Computing and Informatics, Vol. 44, 2025, 1229–1253, doi: 10.31577/cai_2025_5_1229

ENHANCING REAL-TIME RUMOR DETECTION ON WEIBO THROUGH USER AND CONTENT FEATURE INTEGRATION

Yu Zhu

School of Finance, Taxation and Public Administration
Jiangxi University of Finance and Economics
Nanchang 330013, China

Gensheng Wang*, Xuejian Huang

School of Information Management
Jiangxi University of Finance and Economics
Nanchang 330013, China
e-mail: wgs74@126.com

Wenwen Jin

School of Marxism
East China Jiaotong University
Nanchang 330013, China

Sheng Li

School of Finance, Taxation and Public Administration
Jiangxi University of Finance and Economics
Nanchang 330013, China

Abstract. Weibo has emerged as a vital platform for Chinese netizens to share information, but it has also given rise to numerous rumors. Real-time detection methods that do not rely on propagation features are the most effective way to curb the spread of these rumors. Currently, real-time detection methods that mine semantic features of rumor text based on deep learning lack sufficient generalization ability. Therefore, we propose a real-time rumor detection method integrating multiple user and content features. In addition to standard user basic features, our approach utilizes the user's historical posting data to extract two deep-level features: user rationality and professionalism. Regarding content features, in addition to standard statistical features, we use a graph attention network that considers edge weights to learn deep semantic features of the content. The user and content features are concatenated and fed into a multi-layer perceptron for classification. The experimental results on a real Weibo dataset show that the accuracy of the proposed method achieves 92.6 %, which outperforms all the compared baseline methods.

Keywords: Rumor real-time detection, semantic features, user features, feature integration, graph attention network, deep learning

Mathematics Subject Classification 2010: 68T50

1 INTRODUCTION

Weibo is China's largest social media platform, similar to Twitter, and it has a vast user base in China. Users can post short texts, images, videos, and more on Weibo, engaging and sharing with other users. As a social media platform, Weibo plays a crucial role in information dissemination and social interaction, greatly appreciated and relied upon by its extensive user base [1]. According to the 53$^{\mathrm{rd}}$ "Statistical Report on Internet Development in China"[1] released by the China Internet Network Information Center (CNNIC) in 2024, as of December 2023, Weibo's monthly active users reached 598 million.

However, Weibo also serves as a breeding ground for the spread of rumors. According to data released by Weibo's official rumor refutation team[2], in 2023, Weibo effectively dealt with 87 191 false information, debunking 1 532 new rumors and controversial events. Rumors on the Weibo platform spread rapidly, have extensive influence, and can cause significant harm [2, 3]. The dissemination of rumors may lead to public misunderstanding, negative emotions, social turmoil, and even impact social stability and national security [4]. For example, during the early stages of the COVID-19 pandemic, many rumors such as "5G networks caused the spread of COVID-19", "COVID-19 vaccine alters human DNA", and "COVID-19 virus is

[1] https://www.cnnic.cn/NMediaFile/2024/0325/MAIN1711355296414FIQ9XKZV63.pdf

[2] https://weibo.com/1866405545/NCeptblE8

man-made" appeared on Weibo, causing public misunderstanding and panic. Although various organizations have established rumor refutation platforms such as the Weibo Community Management Center, China Internet Joint Rumor Refutation Platform, and the fact-checking platform Snopes in the United States, these platforms mainly rely on manual verification, which is labor-intensive and slow. Therefore, researching real-time detection methods for rumors on Weibo has become urgent and an essential issue to address.

Rumor detection typically uses methods based on content, propagation, and hybrid features. Methods based on propagation features require the rumor to spread to a certain extent to achieve satisfactory detection performance, which delays early identification. Methods based on content features do not rely on propagation data, treating rumor detection as a text classification task, thus achieving real-time detection of rumors. However, these methods are easy-to-learn surface features highly correlated with the dataset, leading to poor generalization ability.

Hybrid feature-based methods integrate multiple features, such as combining content and user features with propagation features, reducing reliance on propagation features, and improving early detection performance. Integrating user and content features improve the generalization ability of real-time rumor detection methods. The key to improving the performance of real-time detection methods is to mine additional features to assist content features. Therefore, we propose a real-time rumor detection method integrating multiple user and content features. In addition to standard user basic features and content statistical features, we utilize users' historical posting data to mine two deep-level features: user rationality and professionalism. Aiming at the discrete and fragmented characteristics of Weibo text, we use a graph attention network considering edge weights to learn deep semantic features of the content. Our main contributions are as follows:

1. We propose a real-time rumor detection method integrating multiple content and user features. This method does not rely on propagation features and can identify rumors early, reducing the harm of their spread.
2. We propose two novel user features of rationality and professionalism and design the calculation methods for these features.
3. We propose a method that utilizes a graph attention network considering edge weights to learn deep semantic features of the text.
4. We contributed a comprehensive Weibo rumor dataset[3], and experimental results on this dataset show that our method outperforms all compared baseline methods.

[3] `https://pan.baidu.com/s/1NGYHWhgZWG3eykLVN9XaSA?pwd=ot3c`

2 RELATED WORK

2.1 Rumor Detection Based on Content Features

The rumor detection methods based on content features do not rely on propagation information and can achieve real-time detection of rumors. The current mainstream approach is to extract semantic features of rumor content using deep learning models. For example, Kaliyar et al. [5] utilized Convolutional Neural Networks (CNNs) to extract semantic features of rumors from text. Ajao et al. [6] proposed a model combining CNNs and Long-Short Term Memory (LSTM) networks to learn semantic features of false rumors on Twitter. Cheng et al. [7] introduced a rumor detection method based on Generative Adversarial Networks (GANs), which strengthens the learning of semantic features of rumors through the mutual reinforcement of the discriminator and generator. Some researchers have explored other auxiliary features based on semantic features to improve detection accuracy. For instance, Xu et al. [8] proposed a topic-driven rumor detection framework (TDRD), which utilizes CNNs to extract the topic information of the content and combines it with textual word embeddings for rumor detection. Ma et al. [9] presented a rumor detection approach based on entity recognition to enhance the semantic understanding of rumor texts. Entity explanations are obtained through knowledge graphs, thereby expanding the content of the original text and enhancing semantic understanding. Studies [10, 11] have shown that due to the limited number of samples in the rumor dataset and lack of sufficient contextual information in short texts [12], deep learning-based rumor detection methods that automatically learn content features tend to learn surface features that are highly correlated with the dataset, leading to poor generalization ability of the method.

2.2 Rumor Detection Based on Propagation Features

The rumor detection methods based on propagation features primarily utilize information such as reposts, comments, and propagation structure during the spread of rumors for detection [13, 14]. For example, Ma et al. [15] used Propagation Tree Kernel (PTK) to capture the high-order patterns of rumor propagation structure. Ma et al. [16] constructed a rumor detection method based on Recurrent Neural Networks (RNNs) to learn temporal comment data. References [17, 18] utilized Tree-structured Recursive Neural Networks (RvNNs) to simultaneously learn the semantic features of comment data and the structural features of rumor propagation. To enhance the model's focus on parts of propagation data with rumor characteristics, Chen et al. [19] proposed a rumor detection approach that combines RNN with an attention mechanism. Song et al. [20] recommend a Credible Early Detection (CED) model to address the issue that current methods often learn the semantic representation of all repost data, making it challenging to detect rumors early. The CED model searches for an early time point in the repost sequence to make a credible prediction. Huang et al. [21] introduced a spatiotemporal rumor detection method that incorporates temporal information and propagation structure to address the issue that current rumor detection approaches based on propagation structure overlook temporal features. With the advancements in Graph Neural Networks (GNNs), some researchers leverage GNNs to learn representations of propagation structure and achieve rumor detection [22, 23, 24, 25, 26]. However, studies [27, 28] have found that detection approaches based on propagation features require the rumor to spread to a certain extent to obtain acceptable performance and cannot identify rumors in real-time.

