TSTL: A Hybrid TextRank-LP and TDNN-LSTM Model for Public Opinion Monitoring in Emergency Events

Shiyu Wang[1,2]
[1]Faculty of Humanities and Arts, Macau University of Science and Technology, Macao 999078, China
[2]Jiangxi Technical College of Manufacturing, Nanchang 330095, China
Email: qqqwsy@163.com

Keywords: TextRank-LP, online public opinion, keyword extraction, LSTM, TDNN

Received: April 27, 2025

Online public sentiment during emergencies often hinders effective crisis management. Timely and accurate identification and prediction of this sentiment are vital, yet existing approaches face challenges related to high identification delays and low accuracy. To address these issues, this study proposes a model for the evolution of network public opinion based on the TextRank-Label Propagation Algorithm (TextRank-LP). The model fully utilizes the information extraction capability of TextRank-LP, while integrating the strengths of Time Delay Neural Networks and Long Short-Term Memory networks in handling time-series data. In the validation of the keyword extraction algorithm, the Natural Language Processing and Chinese Computing dataset was used, which contains tens of thousands of text samples. The results showed that the accuracy of the proposed algorithm reached 95.1%, higher than the 92.1% of Term Frequency Across Document Frequency, 88.1% of Support Vector Machine, and 86.0% of Bidirectional Long Short Term Memory network (Bi-LSTM). The F1 value of the proposed algorithm reached 94.6%. In practical testing using the 2019 NBA China controversy dataset, the model achieved a 100% accuracy rate in identifying declining online public opinion and an average recognition accuracy rate of 92.5% for each stage. Meanwhile, the model's prediction time for online public opinion 12 months ahead is 21 minutes, far lower than Bi-LSTM's 47 minutes. These findings indicate that the proposed model demonstrates strong recognition and predictive capabilities for public sentiment in emergencies and provides a novel approach to studying the evolution of online sentiment in such events. This method offers valuable potential for advancing more accurate and efficient research in the field of public sentiment dynamics.

Povzetek: Prispevek predstavi model TSTL, ki združuje TextRank-LP, spektralno gručenje in TDNN-LSTM za spremljanje spletnega mnenja v krizah.

1 Introduction

Online public opinion during sudden events contains substantial false and exaggerated information, which can easily trigger negative emotions among the masses and even lead to the occurrence of extreme events in severe cases [1]. As deep learning and artificial intelligence continue to advance, their techniques, including natural language processing and knowledge graphing, have increasingly been utilized for monitoring online public opinion during emergencies. However, these methods still have certain flaws and require more accurate and efficient approaches to be applied in this domain [2-3]. The TextRank-Label Propagation Algorithm (TextRank-LP) is widely used in text summarization due to its real-time processing capabilities and low operational costs [4]. Additionally, the integration of neural networks can adapt to different types of data by combining the strengths of various neural networks [5]. The existing network public opinion monitoring technology often suffers from weight bias when processing large-scale network texts due to the complexity and diversity of text sources, as well as limitations in processing time series information, making it difficult to accurately predict the development trend of public opinion. Therefore, we propose a method for extracting keywords from online public opinion during. In addition, a monitoring model is introduced to track online public opinion in such situations, which integrates Time Delay Neural Networks (TDNN) and Long Short-Term Memory (LSTM) networks. The research hypothesis is that integrating Spectral Clustering (SC) can further improve keyword extraction performance compared to standard TextRank, and the time series information processing performance of TDNN-LSTM is better than that of standard LSTM or Bidirectional Long Short Term Memory network (Bi-LSTM). The model is expected to be applied in emergency public opinion monitoring and effectively prevent large-scale public opinion crises.

The innovation of this manuscript lies in: (1)

combining TextRank-LP algorithm and SC algorithm, optimizing the keyword extraction process by comprehensively considering the frequency, position, and part of speech information of words. (2) A multi-to-one TDNN-LSTM network architecture has been proposed, which is suitable for monitoring and predicting network public opinion in emergency situations. (3) Integrating the optimized TextRank-LP keyword extraction method and TDNN-LSTM network, achieving full process monitoring from text preprocessing to public opinion prediction.

