups of the input sequence are empty. Thus, there is no information released at some time intervals. To solve this problem, the model sets a referential input sequence length N . The model attempts to divide the entire propagation cycle into several groups using different lengths of time-separated step to make group number of the longest nonempty continuous packet close to that of N . Only these continuous nonempty groups are taken as input data. Each group of input data is regarded as a document. By calculating the TF-IDF value of each word in the document, the keywords are selected as the input of the sequence unit. Similarly, Yu et al. [3] developed a text-based feature modeling method based on CNN. In this method, the time order of message releasing is used to replace the absolute time, and the messages in the event propagation are divided into 20 groups in sequence. The difference in the number of messages in each group does not exceed 1. The text information of each group is treated as a paragraph, and

the pretrained paragraph vector is used to represent the text information of each group.

However, the method proposed by Chen et al. [32] emphasized on the temporal continuity of text features. A large number of texts are discarded in the process of selecting nonempty continuous time interval due to the inhomogeneity of message releasing time. It fails to maximize the knowledge of full text information. For the method proposed by Yu et al. [3], because the text content is divided into 20 paragraphs in order, the amount of released messages in different samples varies greatly. Therefore, in the process of pretraining, there is an intense difference in input paragraphs' length, which heavily limits the speed and accuracy of paragraph vector training, while the quality of paragraph vector directly affects the prediction ability of the model.

To solve the above problems, we propose a structure method for text content feature based on word vector. The messages in the sample are first padded into N (default of N is 20) groups according to the time order of releasing, and the difference in the number of messages in each group is not more than 1. Each group was regarded as a document. Different from previous works, we calculate the TF-IDF value of the words for each group in the context of all samples. Prune the group by keeping the top- K (default of K is 10) words according to their TF-IDF values. Algorithm 1 details the process.

However, the scale of parameters may be significantly enlarged because of the gated units of GRUs. To reduce the complexity, an embedding layer with a fixed length of 100 is added as the first layer of model [31]. The embedding layer first initializes the embedding vector at random and then uses network optimizer to update it. The average of the embedding words of the top- K keywords is used as the feature vector of the current group.

Because the keywords are extracted in the context of all samples, the vocabulary may contain any words appearing in the text, resulting in a huge scale of the matrix. Furthermore, each text group only uses top- K works with the largest TF-IDF, and there would be many repetitions. Thus, in the actual training process of the model, most of the embedding works do not participate in the calculation and weight updating, and the size of the model parameters will not be greatly affected by the embedding layer used in the structured process.

In the CNN-based model CNN-Text, the text feature vectors of each sample are combined into a 20×100 matrix as input. There are two convolution layers in CNN-Text: the first convolution layer has 8 two-dimensional convolution kernels with a size of 7×100 and translates the input matrix into 8 one-dimensional feature maps with a length of 20 (3 extra rows of zero are padded before the first row and after the last row of the input matrix, respectively); the second convolution layer contains 16 one-dimensional kernels with a length of 3.

We use BiRNN to build the RNN-Text model. The input of the model is a vector stream consisting of the text feature vectors at time-sequential order, and the time step of it is N . The performance of the two text-feature based models is discussed in Section 5.

4.3. Early Detection of Rumor. The rumor detection model needs not only to identify misinformation after the end of event propagation on social networks but also to detect rumors in the early spreading of events. Early detection of rumors can help the government prevent the spread of rumors in time and reduce the adverse influence of rumors on public safety.

Among the existing works, early detection of rumors is based on the same model of rumor detection. The general model is trained with all the samples of complete propagation events, but the researchers measure the performance of rumor early detection by only setting deadlines on test data.

However, because the test data are truncated according to the set deadline, the features of the last part of test data are invisible to the model, resulting in difference in mathematical distribution between training data and test data. The model tends to believe that the propagation is over at the deadline of test data so that the distribution of the data is judged wrongly, and the predicting result is ultimately affected.

We suggest setting the same deadlines on the training data and test data simultaneously to overcome the problem above. By this method, the rest data after deadline of training data and test data are both invisible to the model, which ensures that the mathematical distribution of the two dataset is consistent. A corresponding early detection model is trained for each deadline instead of using all training data. Although there will be more models needed to be trained, it can significantly improve the accuracy of early detection of misinformation.