2.3 Rumor Detection Based on Hybrid Features

The rumor detection approaches based on hybrid features integrate multiple features [29, 30, 31]. For instance, Castillo et al. [32] conducted an initial analysis of rumors on Twitter, proposing a rumor classification approach based on propagation, content, topic, and user features.

Regarding most rumor detection models that only use static statistical features of text content, propagation patterns, and user information, ignoring the temporal changes of these features, Ma et al. [33] designed a Dynamic Series-Time Structure (DSTS) to capture the variation of features over time.

Ruchansky et al. [34] combined the textual information of news, user feedback, and source author information for fake news detection, designing three modules: the Capture module extracts original text and user feedback information using RNN, the Score module evaluates user credibility based on their historical data, and the Integrate module combines the outputs of the previous two modules for fake news identification.

Tu et al. [35] introduced a rumor detection framework called Rumor2vec, which jointly learns propagation structure and text representations to reduce the model's reliance on propagation features and improve early detection performance. Experimental results demonstrate that this approach can detect rumors at least 12 hours earlier.

Lotfi et al. [36] introduced a rumor detection approach that integrates propagation and user information and uses Graph Convolutional Networks (GCNs) to learn features from the user interaction and rumor propagation graphs.

Sun et al. [37] proposed the Dual-Dynamic Graph Convolutional Network (DDGCN), which can capture both the dynamics of rumor propagation and background knowledge from knowledge graphs.

To improve real-time rumor detection accuracy, Singh et al. [38] utilized Attention-based LSTM to extract content features, combined with user features to identify rumors. Kaliyar et al. [39] combined news content, user relationship, and user-news correlation to construct a multi-dimensional tensor matrix, decomposed the tensor matrix to get the fusion features of users and content, and detect rumors based on the fusion features.

Through the analysis of the above studies, it is found that integrating additional features beyond content features is essential for improving the performance of real-time rumor detection. Therefore, this paper proposes a real-time rumor detection method that combines multiple user and content features. Based on the user's historical posting data, we introduce two deep-level features: user rationality and professionalism. Furthermore, we employ a graph attention network that considers edge weights to learn deep semantic features of the content.

3 METHODOLOGY

The overall architecture of the real-time Weibo rumor detection method, called RTD-UCF, proposed in this paper based on user and content features is shown in Figure 1. In terms of user features, we not only use standard basic user features but also mine two deep-level features: user rationality and professionalism, from users' historical posting data. For content features, in addition to standard statistical features, we use a graph attention network that considers edge weights to learn deep semantic features of the content. Finally, we concatenate the user and content features and input them into a multi-layer perceptron for classification.

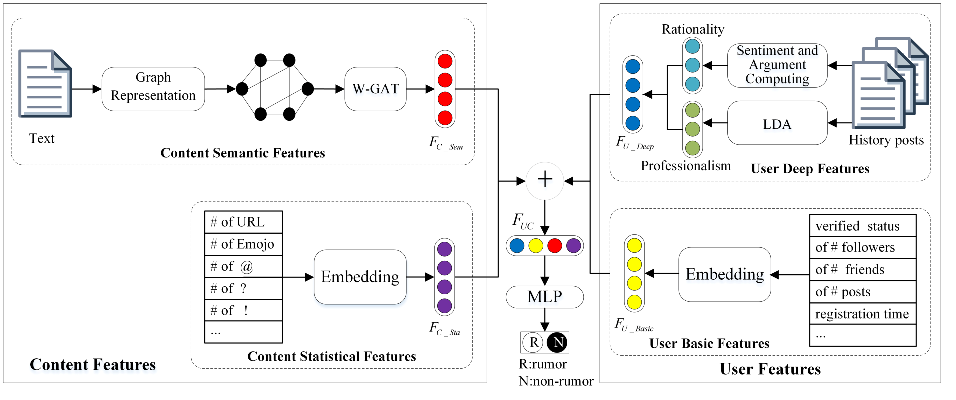

Figure 1. Overall framework of the RTD-UCF

3.1 User Features

3.1.1 User Basic Features

Liu et al. [40] analyzed the basic features of rumor and non-rumor users. They identified five differences: rumor users tend not to use a real person as avatars, users with topic-type usernames are more credible, females are more likely to spread rumors than males, differences in users' locations affect the ability to identify rumors, and non-verified users are more likely to spread rumors than verified users. Morris et al. [41] found that the features of a user's social network can be used to identify rumor users, where users with significantly more friends than followers are more susceptible to manipulation by malicious users and may contribute to the proliferation of misinformation on the network. Castillo et al. [32] discovered that the longer users have been registered, and the more information they have posted, the stronger their ability to identify rumors and the lower their probability of spreading rumors. Therefore, based on these research results, we select basic user features, as shown in Table 1. We use embedding to convert these features into continuous vector representations and concatenate them to obtain the vector representation F_{U_Basic} of basic user features.

$$F_{U\_Basic} = Emb(FU_1) \oplus Emb(FU_2) \cdots \oplus Emb(FU_{10}). \quad (1)$$

Features	Describe
FU_1	Whether the avatar is a real person
FU_2	Whether the username is a topic type
FU_3	Gender
FU_4	Location
FU_5	Whether to verify
FU_6	Number of followers
FU_7	Number of friends
FU_8	Registration duration
FU_9	Number of posts published
FU_{10}	Level

Table 1. User basic features

3.1.2 User Deep Features

User Rationality

Rumors end with the wise. The wise are rational people with independent thinking and judgment. They do not readily believe and spread rumors. Through analysis, we have found that rational users generally do not include strong personal emotions in their posts. Their writing style could be more objective and fair, which is less likely to attract widespread attention. The attitudes of these users' comments are also relatively neutral. However, non-rational users tend to include personal emotions in their posts, which are subjective and provocative in writing style, making them more likely to attract attention. The comments from users are also more controversial, with a clear tendency towards emotional expression. Therefore, we propose to analyze the sentiment of users' historical posts and comments and calculate their rationality value. The calculation method is shown in formula (2).

$$Rat_u = \frac{1}{n} \sum_{i=1}^{n} \left(Sent_{p_i} + Contr_{p_i} \right). \quad (2)$$

Here, Rat_u denotes the rationality value of user u, n denotes the number of historical posts published by user u, and $Sent_{p_i}$ and $Contr_{p_i}$ denote the sentiment score and controversy score of post p_i, computed using formulas (3) and (4), respectively.

$$Sent_{p_i} = \sum_{w_s, w_a \in p_i} \left(\left| SentDegree_{w_s} * AdvDegree_{w_a} \right| \right), \tag{3}$$

$$Contr_{p_i} = \log(neg + pos) * \frac{pos * neg}{pos^2 + neg^2}. \tag{4}$$

In formula (3), w_s and w_a represent the words in the post p_i, $SentDegree_{w_s}$ represents the sentiment value of w_s, and $AdvDegree_{w_a}$ represents the degree value of the adverb w_a modifying w_s. The sentiment score of the post is calculated without distinguishing the polarity of the sentiment by taking the absolute values and adding them together. In formula (4), Neg and Pos denote the number of positive and negative sentiment comments, respectively, ignoring neutral sentiment comments. The higher the number of emotional comments and more opposing the emotional polarity, the higher the controversy score of the post p_i.