2 Related works

TextRank has been widely used in various fields of research by scholars both domestically and internationally due to its advantages, such as no need for pre-training and strong adaptability to sentence structures. Sihombing's team developed a summarization system for Indonesian language articles based on TextRank. In experiments with 100 Indonesian articles, the system achieved an accuracy rate of 80% at compression rates of 50%, 40%, and 30% [6]. Gusra et al. developed an automated news summarization method based on TextRank's ability to identify key information, aiming to generate more concise news articles from a large number of online news sources. In tests, this method successfully condensed a sentence with 387 words down to 75 words [7]. To address the performance degradation of speakers in noisy environments, Benhafid et al. optimized TDNN using a time-limited self-attention mechanism and proposed a new hierarchical TDNN speaker model. In testing experiments, the method achieved an accuracy rate 6.85% higher than the comparison method [8]. Qiao's team applied TDNN to the auxiliary detection of heart sound signals. Their method fused hidden feature layers of heart sound signals within a certain time period using an improved TDNN and improved recognition accuracy by masking irrelevant heart sound frequencies. The experimental results showed that the method outperformed existing abnormal heart sound detection methods [9]. Mohbey et al. applied LSTM in public health research. To determine public perceptions of monkeypox, the research used an LSTM-based model to process a tweet dataset about monkeypox. The processing results achieved an accuracy rate of 91% [10]. To solve the problem of low accuracy in traditional deep learning-based waste classification models, Lilhore et al. proposed a sustainable intelligent waste classification method based on LSTM, improving the accuracy of the waste classification model to 95.45% [11].

The study of the evolution of online public opinion has also developed several mature methods and theories, which have been applied in practical monitoring. For example, Yuan's team explored the process of polarization in international attitudes by integrating the information diffusion process and the development of polarization behavior in a network public opinion monitoring model. The team hoped to find the main factors affecting public opinion by adjusting model parameters [12]. Ding et al. researched the factors influencing social media users' opinion identification behavior based on the Theory of Planned Behavior and Deterrence Theory. The findings showed that the severity and authenticity of online public opinion negatively moderate users' subjective norms and attitudes, thereby accelerating the spread of public opinion [13]. Zhang et al. studied the public opinion diffusion pattern during major epidemics by selecting three quantitative indicators: emotional enhancement, differences, and conversion rates. Based on this, they constructed the SIPINRS public opinion diffusion model. The model analysis revealed that the number of initial opinion spreaders significantly affects the development trend of public opinion [14]. Wang et al. proposed a keyword extraction method for online public opinion based on unsupervised spatiotemporal graphs. This method forms clustering topics through keyword similarity and understands the evolution of online public opinion by analyzing the relationship changes between topics. This method has been effectively tested on five datasets [15]. The related work summary table is shown in Table 1.

Table 1: Summary table of related work

Literature	Method	Data set	Result	Limitation
[6]	TextRank	Indonesian language online news collection	Compress 387-word sentences to 75 words (compression rate 80.6%) while retaining core information	Not considering semantic coherence, relying on title quality
[7]	TextRank + TF-IDF	100 Indonesian scientific articles	Accuracy reaches 80% at different compression rates	Insufficient elimination of long text redundancy
[8]	Time limited self attention + graded TDNN	VoxCeleb1/2	Speaker verification accuracy increased by 6.85%	High computational complexity and weak adaptability to dynamic environments
[9]	Improved TDNN + Dynamic Mask Encoder	PhysioNet Heart Sound Database	Abnormal heart sound detection with F1 value of 92.7%	Sensitive to mixed noise
[10]	CNN-LSTM	Monkeypox related tweet	The accuracy of emotion classification	Unprocessed satirical text

		dataset	is 91%	
[11]	CNN-LSTM + Transfer Learning	TrashNet image dataset	The classification accuracy is 95.45%	Real scene lighting changes affect accuracy
[12]	SIR model + dynamic network structure	Public opinion data on the controversy between China and the United States	The predicted F1 value for polarization trend is 88.3%	Not considering cross platform dissemination differences
[13]	Structural equation model	Emergency event social media data	The fitting index reaches 0.93	The relationship between unquantifiable behavior and actual dissemination
[14]	SIPINRS	Major Epidemic Public Opinion Dataset	The initial number of disseminators has a transmission range R^2 of 0.76	Emotional indicators rely on dictionary annotation
[15]	Unsupervised spatiotemporal graph attention	Public Event Dataset	Topic evolution correlation 0.89	Insufficient real-time performance

In summary, although there have been some achievements in the study of the evolution of online public opinion, current research methods still require pre-training on specific texts and have low accuracy in keyword extraction. TextRank-LP, however, can effectively extract keywords from texts without the need for pre-training. Therefore, this study constructs a model for keyword extraction and the study of the evolution of public opinion in emergencies based on TextRank-LP and the integration of TDNN and LSTM. The aim is to extract keywords from online public opinion quickly and accurately, grasp the development patterns of public opinion.