5. Experiments

In this section, we first present the experimental results of the detection model based on propagation pattern features. Next, we verify the proposed feature aggregation model. The results of early detection of rumors are shown in the end. The experimental dataset consists of 2313 rumor samples and 2351 nonrumor samples, which are based on the public dataset established by Ma et al. [31]. Similar to the study in [3], 10% of all the 4664 samples are randomly chosen for model tuning, and the rest 90% samples are randomly assigned in a 7:3 ratio for training and test. Our source code is accessible at GitHub 2.

5.1. Performance Metrics. For the performance metrics adopted in this work, accuracy, precision, recall, and F_1 values are used in the experiments. Accuracy is the probability that the rumor and nonrumor samples are correctly predicted. Precision is the proportion of correctly classified (non)rumor samples to the total classified (non) rumor samples. Also, the F_1 value is the harmonic average of Precision and Recall. In the formulas (14)–(16), in order to distinguish different types of samples, the rumor sample is represented as R and the nonrumor sample is N . The calculation of each metric is shown in the following equation:

$$\text{Accuracy} = \frac{\text{Correctly classified samples}}{\text{All samples}}, \quad (13)$$

$$\text{Precision}^{(\text{RorN})} = \frac{\text{Correctly classified (non)rumor samples}}{\text{Samples classified as (non)rumors}}, \quad (14)$$

$$\text{Recall}^{(\text{RorN})} = \frac{\text{Correctly classified (non)rumor samples}}{\text{All (non)rumor samples}}, \quad (15)$$

$$F_1^{(\text{RorN})} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (16)$$

5.2. Verification of Propagation Pattern-Based Model. The performance of the propagation-pattern-based model proposed in Section 3 is analyzed in this section. Several methods are used for empirical comparison with ours:

- (1) DT-Rand [14] is a decision tree-based ranking model to identify trending rumors through ranking the clustered disputed factual claims based on statistical features.
- (2) SVM-RBF [23] is a SVM-based model with the RBF kernel.
- (3) DTC [22] is a decision-tree-based classifier to assess information credibility.
- (4) RFC [20] is a random forest-based model with three parameters to fit the temporal tweets volume curve.
- (5) SVM-TS [21] is a linear SVM classifier that uses time series structures to model the variation in context features based on content, users, and propagation patterns.

Table 3 lists the feature domain of the models in the contrast experiment. Compared with the other methods, the rumor detection method proposed in this paper only uses propagation pattern feature to build the classifier. Because this model has fewer feature sources than others and does not rely on feature engineering and domain knowledge, it is easier to obtain sufficient training data.

Table 4 illustrates the experimental results. We adopt accuracy, precision, recall, and F-measure as the evaluation metrics to measure the performance of rumor detection. The CNN-Top is the CNN model based on temporal propagation pattern feature, and the RNN-Top represents the propagation-pattern-based model built by RNN.

It can be seen from the results in Table 4 that although the four contrast models have good rumor identifying ability and the accuracy and F-measure of each model are higher than 0.8, our proposed rumor detection models are superior to these models in all evaluation metrics. Furthermore, the detection performance of the model constructed by CNN is slightly better than that of the model constructed by RNN. Thus, we use the CNN model as the submodel of the aggregation model to handle the temporal propagation feature.

The temporal pattern feature used in the proposed model is essentially a separation of the temporal volume feature in the topological structure of the propagation graph. The results show that although the dimension of the feature increases sharply and the complexity of the model increases after the feature is separated according to the layer, this feature provides more useful knowledge for classification and prediction and can adequately reflect the difference in propagation between misinformation and common information on the social network.