User Professionalism

The user frequently posting on a specific topic indicates that they have some knowledge about the topic, and posts related to that topic have a certain level of credibility. We use Latent Dirichlet Allocation (LDA) to discover the latent topics of posts and propose a user professionalism calculation based on topic similarity, as shown in formula (5).

$$Pro_{u,p} = \sum_{i=1}^{n} \cos\left(\overrightarrow{\theta_{p_i}}, \overrightarrow{\theta_p} \right), \tag{5}$$

where $Pro_{u,p}$ represents the professionalism of user u towards the detection post p, n represents the number of historical posts published by user u, $\overrightarrow{\theta_{p_i}} = (p_1, p_2, \ldots, p_k)$ and $\overrightarrow{\theta_p} = (p_1, p_2, \ldots, p_k)$ represent the topics probability distribution of historical post p_i and detection post p, respectively, and k represents the number of topics. The higher the cosine value $\cos\left(\overrightarrow{\theta_{p_i}}, \overrightarrow{\theta_p} \right)$ between the probability distribution of topics of the historical posts p_i published by the user and the post p being detected, the higher the user's professionalism towards the post p. After obtaining the user's rationality and professionalism, we concatenate them and input them into a fully connected layer to obtain the user's deep feature F_{U_Deep}:

$$F_{U\_Deep} = f(W(Rat_u \oplus Pro_{u,p})). \tag{6}$$

Here, the symbol $\oplus$ represents vector concatenation, W and b represent the parameter matrix and bias term of the fully connected layer, and f represents the activation function.

3.2 Content Features

3.2.1 Content Statistical Features

Although symbols, emojis, URLs, and other types of information in the text are often disregarded when learning semantic features, they can have some auxiliary effects on rumor identification. For example, some rumors may deliberately use strong emojis to attract attention. Therefore, we also incorporate content statistics as additional features, as shown in Table 2. Similarly, we utilize embedding to obtain the vector representation F_{C_Sta} of the content statistical features.

$$F_{C\_Sta} = Emb(FC_1) \oplus Emb(FC_2) \oplus \cdots \oplus Emb(FC_7). \tag{7}$$

Features	Describe
FC_1	Is there a topic marked with #
FC_2	Number of URLs
FC_3	Number of emoji
FC_4	Number of @ symbols
FC_5	Number of ? symbols
FC_6	Number of ! symbols
FC_7	Is there a picture or video attached

Table 2. Content statistical features

3.2.2 Content Semantic Features

For post texts with discrete and fragmented characteristics, we utilize a Weighted Graph Attention Network (W-GAT) [42, 43] considering edge weights to learn content semantic features. To begin with, we represent the post's text as a graph $G = (V, E)$, where nodes V represent words, edges E represent the correlations between words, and edge weights represent the degree of correlation between words. We utilize pointwise mutual information (PMI) to measure word correlation, computed using a fixed window to collect co-occurrence statistics of words across all posts. The PMI calculation for word pairs is presented in formulas (8), (9):

$$p(w_i) = \frac{|W(w_i)|}{|W|}, \tag{8}$$

$$p(w_i, w_j) = \frac{|W(w_i, w_j)|}{|W|}, \tag{9}$$

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \tag{10}$$

Here, $|W|$, $|W(w_i)|$, and $|W(w_i, w_j)|$ represent the total number of sliding windows, the number of sliding windows including word w_i, and the number of sliding windows including both w_i and w_j, respectively. We utilize statistics data from a global corpus rather than a specific post. The PMI value reflects the correlation between words, with a positive PMI value indicating a high semantic correlation. Thus, we only retain edges with positive PMI values and discard those with non-positive PMI values, as presented in formula (11).

$$A_{i,j} = \begin{cases} PMI(w_i, w_j), & PMI(w_i, w_j) > 0, \\ 0, & PMI(w_i, w_j) \geq 0. \end{cases} \tag{11}$$

After obtaining the adjacency weight matrix A, we use W-GAT to learn semantic features. The construction process of W-GAT is shown in Figure 2. Where v_i represents node i, h_i^l represents the feature representation of node i in the l^{th} layer of the network, $a_{i,j}^l$ represents the attention weight of neighboring node j to node i in the l^{th} layer of the network. F represents the feature representation of the entire graph obtained after aggregating all node feature representations.

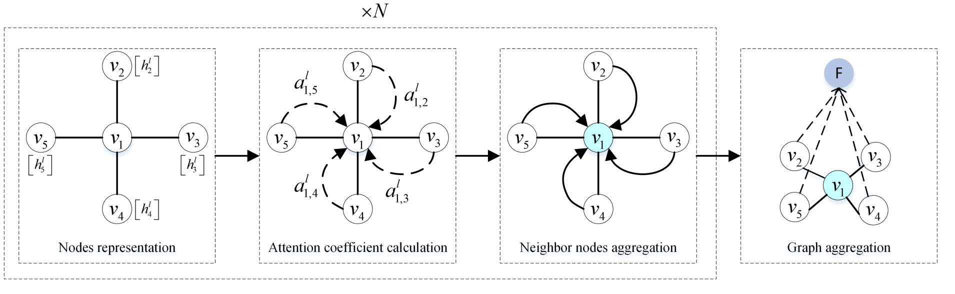

Figure 2. The construction process of W-GAT

We first represent each node as a vector and use it as input to the network. We then calculate the attention coefficient of each neighbor node to the center node. Next, we aggregate the representations of neighboring nodes, update the center node's representation, and use it as input to the next network layer. Finally, we aggregate the representations of all nodes in the last layer to obtain the representation of the entire graph. While the construction process of W-GAT is similar to GAT, W-GAT considers edge weights as an additional input factor when calculating the attention coefficients, which provides more accurate global information. The calculation of W-GAT is shown in formula (12).