3 Online public opinion monitoring model integrating Textrank-LP and neural networks

3.1 Design of keyword extraction method for online public opinion based on optimized TextRank-LP

TextRank-LP is a text summarization algorithm based on PageRank and label propagation algorithms. It treats sentences as individual points or vectors and determines text similarity based on point similarity [16-17]. The online public opinion of sudden events is highly dynamic and unpredictable, and traditional supervised learning-based keyword extraction methods often require a large amount of labeled data for training, which consumes a lot of time. TextRank-LP, based on graph structure, can utilize the intrinsic information of text to calculate the importance of keywords, avoiding dependence on a large amount of annotated data and enabling quick response and processing of new public opinion data. Therefore, this study designs a keyword extraction method for emergency online public opinion based on TextRank-LP's text summarization ability, with the specific process of keyword extraction shown in Figure 1.

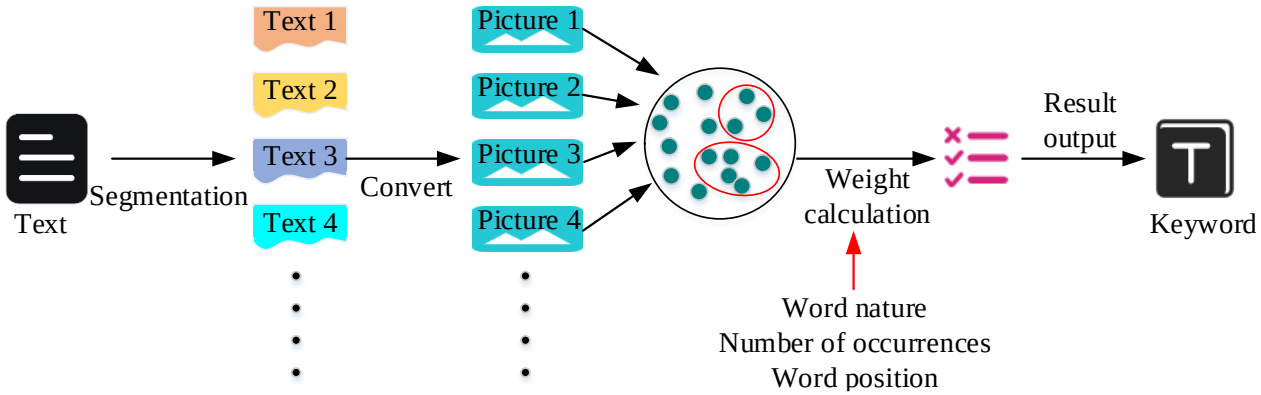


Figure 1: Textrank-LP keyword extraction process diagram

As shown in Figure 1, when extracting keywords, TextRank-LP transforms the text into an undirected graph, where each word is treated as a node in the graph. A node's weight is defined by the quantity of adjacent nodes

and the edges surrounding it. Based on this weight, it is determined whether a word is a keyword. The calculation process is shown in Equation (1).

$$S(v_i) = (1-d) + \sum_{j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}} S(v_j) \quad (1)$$

In Equation (1), $S(v_i)$ represents the weight value of point v_i, d is the damping factor, $In(v_i)$ is the set of all points pointing to point v_i, $Out(v_i)$ is the set of all points pointed to by point v_i, w is the weight of the edge connecting two nodes, and $S(v_j)$ is the weight value of point v_j. Then, a comprehensive weighting process considering word frequency, position, and part of speech is conducted. The calculation process is shown in Equation (2).

$$W(v_i) = W_1(v_i) * T - OT + W_2(v_i) * Lo + W_3(v_i) * Po \quad (2)$$

In Equation (2), $W(v_i)$ is the comprehensive weight. W_1, W_2, and W_3 represent the weights for word frequency, position, and part of speech, respectively. T and OT refer to the occurrence counts of a word and other words, while Lo and Po represent the position and part of speech attributes of the word. The position of words in a sentence often reflects their importance. Usually, words located at the beginning and end of a sentence are easier to express the core idea. Therefore, the sentence is divided into three regions: the beginning (S), middle (M), and end (E). The weights of each region are w_S, w_M, and w_E, adjusted according to actual needs to ensure that the sum of the weights of the three regions is 1. The importance of different parts of speech varies in text, and by analyzing a large amount of text, the weight of different parts of speech can be determined. For each word, assign corresponding weights based on its part of speech, such as nouns, verbs, adjectives, etc. The final weight value of TextRank-LP is calculated using Equation (3).

$$S(v_i) = (1-d) * W(v_i) + d * W(v_i) * \sum_{v_i \in In(v_j)} \frac{W_{ji}}{\sum_{v_k \in Out(v_j)} W_{jk}} S(v_j) \quad (3)$$

In Equation (3), w is replaced by the comprehensive weight W, which takes more information into account. The SC algorithm captures complex information relationships and groups similar information based on different features [18]. Using the SC algorithm for text classification can effectively improve the keyword extraction speed by keeping similar information within a smaller spatiotemporal range. Moreover, the Laplacian matrix in SC has global optimization properties during feature decomposition, reducing weight propagation bias during keyword extraction and improving accuracy. The SC algorithm process is shown in Figure 2.