5.3. Verification of Text Content Feature. Text feature handling models CNN-Text and RNN-Text are built based on the structure method proposed in Section 4.2. Similar works are used for empirical comparison with ours:

- (1) GRU-2 [31] is the first end-to-end model that identifies rumors. This RNN-based model learns the hidden representations that capture the variation in contextual information on relevant posts over time. For experiment setting, the vocabulary size K , the embedding size, the size of the hidden units, and the learning rate are empirically set as 5000, 100, 100, and 0.5.
- (2) CAMI [3] is a CNN-based model that extracts critical features scattered among an input sequence and shapes high-level interactions among significant features. The parameters of CAMI are set as $m = [6, 4]$ and $w = [7, 5]$, where m and w represent the numbers of feature maps and filter width. Table 5 presents the experimental results of the text content feature based models with a similar mode structure.

The results in Table 5 show that the RNN-Text model has a higher accuracy than the GRU-2 model which is also based on RNN. Although the GRU-2 model performs better in recall and accuracy of nonrumor detection, the RNN-Text model is more balanced in each aspect. For the model based on CNN, the CNN-Text model is superior to CAMI in almost all evaluation metrics. The experiment proves that our structure method of text content feature can more effectively represent the characteristics of rumor and nonrumor events than comparative researches. On the other hand, although RNN model usually makes more intuitive sense to text input (it resembles how humans process language: reading sequentially from left to right), the CNN seems to be more suitable for handling the text content feature using our structure method. The CNN-Text model achieves better performance in each evaluation metric than the RNN-Text model.

5.4. Verification of Feature Aggregation Model. We verify the effect of the aggregation model based on propagation pattern feature and text content feature in this section. According to the analysis above, we use CNN to construct the submodels CNN-text and CNN-Top to handle text content feature and propagation pattern feature, respectively.

```

Input:  $E_i = (m_{ij}, t_{ij})$  temporal text data related to event  $i$ 
Output:  $S_i = (s_{i0}, s_{i1}, \dots, s_{in})$  structured sequence of temporal text related to event  $i$ 
batched_post  $\leftarrow$  0;
 $E_i \leftarrow$  sort_by_time ( $E_i$ );
for  $t$  in 0,  $\dots$ , 19 do
  batch_size  $\leftarrow$   $\lceil (E_i \cdot \text{length} - \text{batched\_post}) / (20 - t) \rceil$  //Calculate the batch size.
  start  $\leftarrow$  batched_post;
  doc  $\leftarrow$  ( $m_{i,\text{start}}, m_{i,\text{start}+1}, \dots, m_{i,\text{start}+\text{batch\_size}-1}$ );
  batch_ $[t]$   $\leftarrow$  tf_idf (doc, IDF,  $K$ )//Calculate the top- $k$  words in TF-IDF value.
  embedded_batch [ $t$ ]  $\leftarrow$  word_embedding (batch [ $t$ ]); //Obtaining word vector.
  batched_post  $\leftarrow$  batched_post + batch_size;
return embedded_batch

```

ALGORITHM 1: Temporal text feature construction.

TABLE 3: Feature domain of the methods in contrast experiment.

Method	Propagation	User	Content	Others
DT-Rank	✓	✓	✓	
SVM-RBF	✓	✓		✓
DTC	✓	✓	✓	✓
RFC	✓			
SVM-TS	✓	✓	✓	✓
Ours	✓			

One more similar work CallAtRumors [32] using attention mechanism is added for empirical comparison with feature aggregation model. CallAtRumors set the amount of posts N for each time step as 50 and the minimum post series length Min as 5 and $K = 10,000$. Table 6 shows the results of the contrast experiment for the aggregation model. These models are trained and tested in the same dataset.

The two submodels of the aggregation model produce an ideal complementary effect according to the results in Table 6. The proposed aggregation model can effectively identify the rumors at an average accuracy close to 95%, which is better than the other three contrast models. In addition, there are apparent enhancements in F-measure and other evaluation metrics.

5.5. Performance on Early Detection. We set a series of detection check points in the test set and utilize the messages from the initial broadcast to corresponding check point during the test process. It needs to note that, for our proposed model, the training set is also partitioned into several subsets based on the same check point in the test set. We trained 9 separate aggregation models using the data from the initial to the 1st, 2nd, 6th, 12th, 24th, 36th, 48th, 72nd, and 96th hours of event propagation cycle, respectively.