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} a_{i,j}^l W^l h_j^l\right). \tag{12}$$

Here σ denotes the activation function, $N(i)$ denotes the set of neighboring nodes of node v_i, W^l denotes the learning parameters of the W-GAT at the l^{th} layer, h_j^l

denotes the output of neighboring nodes v_j in the previous layer, $a_{i,j}^l$ denotes the weight of neighboring nodes v_j to node v_i in the l^{th} layer, which is computed as shown in formula (13). We use Word2vec as the initial feature representation h_i^0 for nodes v_i.

$$a_{i,j}^l = \frac{\exp\left(e_{i,j}^l\right)}{\sum_{j \in N(i)} \exp\left(e_{i,j}^l\right)}, \tag{13}$$

$$e_{i,j}^l = \alpha\left(h_i^l, h_j^l, A_{i,j}\right), \tag{14}$$

where $e_{i,j}^l$ denotes the attention coefficient between node v_i and node v_j in the l^{th} layer of W-GAT, which is computed using formula (14). α is a learnable function, and $A_{i,j}$ is the weight of the edge connecting nodes v_i and v_j, which is the PMI value between words w_i and w_j. After L layers of W-GAT learning, we use global average pooling to aggregate the feature $h_i^{(l+1)}$ of each node v_i in the graph to obtain the semantic features F_{C_Sem}, as calculated in the formula (15). Here, V is the set of nodes, and $|V|$ is the size of V.

$$F_{C\_Sem} = \frac{1}{|V|} \sum_{i \in V} h_i^{(l+1)}. \tag{15}$$

3.3 Classification

After obtaining the user basic feature representation F_{U_Basic}, user deep feature representation F_{U_Deep}, content statistical feature representation F_{C_Sta}, and content semantic feature representation F_{C_Sem}, we concatenate them to get the final fused feature F_{UC} using formula (16). Subsequently, the fused feature F_{UC} as input to a Multi-Layer Perceptron (MLP) and a softmax layer to generate the outputs, as presented in formula (17).

$$F_{UC} = F_{U\_Basic} \oplus F_{U\_Deep} \oplus F_{C\_Sta} \oplus F_{C\_Sem}, \tag{16}$$

$$\hat{y} = \mathit{softmax}(W * MLP(F_{UC}) + b). \tag{17}$$

Here W and b represent linear layer parameters and bias terms, respectively. We train the model by minimizing the cross-entropy loss, as displayed in the formula (18).

$$L = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\right) + \frac{\lambda}{2} \|W\|_2^2, \tag{18}$$

where y_i denotes the real label value of sample p_i, $\hat{y}_i$ denotes the model's predicted value, $\frac{\lambda}{2} \|W\|_2^2$ is $L2$ regularization to reduce the degree of overfitting, and W is the model parameters.

4 EXPERIMENTS

4.1 Experimental Dataset and Evaluation Metrics

4.1.1 Experimental Dataset

The current public rumor datasets do not contain the historical data of the posting users, so the data used in this experiment is collected from the Weibo Community Management Center and the Weibo site using a web crawler. The Weibo Community Management Center has published nearly 50 000 rumors since 2012.

1) **Rumor Samples**

   Over time, users' attributes may undergo significant changes, so for this experiment, we only collect the rumor samples published by the Weibo Community Management Center in the last two years. First, we crawl the rumor posts with text of at least 30 characters. If the text is too short, it lacks semantic information. Then, we collect the basic information of the rumor publisher and remove the corresponding rumor sample if the publisher has deactivated their account. Finally, we crawl the publisher's 200 most recent historical posts and corresponding comment data. We only keep the latest one if multiple rumors belong to the same user. Finally, we obtained 3 756 rumor samples.

2) **Non-rumor Samples**

   Statistical analysis indicates that 88.9 % of Weibo rumors are reported within a week. Therefore, for this experiment, the non-rumor samples are obtained by crawling popular Weibo posts (with total comments, reposts, and likes exceeding 100) that have been posted for more than a week and have not been reported as rumors. The collection of user information and historical data of non-rumor samples is consistent with rumor samples. The statistical information of the experimental dataset is presented in Table 3. We divide the dataset according to the ratio of 3:1:1 to obtain the training, verification, and test sets.

To ensure compliance with the terms and conditions of the data source institution, we have anonymized the collected data. Anonymization involves removing personal identifying information and obfuscating sensitive data.

Statistic	Amount
Number of users	7 512
Number of rumors	3 756
Number of non-rumors	3 756
Number of historical posts	1 209 432
Number of comments on all historical posts	13 454 931

Table 3. Statistics of the datasets

4.1.2 Evaluation Metrics

We utilize accuracy, rumor precision, rumor recall, and F1-score as performance evaluation metrics, calculated according to formulas (19), (20), (21), and (22), respectively.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

$$precision = \frac{TP}{TP + FP}, \tag{20}$$

$$recall = \frac{TP}{TP + FN}, \tag{21}$$

$$F1\text{-}score = \frac{2 * precision * recall}{precision + recall}. \tag{22}$$

Here, TP, TN, FP, and FN represent the number of true positives, the number of true negatives, the number of false positives, and the number of false negatives, respectively.

4.2 Implementation Details

To calculate the user's level of rationality, we first compute the sentiment score of their historical posts using the BosonNLP sentiment dictionary and HowNet's degree adverb list, as shown in formula (3). BosonNLP sentiment lexicon is constructed based on corpora from sources such as Weibo, news, and forums, offering extensive coverage for non-standardized text like Weibo posts. Subsequently, we use the SnowNLP sentiment analysis tool to determine the sentiment polarity of comments on the user's past posts, classifying those with a sentiment value greater than 0.6 as positive comments and those with a value less than 0.3 as negative comments. SnowNLP is a Python library designed to facilitate sentiment analysis of Chinese text. We then apply formula (4) to calculate the controversy of the user's historical posts. Ultimately, we determine the user's rationality value utilizing the formula (2). In computing the user's professionalism value, we establish the topic number K of LDA to be 50, with the document-topic distribution parameter α and topic-word distribution parameter η set to 0.002. The output dimension of the fully connected network, which outputs the user's deep features, is configured to be 8. To acquire content semantic features, we set the sliding window size for PMI calculation to 6. The W-GAT consists of two layers, with the first and second layers having output dimensions of 128 and 64, respectively. We use the pre-trained word embedding library Chinese-World-Vectors to extract word vector representations. The vector dimension of the word embedding library is 300. It contains word vectors specifically designed for Weibo, rendering it more appropriate for Weibo rumor identification tasks than other general-purpose word embedding libraries. The embedding dimensions of user basic and content statistical features are fixed at 4. In the classification network, we utilize a 2-layer perceptron, with output dimensions of 64 and 32 for the first and second layers, respectively. During model training, we use the Adam optimizer with a learning rate of 0.001 and a batch size of 32. The primary hyperparameters for the model and training are presented in Table 4.

Type	Name	Value
Model parameters	# of LDA Topics K	50
	Document-topic distribution parameter α	0.02
	Topic-word distribution parameter η	0.02
	Output dimension of the fully connected layer	8
	# of layers of the W-GAT	2
	Output dimension of the W-GAT	128, 64
	Sliding window size	6
	Word2vec dimension	300
	# of layers MLP	2
	Output dimension of MLP	64, 32
	Embedding dimensions	4
Training parameters	Learning rate	1e−3
	Regularization parameter	1e−5
	Maximum training epochs	30
	Early stopping patience	8
	Batch size	32
	Dropout rate	0.1
	Optimizer	Adam

Table 4. Hyperparameters setting

4.3 Results and Analysis

4.3.1 Hyperparameter Experimental Analysis

In the calculation process of pointwise mutual information, the size of the sliding window is a critical hyperparameter. We employ grid search to determine the optimal sliding window size, with a search range of 3, 6, 9, 12, 15, 18, 21. The experimental results of the RTD-UCF method under different sliding window sizes are depicted in Figure 3. From the figure, it can be observed that selecting a sliding window size that is too small or too large can adversely affect the performance of the RTD-UCF method. When the sliding window is too small, insufficient context information exists to capture the correlation between two words. Conversely, when the sliding window is too large, it may contain irrelevant information, potentially overshadowing the crucial information between the two words.