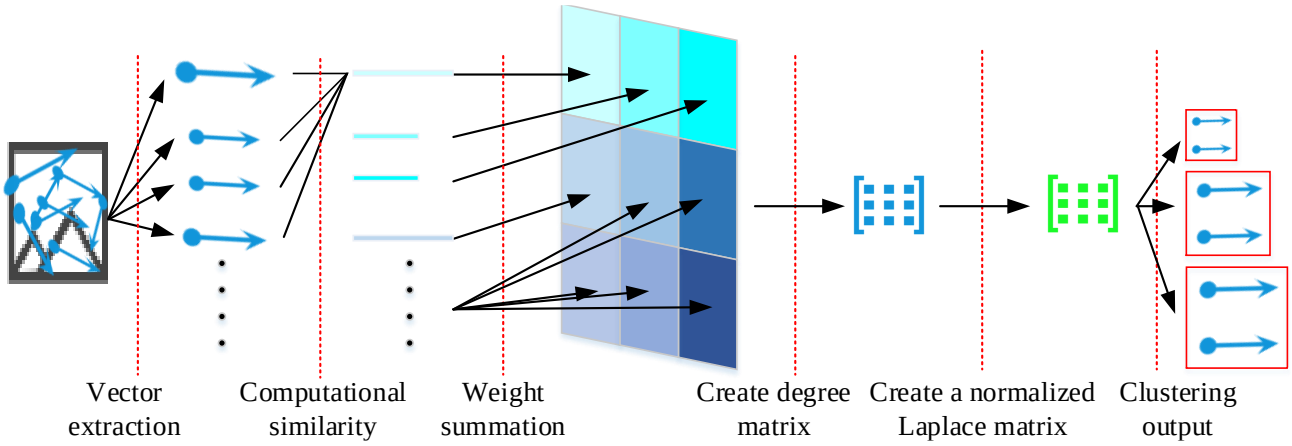


Figure 2: SC algorithm workflow diagram

As shown in Figure 2, before performing clustering analysis, the similarity between vectors is determined by the relationship between them, as calculated in Equation (4).

$$s_{i,j} = \left\| x_i - x_j \right\| \quad (4)$$

In Equation (4), i and j represent two different vectors in the image, $s_{i,j} = \left\| x_i - x_j \right\|^2$ is the Euclidean norm between vectors i and j, and x_i and x_j are the lengths of vectors i and j, respectively. Then, the similarity between the two vectors is determined based on the value of $s_{i,j} = \left\| x_i - x_j \right\|^2$, and the similarity matrix S

is constructed, as shown in Equation (5).

$$\begin{cases} 0, if\ s_{i,j} > \varepsilon \\ \varepsilon, if\ s_{i,j} \le \varepsilon \end{cases} \quad (5)$$

In Equation (5), ε is the specified range radius. If the result is nonzero, it indicates the two vectors are similar. However, binary truncation is too restrictive, so this study uses radial basis function kernel (RBF) instead of binary truncation, combined with adaptive thresholding, to more accurately measure the similarity between text feature vectors. RBF kernel maps feature vectors to high-dimensional space and calculates their similarity, as shown in Equation (6).

$$Sim(x_i, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2} \right) \quad (6)$$

In Equation (6), x_i and x_j are text feature vectors. σ is the kernel width, which controls the decay rate of similarity. Then, dynamically adjust the threshold θ of the similarity matrix, as shown in Equation (7).

$$\theta = \mu + k\chi \quad (7)$$

In Equation (7), μ and χ are the mean and standard deviation of the feature vector length, respectively. χ is the adjustment coefficient. By adaptively adjusting the threshold of the similarity matrix, it is possible to better cope with datasets of different types and sizes, and improve the accuracy and robustness of clustering. Afterwards, the degree matrix D is created based on the sum of the weights of each vector with other vectors in the S matrix, as calculated in Equation (8).

$$D_{ij} = \sum_{j=1}^{n} S_{ij} \quad (8)$$

In Equation (8), S_{ij} represents the similarity weight of a specific vector in the S matrix. In the degree matrix D, the value of each vector is the sum of the weights of all other vectors in the range ε. Then, the Laplacian matrix of D is computed and normalized, as shown in Equation (9).