The CNN-Text/CNN-Top model is still selected as the test instance. Table 7 presents the results of early detection contrast experiment at the 9 selected check points, of which 72 hours is the check points for Sina Weibo to conduct manual investigation and judgment on controversial events. In this section, the model is tested via implementing the Rmsprop optimizer [33], and the number of iterations of the training set is more than 5.

TABLE 4: Performance of proposed propagation pattern model (R: rumor; N: nonrumor).

Method	Label	Accuracy	Precision	Recall	F_1
DT-Rank	R	0.732	0.738	0.715	0.726
	N		0.726	0.749	0.737
SVM-RBF	R	0.818	0.822	0.812	0.817
	N		0.815	0.824	0.819
DTC	R	0.831	0.847	0.815	0.831
	N		0.815	0.847	0.830
RFC	R	0.849	0.786	0.959	0.864
	N		0.947	0.739	0.830
SVM-TS	R	0.857	0.839	0.885	0.861
	N		0.878	0.830	0.857
CNN-Top	R	0.874	0.878	0.874	0.876
	N		0.869	0.873	0.871
RNN-Top	R	0.868	0.853	0.897	0.874
	N		0.886	0.838	0.861

TABLE 5: Performance of proposed text content model (R: rumor; N: nonrumor).

Method	Label	Accuracy	Precision	Recall	F_1
GRU-2	R	0.910	0.876	0.956	0.914
	N		0.952	0.864	0.906
CAMI	R	0.933	0.921	0.945	0.933
	N		0.945	0.921	0.932
RNN-Text	R	0.915	0.916	0.918	0.917
	N		0.914	0.912	0.913
CNN-Text	R	0.937	0.923	0.956	0.939
	N		0.952	0.917	0.934

From the experimental results in Table 7, we can see that our method can achieve relatively high accuracy of rumor detection in a short period of time. The performance of our method still maintains a high level in the middle and late stages of the event propagation. At the first hour and the second hour of the event propagation, the accuracy rate exceeds 90% and 92%, respectively. The model is more than 94 percent accurate in detecting early rumors at the 72nd hour.

TABLE 6: Performance of feature aggregation model (R: rumor; N: nonrumor).

Method	Label	Accuracy	Precision	Recall	F_1
GRU-2	R	0.910	0.876	0.956	0.914
	N		0.952	0.864	0.906
CallAtRumors	R	—	0.871	0.863	0.867
	N		—	—	—
CAMI	R	0.933	0.921	0.945	0.933
	N		0.945	0.921	0.932
CNN-Text/CNN-Top	R	0.944	0.935	0.958	0.946
	N		0.955	0.930	0.942

TABLE 7: Early detection of rumor (R: rumor; N: nonrumor).

Deadline (h)	Label	Accuracy	Precision	Recall	F_1
1	R	0.908	0.878	0.952	0.913
	N		0.945	0.862	0.901
2	R	0.928	0.920	0.941	0.931
	N		0.937	0.914	0.926
6	R	0.912	0.930	0.895	0.912
	N		0.895	0.930	0.912
12	R	0.924	0.913	0.941	0.927
	N		0.937	0.906	0.921
24	R	0.941	0.949	0.935	0.942
	N		0.933	0.947	0.940
36	R	0.944	0.936	0.956	0.946
	N		0.953	0.932	0.942
48	R	0.927	0.944	0.912	0.928
	N		0.911	0.943	0.927
72	R	0.942	0.940	0.948	0.944
	N		0.945	0.936	0.941
96	R	0.940	0.924	0.962	0.943
	N		0.959	0.917	0.937

Figure 6 shows the contrast results of early detection performance between our feature aggregation model and some other models. GRU-2 and CAMI are novel and high-performance models based on deep learning. SVM-TS performs best in the previous works based on traditional machine learning. DT-Rank is a specific model designed for the early detection of rumors. Compared with other methods, the feature aggregation model is significantly superior to the GRU-2, SVM-TS, and DT-Rank in early detection. The CAMI method has a high accuracy in the early stage of event propagation, but its detection performance is still slightly lower than the feature aggregation method at all check points in this experiment. The experimental results show the measure that is synchronously setting the same check points on the training data and test data makes the feature aggregation model more effectively applied to early detection of rumors. As for the performance in execution time, we test 932 rumor and nonrumor samples with a total of 17.73 seconds time consumption. The average time consumption for single sample is 0.019 seconds. We believe the time consumption is in compliance with online deployment requirements.