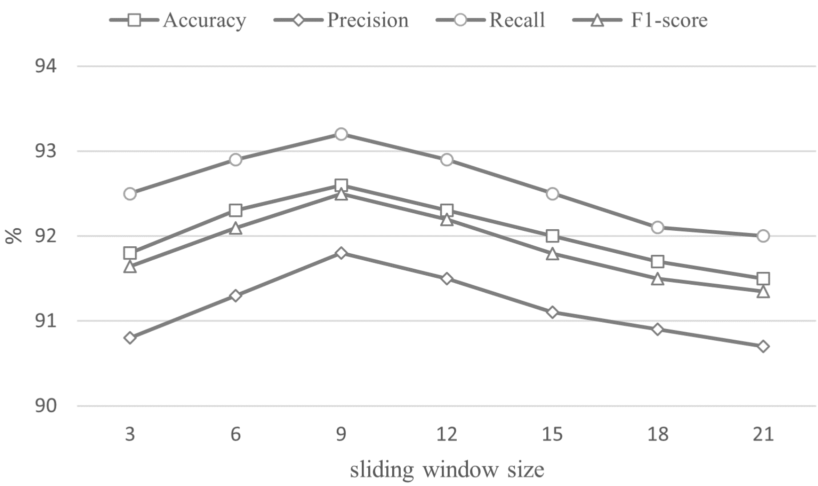

Figure 3. Hyperparameter search

4.3.2 Method Comparative Analysis

To verify the effectiveness of our proposed RTD-UCF method, we select the following rumor detection methods as baselines, and the experimental results are presented in Table 5.

SVM-TS [33]: A rumor detection method based on artificial feature engineering.

LSTM-CNN [6]: A rumor detection method that combines LSTM and CNN to extract semantic features.

FNDNet [5]: A rumor detection method utilizes CNN to extract semantic features.

TDRD [8]: A Rumor detection method combining topic information and semantic features.

LSTM-Attention [38]: A rumor detection method that integrates user, content statistical, and semantic features.

GAN_based [7]: A rumor detection approach using Adversarial Networks to enhance semantic features learning.

gzip [44]: A parameter-free classification method that combines a primary compressor such as gzip with a k-nearest-neighbor classifier.

According to the experimental results presented in Table 5, all methods that use deep learning models to automatically learn the semantic features of rumor content perform better than the SVM-TS method based on manual feature engineering. This is because manually designed features lack comprehensiveness and flexibility

Methods	Accuracy	Precision	Recall	F1-score
gizp	82.6	83.4	83.6	83.5
SVM-TS	83.1	84.6	84.2	84.4
FNDNet	85.6	85.9	86.4	86.1
LSTM-CNN	86.2	86.3	86.9	86.6
TDRD	87.5	87.6	86.8	87.2
GAN-based	87.8	86.7	88.4	87.5
LSTM-Attention	90.0	90.2	91.7	90.9
RTD-UCF	**92.6**	**91.8**	**93.2**	**92.5**

Table 5. Experimental results

and cannot represent the deep semantic features of rumors. Even though gzip was not explicitly designed for rumor detection tasks and does not require parameter training, it still achieved an F1-score of 83.5 %, demonstrating the effectiveness of this simple, lightweight, and versatile method. The LSTM-CNN model performs slightly better than FNDNet because it combines LSTM and CNN to learn spatial and temporal features of semantics simultaneously. TDRD incorporates topic and semantic features, leading to higher accuracy than FNDNet and LSTM-CNN, which only use semantic features. The GAN-based approach enhances the learning of rumor semantic features through the mutual promotion of the generator and discriminator in the adversarial network, resulting in improved accuracy compared to FNDNet and LSTM-CNN. The performance of LSTM-Attention is significantly better than other baseline methods because it uses an Attention-based LSTM to extract text semantic features and combines user and content statistical features, providing a more comprehensive feature selection for the classification model. Our proposed RTD-UCF method learns the semantic features of the content through W-GAT and explores two deep-level features: user rationality and professionalism. As a result, it achieves the highest values in all metrics, with an increase in accuracy, precision, recall, and F1-score of 2.6, 1.6, 1.5, and 1.6, respectively, compared to the best baseline method LSTM-Attention, demonstrating the effectiveness of our approach.

4.3.3 Generalizability Analysis

To assess the generalization capability of the RTD-UCF method, we subdivided the experimental data into four domains: politics (P), economics (E), health (H), and society (S). For each iteration, three domains were used as training and validation sets, while the remaining domain served as the test set. The experimental results are presented in Table 6. In the table, (P, E, H) $\rightarrow$ S denotes that posts from the political, economic, and health domains were used for training and validation, while posts from the society domain were used for testing; similar procedures were followed for the other three groups. F1_v and F1_t represent the F1-Score on the validation and test sets, respectively. The metric MD denotes the mean difference between the F1_v and F1_t values across the four groups, where a smaller value indicates better generalization capability of the method.

Methods	(P, E, H) → S		(P, E, S) → H		(P, H, S) → E		(E, H, S) → P		MD
	F1_v	F1_t	F1_v	F1_t	F1_v	F1_t	F1_v	F1_t	
SVM-TS	83.3	82.3	83.4	82.4	83.3	82.1	83.2	82.0	1.1
FNDNet	86.4	81.5	86.2	81.2	86.0	81.3	86.2	81.6	4.8
LSTM-CNN	86.8	81.1	85.8	81.2	86.7	81.0	85.9	81.5	5.1
TDRD	88.1	82.3	88.2	82.4	88.0	81.8	88.3	82.2	6.0
GAN-based	88.7	83.2	88.5	83.6	88.6	82.6	88.5	82.8	5.5
LSTM-Attention	90.6	87.0	90.5	87.2	90.7	86.7	90.8	86.9	3.7
RTD-UCF	93.0	90.0	92.9	90.5	93.1	90.3	93.0	90.9	2.6

Method gzip does not require training, so it is not considered in the generalization analysis.

Table 6. Generalization ability experimental results

The experimental results show that the generalization capability of all deep learning-based methods is significantly inferior to that of the SVM-TS method based on feature engineering. This discrepancy may stem from manually crafted features exhibiting global characteristics, resulting in minor differences across various domains. Deep learning methods in scenarios with limited datasets tend to capture superficial features highly correlated with the dataset, leading to weaker generalization capabilities. Additionally, as models become more complex, their learning capacity increases while their generalization ability decreases. There are substantial differences in topic information across different domains, leading to poor generalization capability of the TDRD method that incorporates topic information. The LSTM-Attention method exhibits relatively better generalization capability than other deep learning methods, possibly due to its integration of user and content statistical features. Our RTD-UCF method, which explores users' deep features alongside traditional statistical features, exhibits superior generalization capability compared to other deep learning methods.

4.3.4 Ablation Experimental Analysis

We conduct eight ablation experiments to analyze the contribution of different feature types to the RTD-UCF. Methods ①-⑤ remove user basic features F_{U_Basic}, user rationality Rat, user professionalism Pro, content statistical features F_{C_Sta}, and content semantic features F_{C_Sem}, respectively. Method ⑥ replaces W-GAT with BERT to learn semantic features. Methods ⑦ and ⑧ replace W-GAT with the standard GAT and GCN, respectively. Table 7 shows each method's accuracy, precision, recall, and F1-score changes compared to RTD-UCF.