$$\begin{cases} L = D - S \\ L' = D^{-1}L \end{cases} \quad (9)$$

In Equation (9), L is the Laplacian matrix, and L' is the normalized Laplacian matrix. The eigenvectors corresponding to the eigenvalues of the normalized matrix are then used as samples for classification, yielding the final multidimensional classification samples. In large-scale text data, sentence length is a simple and low-cost feature that requires minimal preprocessing, and can partly reflect the information content and complexity of the text. Therefore, when implementing spectral clustering, this study chose sentence length as the key classification feature. Divide the text into three clusters based on sentence length: short text (no more than 15 words), medium text (16-40 words), and long text (more than 40 words). In short texts, higher weights are assigned to the beginning and end positions of sentences. In long texts, nouns are given higher weights and verbs are given lower weights. Implement TF-IDF smoothing for medium text to avoid distortion of short sentence frequency. This classification ensures that the TextRank-LP weight calculation shown in equation (2) is performed within the same sentence structure, eliminating statistical distribution differences across length texts. Finally, the SC algorithm is integrated with TextRank-LP to form the TS algorithm, with the specific process shown in Figure 3.

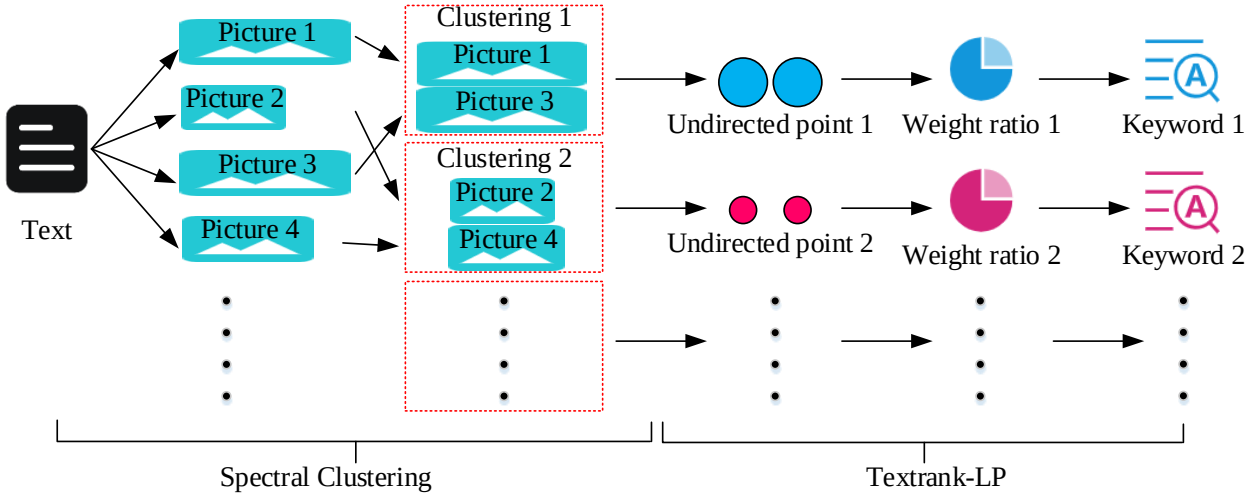


Figure 3: TS algorithm keyword extraction flow chart

As shown in Figure 3, the TS algorithm first performs text preprocessing. Secondly, use the SC algorithm to cluster text fragments. Then, treat each word as a node in the graph and use TextRank-LP to calculate the weight of each node. Subsequently, the results of spectral clustering are combined with the weight calculation of TextRank-LP to obtain the final comprehensive weight, and keywords are extracted based on the weight size. Finally, sort the keywords based on their comprehensive weights and output a list of keywords.

3.2 Construction of neural network-based public opinion research model

However, the TS algorithm can only extract keywords, and a complete study of public opinion evolution requires additional processes such as monitoring range division and data analysis [19]. Therefore, the study proposes an emergency online public opinion monitoring process with the TS algorithm as the keyword extraction module. The specific steps of this process are shown in Figure 4.