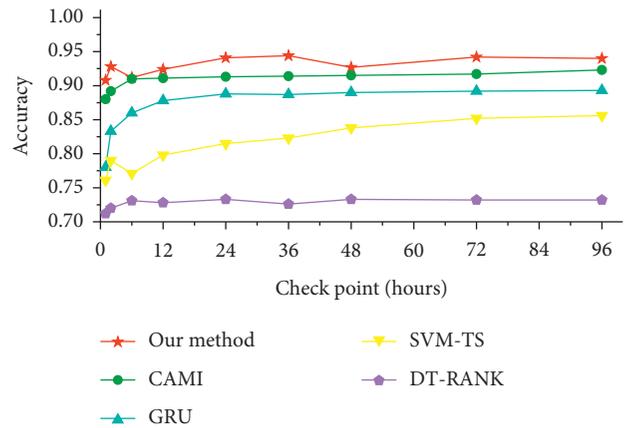


FIGURE 6: Early detection on Weibo.

6. Conclusion

In this paper, an end-to-end model based on feature aggregation is studied to solve the problem of underutilization of heterogeneous features in existing rumor detection methods on social networks. Based on DNN, the text content feature and temporal propagation feature are aggregated effectively. We first propose a propagation pattern feature modeling method, which is independent of feature engineering and domain knowledge and can effectively utilize the temporal information of propagation. By abstracting the volume and the topology of the event propagation cycle, we construct the temporal feature as an acceptable input of DNN. The experiment proves the end-to-end model based on the propagation pattern feature achieves better rumor detection effect than the other models based on traditional manual features. Secondly, we propose a feature aggregation model that efficiently use the rich and independent knowledge of text content feature and propagation pattern feature. The type of features used in relevant works is limited, and the proposed aggregation model overcomes the problem of heterogeneous feature utilization, which enables the learner to cover different types of features simultaneously. Experimental results show that the feature aggregation model has an excellent accuracy rate as high as 94.4% for rumor detection. Moreover, the aggregation model is also effective in the early detection of rumors. The detection accuracy rate at all check points is above 90%, which is much higher than compared works. In our future work, we will research on the combination of other deep-learning models and heterogeneous features to explore the potential for feature aggregation. We believe this work has a substantial practical value and provides theoretical essence for further researches.

Data Availability

The experimental dataset consists of 2313 rumor samples and 2351 nonrumor samples, which are based on the public dataset established by Ma et al. [31]

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Project of National Natural Science Foundation of China (Grant no. 61876134) and the Key Project of National Natural Science Foundation of China (Grant nos. U1836112 and U1536204).