From Table 7, we can observe that each type of feature plays a different role, and removing or replacing any of them will affect the performance of RTD-UCF. The user's basic features, such as user level, certification status, and number of follow-

Methods	$\Delta_{accuracy}$	$\Delta_{precision}$	Δ_{recall}	$\Delta_{F1-score}$
① (-) F_{U_Basic}	−1.0	−0.8	−1.0	−0.9
② (-) Rat	−0.6	−0.7	−0.5	−0.6
③ (-) Pro	−0.5	−0.4	−0.4	−0.4
④ (-) F_{C_Sta}	−0.6	−0.5	−0.5	−0.5
⑤ (-) F_{C_Sem}	−9.2	−8.4	−9.6	−9.0
⑥ W-GAT → BERT	−0.4	−0.4	−0.5	−0.5
⑦ W-GAT → GAT	−0.5	−0.4	−0.4	−0.4
⑧ W-GAT → GCN	−0.7	−0.8	−0.7	−0.7

Table 7. Ablation experiment results

ers, can reflect the user's credibility. Therefore, removing the user's basic features F_{U_Basic} will decrease the performance of RTD-UCF. Rational users are less likely to believe and spread rumors, and users with professional knowledge in a particular field are less likely to spread rumors in that field. Therefore, removing the features of user rationality Rat and professionalism Pro will also reduce the performance of RTD-UCF. The symbols, emojis, URLs, and other information also play an auxiliary role in identifying rumors, so removing these content statistical features F_{C_Sta} will reduce the method's performance. Content semantic features F_{C_Sem} are crucial for real-time rumor classification, so removing them will significantly impact the performance of RTD-UCF. Social media posts often exhibit the characteristics of discretization and fragmentation. Graph neural networks can better capture the text's non-continuous and long-distance dependent semantic features. Therefore, replacing W-GAT with BERT to extract content semantic features led to a decline in the method's performance. Replacing W-GAT with GAT reduced the method's performance because W-GAT considers both the feature similarity between nodes and the edge weight when calculating the attention coefficient, which can get more accurate global information. Replacing W-GAT with GCN has a more significant impact on the performance of RTD-UCF than replacing it with GAT because compared to the average pooling of GCN, the attention mechanism in GAT allows nodes to more flexibly aggregate information from neighboring nodes, thereby improving the performance of the model.

4.3.5 Experimental Results on Twitter

To validate the performance of our method on other social media platforms, we conducted further comparative experiments on the publicly available PHEME dataset. The PHEME dataset includes posts published on Twitter about five breaking news events (comprising 2 094 rumors, 3 654 non-rumors, and the social characteristics of the users who posted them). It is important to note that the PHEME dataset does not include the users' historical data, and we could not recollect this data, making it impossible to compute the features of user rationality and professionalism. Therefore, in the experiments on the PHEME dataset[4], we had to omit these two features, referring to the simplified method as RTD-UCF^{-}. The experimental results are shown in Table 8. Despite the absence of user rationality and professionalism features, our method outperformed the baseline methods on the PHEME dataset, demonstrating that our approach performs well on social media platforms with different linguistic and cultural backgrounds.

Methods	Accuracy	Precision	Recall	F1-score
gizp	73.5	72.2	72.3	72.2
SVM-TS	78.3	69.2	73.1	71.1
FNDNet	79.7	79.3	80.4	79.8
LSTM-CNN	80.4	80.1	81.1	80.6
TDRD	82.7	81.3	78.6	79.9
GAN-based	82.7	81.6	79.1	80.3
LSTM-Attention	83.0	82.3	81.6	81.9
RTD-UCF^{-}	**84.2**	**83.4**	**82.9**	**83.1**

Table 8. Experimental results on Twitter

4.3.6 Case Analysis

To clearly understand the role of the user rationality and professionalism features, we selected the following case for illustration.

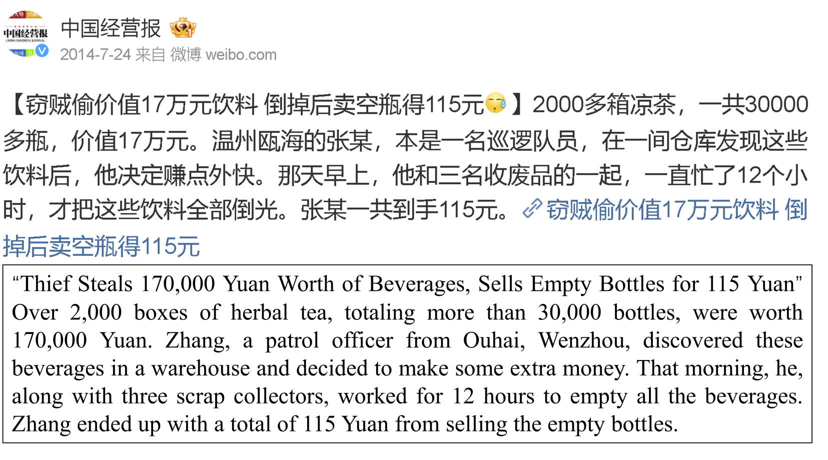

Figure 4. A non-rumor post on Weibo

Figure 4 shows a post published by the China Economic Daily on Weibo on July 24, 2014, titled "Thief Steals 170,000 Yuan Worth of Beverages, Sells Empty

[4] `https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078`

Bottles for 115 Yuan." Without considering the features of user rationality and professionalism, our method would incorrectly classify this post as a rumor due to its seemingly unbelievable content and low credibility. However, when incorporating the features of user rationality and professionalism, our method correctly classifies the post as non-rumor. This is because the China Economic Daily, as an authoritative official media outlet, has high rationality and professionalism, making the likelihood of it publishing a rumor extremely low.

5 CONCLUSION

Real-time rumor detection that does not rely on propagation features is one of the most effective ways to control the spread of rumors. However, current real-time rumor detection methods based solely on content semantic features have the issue of insufficient generalization ability. Mining additional features is an essential means to improve the performance of real-time rumor detection. In addition to integrating traditional user basic features and content statistical features, we use the user's historical posting data to mine two deep features: rationality and professionalism. To deal with the discretization and fragmentation of post texts, we utilize a graph attention network that considers the edge weights to learn deep semantic features of the content. The experimental results on our self-collected Weibo rumor dataset show that our method outperforms all the compared real-time rumor detection baselines.

Despite these improvements, our method may encounter limitations when processing large-scale data due to increased computational and storage requirements. Handling vast amounts of data in real-time scenarios can pose challenges related to efficiency and resource consumption. Future work will focus on optimizing our model's computational efficiency and exploring scalable solutions such as distributed computing frameworks and incremental learning techniques. Additionally, implementing effective feature selection and dimensionality reduction methods could further enhance the scalability and applicability of our approach in large-scale environments. We also plan to extend our method by incorporating multimodal rumor detection through joint analysis of textual content with attached images and videos to improve detection accuracy and robustness.

Acknowledgement

This work was supported in part by the Jiangxi Provincial Natural Science Foundation under Grant 20242BAB20074, in part by the National Natural Science Foundation of China under Grant 72061015, and in part by Jiangxi Provincial University Humanities and Social Sciences Research Project under Grant JC24219.

REFERENCES

[1] Shi, W.—Qin, W.: Public Opinion Monitoring and Guidance Analysis in the Process of News Dissemination from the Perspective of Big Data. Computing and Informatics, Vol. 42, 2023, No. 5, pp. 1114–1135, doi: 10.31577/cai_2023_5_1114.