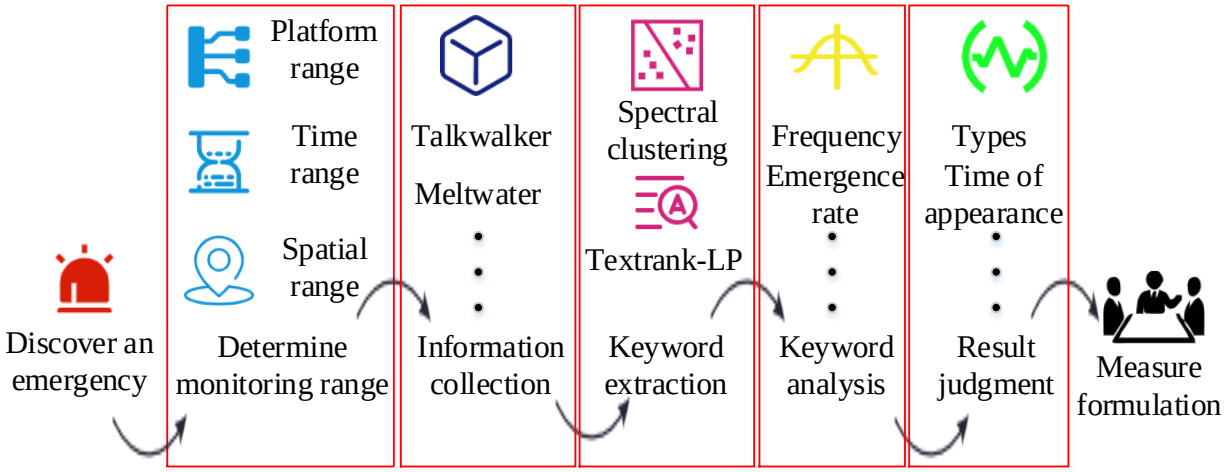


Figure 4: Emergency online public opinion monitoring process

As shown in Figure 4, before keyword extraction, the monitoring range is determined. Then, computer software is used to collect online texts from different channels. Next, the TS algorithm is applied to extract keywords, identifying frequently occurring and widely covered words as keywords. To grasp the development trend of emergency online public opinion, the properties of these keywords and their appearance speed are further analyzed. TDNN captures local contextual changes, while LSTM networks effectively process time-series information and capture global changes [20]. Therefore, to better grasp the temporal changes of keywords, the research integrates TDNN and LSTM to create a keyword analysis network that combines the advantages of both. The structure of this network is shown in Figure 5.

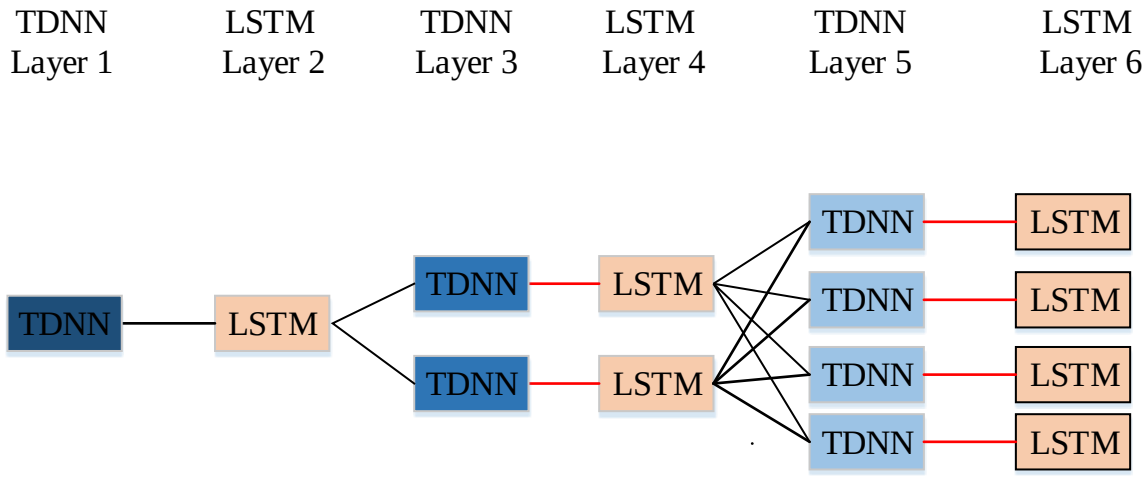


Figure 5: Structure diagram of TDNN-LSTM network

As shown in Figure 5, the TDNN-LSTM network is formed by connecting TDNN and LSTM sequentially, and belongs to a many to one time series model. It includes 6 hidden layers, among which the 1st, 3rd, and 5th layers are TDNN structures. The second, fourth, and sixth layers are LSTM structures, with each TDNN and LSTM layer containing 512 nodes. The input layer of TDNN-LSTM receives time series data with a length of T, and the feature dimension of each time step is D. After multiple layers of processing, it outputs the prediction result of a single time step. Using the Adam optimizer, the learning rate parameter is set to 0.001 and adjusted based on changes in loss during training, using Mean Squared Error (MSE) as the loss function. The output of a single TDNN neuron is represented in Equation (10).

$$h(t) = f(\sum_{n=1}^{N}[w_n * x(t+n) + b]) \quad (10)$$

In Equation (10), f is the activation function, N is the total time the information stays in that layer, t represents the input time, n is the time step, w_n is the weight for different time steps, and b is the bias. The total output of each layer neuron is represented in Equation (11).