References

- [1] G. W. Allport and L. Postman, *The Psychology of Rumor*, American Psychological Association, Washington, DC, USA, 1947.
- [2] Z. Tan, J. Ning, Y. Liu, X. Wang, G. Yang, and W. Yang, "ECRModel: an elastic collision-based rumor-propagation model in online social networks," *IEEE Access*, vol. 4, pp. 6105–6120, 2016.
- [3] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3901–3907, Melbourne, Australia, August 2017.
- [4] J. Ma, W. Gao, and K. F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *The 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, May 2018.
- [5] N. Ruchansky, S. Seo, and Y. C. Liu, "A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797–806, Singapore, November 2017.
- [6] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rumor source detection with multiple observations: fundamental limits and algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, pp. 1–13, 2014.
- [7] R. G. Guimaraes, R. L. Rosa, D. De Gaetano, D. Z. Rodriguez, and G. Bressan, "Age groups classification in social network using deep learning," *IEEE Access*, vol. 5, pp. 10805–10816, 2017.
- [8] B. Guo, Y. Ding, L. Yao, Y. Liang, and Z. Yu, "The future of false information detection on social media," in *Proceedings of the ACM Computing Surveys (CSUR)*, New York, NY, USA, July 2020.
- [9] J. Ratkiewicz, M. Conover, M. R. Meiss, B. Gonçalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*, pp. 297–304, Barcelona, Spain, July 2011.
- [10] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1589–1599, July 2011.
- [11] M. Vedova, E. Tacchini, S. Moret, G. Ballarin, and L. D. Alfaro, "Automatic online fake news detection combining content and social signals," in *Proceedings of the 22nd Conference of Fruct Association*, Petrozavodsk, Russia, April 2018.
- [12] T. Takahashi and N. Igata, "Rumor detection on Twitter," in *Proceedings of the 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS)*, pp. 452–457, Kobe, Japan, November 2012.
- [13] Q. Zhang, S. Zhang, J. Dong, J. Xiong, and X. Cheng, "Automatic detection of rumor on social network," in *Proceedings of the Natural Language Processing and Chinese Computing*, pp. 113–122, Nanchang, China, October 2015.
- [14] Z. Zhao, P. Resnick, and 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*, pp. 1395–1405, Florence, Italy, May 2015.
- [15] M. Alizadeh, J. N. Shapiro, C. Buntain, and J. A. Tucker, "Content-based features predict social media influence operations," *Science Advances*, vol. 6, Article ID eabb5824, 2020.
- [16] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: can we trust what we RT?" in *Proceedings of the First Workshop on Social Media Analytics*, pp. 71–79, New York, NY, USA, July 2010.
- [17] Y. Liu and Y. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *Proceedings of the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, New Orleans, LO, USA, February 2018.
- [18] N. Rosenfeld, A. Szanto, and D. C. Parkes, "A kernel of truth: determining rumor veracity on twitter by diffusion pattern alone," in *Proceedings of the Web Conference 2020*, Taipei, Taiwan, April 2020.
- [19] T. Bian, X. Xiao, T. Xu et al., "Rumor detection on social media with bi-directional graph convolutional networks," in *Proceedings of the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, New York, NY, USA, February 2020.
- [20] S. Kwon, M. Cha, K. Jung, and W. Chen, "Prominent features of rumor propagation in online social media," in *Proceedings of the International Conference on Data Mining*, Dallas, TX, USA, December 2013.
- [21] J. Ma, W. Gao, Z. Wei, Y. Lu, and K. F. 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*, pp. 1751–1754, Melbourne, Australia, October 2015.
- [22] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proceedings of the 20th International Conference on World Wide Web*, pp. 675–684, Hyderabad, India, March 2011.
- [23] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, p. 13, Beijing, China, August 2012.
- [24] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *Proceedings of the 2015 IEEE 31st International Conference on Data Engineering (ICDE)*, pp. 651–662, Seoul, South Korea, April 2015.
- [25] Z. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Systems with Applications*, vol. 158, Article ID 113595, 2020.
- [26] H. S. Al-Khalifa and R. M. Al-Eidan, "An experimental system for measuring the credibility of news content in Twitter," *International Journal of Web Information Systems*, vol. 7, pp. 130–151, 2011.
- [27] A. Gupta and P. Kumaraguru, "Credibility ranking of tweets during high impact events," in *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, p. 2, Lyon, France, April 2012.
- [28] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, "Rumor identification in microblogging systems based on users' behavior," *IEEE Transactions on Computational Social Systems*, vol. 2, pp. 99–108, 2015.

- [29] S. Sun, H. Liu, J. He, and X. Du, "Detecting event rumors on sina weibo automatically," in *Proceedings of the Web Technologies and Applications*, pp. 120–131, Sydney, Australia, April 2013.
- [30] Z. Wang and Y. Guo, "Rumor events detection enhanced by encoding sentimental information into time series division and word representations," *Neurocomputing*, vol. 397, pp. 224–243, 2020.
- [31] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 3818–3824, New York, NY, USA, July 2016.
- [32] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang, "Call attention to rumors: deep attention based recurrent neural networks for early rumor detectio," *n*, <http://arxiv.org/abs/1704.05973>, 2017.
- [33] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning. Technical Report*, <https://zh.coursera.org/learn/neuralnetworks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude>, 2017.