[2] Cui, W.—Shang, M.: KAGN: Knowledge-Powered Attention and Graph Convolutional Networks for Social Media Rumor Detection. Journal of Big Data, Vol. 10, 2023, No. 1, Art. No. 45, doi: 10.1186/s40537-023-00725-4.

[3] Yuan, D.—Sun, H.: Reverse Intervention for Dealing with Malicious Information in Online Social Networks. Computing and Informatics, Vol. 39, 2020, No. 1-2, pp. 156–173, doi: 10.31577/cai_2020_1-2_156.

[4] Huang, X.—Ma, T.—Rong, H.—Jia, L.—Su, Y.: Dual Evidence Enhancement and Text–Image Similarity Awareness for Multimodal Rumor Detection. Engineering Applications of Artificial Intelligence, Vol. 153, 2025, Art. No. 110845, doi: 10.1016/j.engappai.2025.110845.

[5] Kaliyar, R. K.—Goswami, A.—Narang, P.—Sinha, S.: FNDNet – A Deep Convolutional Neural Network for Fake News Detection. Cognitive Systems Research, Vol. 61, 2020, pp. 32–44, doi: 10.1016/j.cogsys.2019.12.005.

[6] Ajao, O.—Bhowmik, D.—Zargari, S.: Fake News Identification on Twitter with Hybrid CNN and RNN Models. Proceedings of the 9th International Conference on Social Media and Society (SMSociety '18), ACM, 2018, pp. 226–230, doi: 10.1145/3217804.3217917.

[7] Cheng, M.—Li, Y.—Nazarian, S.—Bogdan, P.: From Rumor to Genetic Mutation Detection with Explanations: A GAN Approach. Scientific Reports, Vol. 11, 2021, No. 1, Art. No. 5861, doi: 10.1038/s41598-021-84993-1.

[8] Xu, F.—Sheng, V. S.—Wang, M.: Near Real-Time Topic-Driven Rumor Detection in Source Microblogs. Knowledge-Based Systems, Vol. 207, 2020, Art. No. 106391, doi: 10.1016/j.knosys.2020.106391.

[9] Ma, T.—Zhou, H.—Tian, Y.—Al-Nabhan, N.: A Novel Rumor Detection Algorithm Based on Entity Recognition, Sentence Reconfiguration, and Ordinary Differential Equation Network. Neurocomputing, Vol. 447, 2021, pp. 224–234, doi: 10.1016/j.neucom.2021.03.055.

[10] Sun, M.—Zhang, X.—Ma, J.—Xie, S.—Liu, Y.—Yu, P. S.: Inconsistent Matters: A Knowledge-Guided Dual-Consistency Network for Multi-Modal Rumor Detection. IEEE Transactions on Knowledge and Data Engineering, Vol. 35, 2023, No. 12, pp. 12736–12749, doi: 10.1109/TKDE.2023.3275586.

[11] Athira, A. B.—Kumar, S. D. M.—Chacko, A. M.: A Systematic Survey on Explainable AI Applied to Fake News Detection. Engineering Applications of Artificial Intelligence, Vol. 122, 2023, Art. No. 106087, doi: 10.1016/j.engappai.2023.106087.

[12] Jia, H.—Wang, H.—Zhang, X.: Early Detection of Rumors Based on Source Tweet-Word Graph Attention Networks. PLoS ONE, Vol. 17, 2022, No. 7, Art. No. e0271224, doi: 10.1371/journal.pone.0271224.

[13] Davoudi, M.—Moosavi, M. R.—Sadreddini, M. H.: DSS: A Hybrid Deep Model for Fake News Detection Using Propagation Tree and Stance Net-

work. Expert Systems with Applications, Vol. 198, 2022, Art. No. 116635, doi: 10.1016/j.eswa.2022.116635.

[14] Zhang, P.—Ran, H.—Jia, C.—Li, X.—Han, X.: A Lightweight Propagation Path Aggregating Network with Neural Topic Model for Rumor Detection. Neurocomputing, Vol. 458, 2021, pp. 468–477, doi: 10.1016/j.neucom.2021.06.062.

[15] Ma, J.—Gao, W.—Wong, K. F.: Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), 2017, pp. 780–717, doi: 10.18653/v1/P17-1066.

[16] Ma, J.—Gao, W.—Mitra, P.—Kwon, S.—Jansen, B. J.—Wong, K. F.—Cha, M.: Detecting Rumors from Microblogs with Recurrent Neural Networks. Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), 2016, pp. 3818–3824.

[17] Ma, J.—Gao, W.—Wong, K. F.: Rumor Detection on Twitter with Tree-Structured Recursive Neural Networks. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), 2018, pp. 1980–1989, doi: 10.18653/v1/P18-1184.

[18] Ma, J.—Gao, W.—Joty, S.—Wong, K. F.: An Attention-Based Rumor Detection Model with Tree-Structured Recursive Neural Networks. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 11, 2020, No. 4, Art. No. 42, doi: 10.1145/3391250.

[19] Chen, T.—Li, X.—Yin, H.—Zhang, J.: Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. In: Ganji, M., Rashidi, L., Fung, B. C. M., Wang, C. (Eds.): Trends and Applications in Knowledge Discovery and Data Mining (PAKDD 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11154, 2018, pp. 40–52, doi: 10.1007/978-3-030-04503-6_4.

[20] Song, C.—Yang, C.—Chen, H.—Tu, C.—Liu, Z.—Sun, M.: CED: Credible Early Detection of Social Media Rumors. IEEE Transactions on Knowledge and Data Engineering, Vol. 33, 2021, No. 8, pp. 3035–3047, doi: 10.1109/TKDE.2019.2961675.

[21] Huang, Q.—Zhou, C.—Wu, J.—Liu, L.—Wang, B.: Deep Spatial–Temporal Structure Learning for Rumor Detection on Twitter. Neural Computing and Applications, Vol. 35, 2023, No. 18, pp. 12995–13005, doi: 10.1007/s00521-020-05236-4.

[22] Wu, Z.—Pi, D.—Chen, J.—Xie, M.—Cao, J.: Rumor Detection Based on Propagation Graph Neural Network with Attention Mechanism. Expert Systems with Applications, Vol. 158, 2020, Art. No. 113595, doi: 10.1016/j.eswa.2020.113595.

[23] Choi, J.—Ko, T.—Choi, Y.—Byun, H.—Kim, C. k.: Dynamic Graph Convolutional Networks with Attention Mechanism for Rumor Detection on Social Media. PLoS ONE, Vol. 16, 2021, No. 8, e0256039 pp., doi: 10.1371/journal.pone.0256039.

[24] Bai, N.—Meng, F.—Rui, X.—Wang, Z.: Rumour Detection Based on Graph Convolutional Neural Net. IEEE Access, Vol. 9, 2021, pp. 21686–21693, doi: 10.1109/ACCESS.2021.3050563.

[25] Huang, X.—Ma, T.—Jin, W.—Rong, H.—Jia, L.—Yang, B.—Xie, X.: Multiview Spatio-Temporal Learning with Dual Dynamic Graph Convolutional Networks for Rumor Detection. IEEE Transactions on Computational Social Systems, Vol. 12,

2025, No. 5, pp. 2469–2479, doi: 10.1109/TCSS.2024.3520105.

[26] Vu, D. T.—Jung, J. J.: Rumor Detection by Propagation Embedding Based on Graph Convolutional Network. International Journal of Computational Intelligence Systems, Vol. 14, 2021, No. 1, pp. 1053–1065, doi: 10.2991/ijcis.d.210304.002.