$$h(t) = f(\sum_{m=1}^{M}\left[\sum_{n=1}^{N} win * x_m(t+n) + b_m\right]) \quad (11)$$

In Equation (11), M is the total number of neurons in the layer, x_m represents the m-th neuron, and b_m represents the bias of the m-th neuron. Then, the LSTM subunit processes the short-sequence features from the TDNN layer, extracts the dependency relationships between different short-sequence feature segments, and stores and outputs the features with higher correlations. The computational process for the forget gate, input gate, and output gate of the LSTM subunit is shown in Equation (12).

$$\begin{cases} f_t = \sigma(w_f \bullet (h_{t-1}, x_t) + b_f) \\ i_t = \sigma(w_i \bullet (h_{t-1}, x_t) + b_i) \\ o_t = \sigma(w_o \bullet (h_{t-1}, x_t) + b_o) \end{cases} \quad (12)$$

In Equation (12), h_{t-1} is the previous time output state, and x_t is the current time input. σ is the activation function. w_f, w_i, and w_o are the forget, input, and output gate forget weights, while b_f, b_i, and b_o are the neuron biases for the corresponding layers. f_t, i_t, and o_t are the output data for the corresponding layers. The data is output by the output gate, and the internal data of the LSTM subunit is fully updated, as shown in Equation (13).

$$C_t = f_t \bullet C_{t-1} + [\tanh(w_c \bullet (h_{t-1}, x_t) + b_c)] \bullet i_t \quad (13)$$

In Equation (13), C_{t-1} represents the state of the neuron before the update, and C_t represents the updated state. The final output of the neuron requires activation using the tanh function, as shown in Equation (14).

$$h_t = o_t \bullet \tanh(C_t) \quad (14)$$

In Equation (14), h_t refers to the final output of the memory cell at time t. By combining the keyword extraction advantage of the TS algorithm with the powerful time-series correlation processing ability of the TDNN-LSTM network, the study constructs the TSTL model. The model structure is shown in Figure 6.

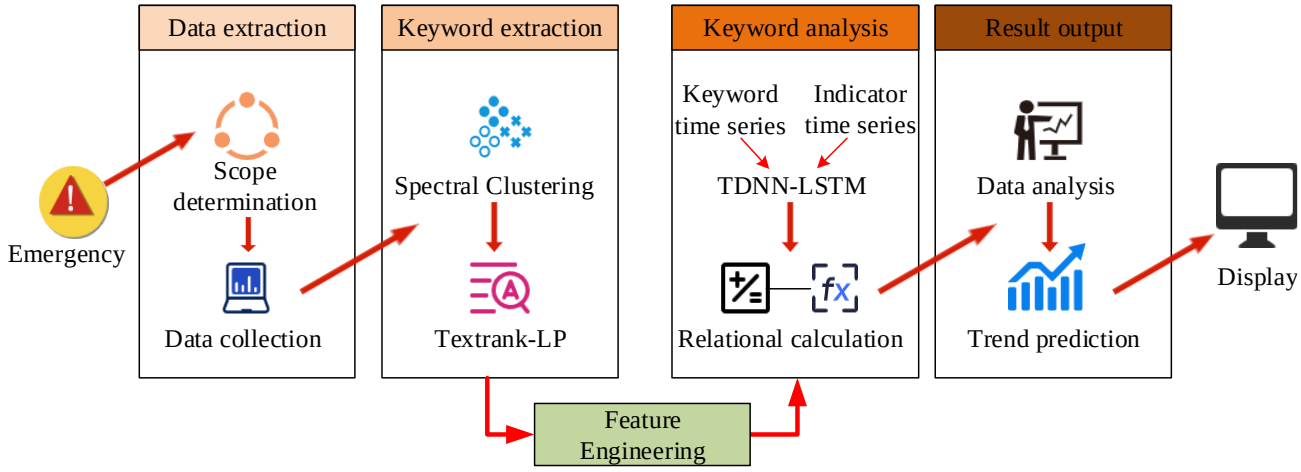


Figure 6: Workflow diagram of TSTL model for online public opinion research

As shown in Figure 6, in the TSTL model, the first step is to perform clustering analysis on the text from various network environments, a process completed by the SC algorithm in the model. Next, keyword extraction is carried out, performed by the TextRank-LP part, which fully considers factors such as word type and position. The TDNN-LSTM network is then used to analyze the time-series relationships, such as the occurrence time of keywords. The input of the TDNN-LSTM network includes two parts: keyword time series and exported indicator time series. In this process, the study selects the propagation speed, attention, and dissent ratio of texts with keywords as the distinguishing criteria. The calculation of the propagation speed is shown in Equation (15).