[27] Zhang, Q.—Yang, Y.—Shi, C.—Lao, A.—Hu, L.—Wang, S.—Naseem, U.: Rumor Detection with Hierarchical Representation on Bipartite Ad Hoc Event Trees. IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, 2024, No. 10, pp. 14112–14124, doi: 10.1109/TNNLS.2023.3274694.

[28] Bian, T.—Xiao, X.—Xu, T.—Zhao, P.—Huang, W.—Rong, Y.—Huang, J.: Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, No. 1, pp. 549–556, doi: 10.1609/aaai.v34i01.5393.

[29] Chen, X.—Zhou, F.—Trajcevski, G.—Bonsangue, M.: Multi-View Learning with Distinguishable Feature Fusion for Rumor Detection. Knowledge-Based Systems, Vol. 240, 2022, Art. No. 108085, doi: 10.1016/j.knosys.2021.108085.

[30] Huang, X.—Ma, T.—Jia, L.—Zhang, Y.—Rong, H.—Alnabhan, N.: An Effective Multimodal Representation and Fusion Method for Multimodal Intent Recognition. Neurocomputing, Vol. 548, 2023, Art. No. 126373, doi: 10.1016/j.neucom.2023.126373.

[31] Zhu, Y.—Wang, G.—Li, S.—Huang, X.: A Novel Rumor Detection Method Based on Non-Consecutive Semantic Features and Comment Stance. IEEE Access, Vol. 11, 2023, pp. 58016–58024, doi: 10.1109/ACCESS.2023.3284308.

[32] Castillo, C.—Mendoza, M.—Poblete, B.: Information Credibility on Twitter. Proceedings of the 20th International Conference on World Wide Web (WWW '11), ACM, 2011, pp. 675–684, doi: 10.1145/1963405.1963500.

[33] Ma, J.—Gao, W.—Wei, Z.—Lu, Y.—Wong, K. F.: Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), 2015, pp. 1751–1754, doi: 10.1145/2806416.2806607.

[34] Ruchansky, N.—Seo, S.—Liu, Y.: CSI: A Hybrid Deep Model for Fake News Detection. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17), 2017, pp. 797–806, doi: 10.1145/3132847.3132877.

[35] Tu, K.—Chen, C.—Hou, C.—Yuan, J.—Li, J.—Yuan, X.: Rumor2vec: A Rumor Detection Framework with Joint Text and Propagation Structure Representation Learning. Information Sciences, Vol. 560, 2021, pp. 137–151, doi: 10.1016/j.ins.2020.12.080.

[36] Lotfi, S.—Mirzarezaee, M.—Hosseinzadeh, M.—Seydi, V.: Detection of Rumor Conversations in Twitter Using Graph Convolutional Networks. Applied Intelligence, Vol. 51, 2021, No. 7, pp. 4774–4787, doi: 10.1007/s10489-020-02036-0.

[37] Sun, M.—Zhang, X.—Zheng, J.—Ma, G.: DDGCN: Dual Dynamic Graph Convolutional Networks for Rumor Detection on Social Media. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, No. 4, pp. 4611–4619, doi: 10.1609/aaai.v36i4.20385.

[38] Singh, J. P.—Kumar, A.—Rana, N. P.—Dwivedi, Y. K.: Attention-Based

LSTM Network for Rumor Veracity Estimation of Tweets. Information Systems Frontiers, Vol. 24, 2022, No. 2, pp. 459–474, doi: 10.1007/s10796-020-10040-5.

[39] Kaliyar, R. K.—Goswami, A.—Narang, P.: DeepFakE: Improving Fake News Detection Using Tensor Decomposition-Based Deep Neural Network. The Journal of Supercomputing, Vol. 77, 2021, No. 2, pp. 1015–1037, doi: 10.1007/s11227-020-03294-y.

[40] Liu, Y.—Jin, X.—Shen, H.—Bao, P.—Cheng, X.: A Survey on Rumor Identification over Social Media. Chinese Journal of Computers, Vol. 47, No. 7, pp. 1536–1558 (in Chinese).

[41] Morris, M. R.—Counts, S.—Roseway, A.—Hoff, A.—Schwarz, J.: Tweeting Is Believing? Understanding Microblog Credibility Perceptions. Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12), 2012, pp. 441–450, doi: 10.1145/2145204.2145274.

[42] Inan, E.: ZoKa: A Fake News Detection Method Using Edge-Weighted Graph Attention Network with Transfer Models. Neural Computing and Applications, Vol. 34, 2022, No. 14, pp. 11669–11677, doi: 10.1007/s00521-022-07057-z.

[43] Huang, X.—Wu, Z.—Wang, G.—Li, Z.—Luo, Y.—Wu, X.: ResGAT: An Improved Graph Neural Network Based on Multi-Head Attention Mechanism and Residual Network for Paper Classification. Scientometrics, Vol. 129, 2024, No. 2, pp. 1015–1036, doi: 10.1007/s11192-023-04898-w.

[44] Jiang, Z.—Yang, M.—Tsirlin, M.—Tang, R.—Dai, Y.—Lin, J.: "Low-Resource" Text Classification: A Parameter-Free Classification Method with Compressors. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (Eds.): Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023). 2023, pp. 6810–6828, doi: 10.18653/v1/2023.findings-acl.426.

Yu Zhu received his B.Sc. in auditing from the Pearl River College of Tianjin University of Finance and Economics, China, in 2022 and his M.Sc. in administrative management from the Jiangxi University of Finance and Economics, China, in 2025. He is currently working towards his Ph.D. in public economics and management at the Jiangxi University of Finance and Economics. His research interests include public crisis management and internet public opinion analysis.

Gensheng Wang received his M.Sc. degree in computer software and theory from the Jiangxi Normal University, Nanchang, China, in 2005, and his Ph.D. degree in management science and engineering from the Jiangxi University of Finance and Economics, Nanchang, China, in 2011. From 2014 to 2021, he was Assistant Professor and Master Tutor of Jiangxi Finance and Economics. Since 2021, he has been Professor at the Jiangxi University of Finance and Economics. He has published more than 20 research papers on Internet public opinion and presided over 2 National Natural Science Foundation of China projects. His main research interests include Internet public opinion and data mining.

Xuejian Huang is Lecturer at the Jiangxi University of Finance and Economics. He received his Bachelor's degree in information security from the Nanchang University in 2013 and his Master's degree in software engineering from the Nanchang University in 2016. Currently, he is a Ph.D. student at the Nanjing University of Information Science and Technology. His primary research interests include online public opinion and data mining.

Wenwen Jin received her Bachelor's degree from the Dongbei University of Finance and Economics in 2020. She is currently pursuing her Master's degree at the East China Jiaotong University. Her research interests primarily focus on online public opinion analysis.

Sheng Li received his B.Sc. and M.Sc. degrees in administrative management and management science and engineering from the Hunan University, Hunan, China, in 2007 and 2011, respectively. Since 2017, he has been Associate Professor at the Jiangxi University of Finance and Economics, and in 2021, he was Doctoral Supervisor at the Jiangxi University of Finance and Economics. He has published seven research papers on emergency management, two monographs, and presided over 1 National Social Science Fund project and 1 National Social Science Fund major project sub-project. His main research interests include emergency management, environmental governance, and social governance.