$$sp = \frac{\Delta R}{\Delta t} \quad (15)$$

In Equation (15), ΔR represents the change in reading volume per unit time, and Δt represents the time unit. The faster the propagation speed, the more likely it is for a network public opinion event to occur. The process for calculating attention is shown in Equation (16).

$$A = \alpha * r + \beta * d + \gamma * c \quad (16)$$

In Equation (16), α, β, and γ are the weight coefficients, while r, d, and c represent the reading volume, transmission volume, and comment volume, respectively. The larger the attention, the more likely it is for a network public opinion event to occur. The method for calculating the dissent ratio is shown in Equation (17).

$$R = \frac{V_n}{V} * 100\% \quad (17)$$

In Equation (17), V and V_n represent the total voice and the non-mainstream voice, respectively. A larger dissent ratio indicates a greater divergence within the public. The final model uses the temporal variations of these parameters to predict whether a network public opinion event will occur during an emergency.

4 Validation of emergency online public opinion research model

4.1 Keyword extraction performance verification of TS algorithm

To evaluate the performance of the TS algorithm, the study used the Natural Language Processing and Chinese Computing (NLPCC) dataset for testing. The NLPCC dataset, constructed by the Chinese Information Society, is a Chinese natural language processing evaluation dataset, which contains a large amount of data for tasks such as sentiment analysis and text summarization. This study used NLPCC2014 Task 2, an open-source dataset obtained through the official website of the Chinese Information Society, including 5000 positive and negative training samples and 1250 positive and negative testing samples. Term Frequency-Inverse Document Frequency (TF-IDF), Bi-LSTM, and Support Vector Machine (SVM) were used as comparison models to verify the effectiveness of the proposed algorithm in keyword extraction. All algorithms were run under the same conditions during the experiment. The experimental settings are shown in Table 2.

Table 2: Experimental settings and configurations

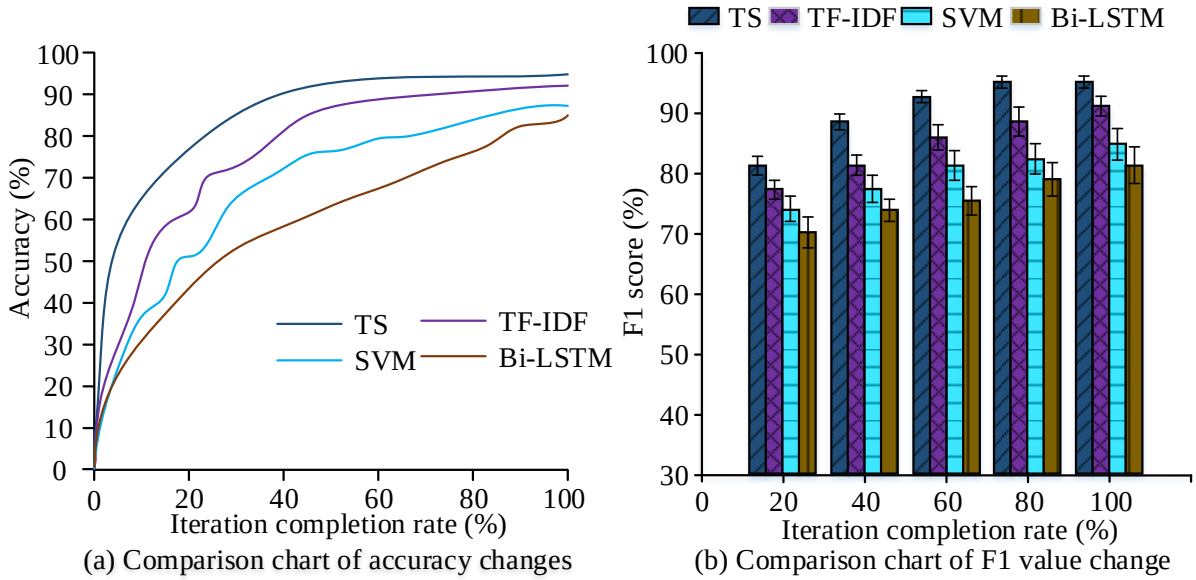
/	Category	Version	Category	Version
Software	Operating system	Windows 11	Deep learning framework	PyTorch 2.1
	Programming language	Python 3.8	Visualization library	Mtplotlib
Hardware	CPU	Ryzen 7 9800X3D	GPU	NVIDIA RTX 3080
	Memory	64GB RAM	Storage	128 GB SSD

To quantify the accuracy of SC algorithm pre-classification, the study used clustering purity to evaluate the length classification effect. The experiment used 5000 texts from the NLPCC dataset, labeled as 1520 short texts, 2480 medium texts, and 1000 long texts according to their actual lengths. The confusion matrix between the pre classification results of the SC algorithm and the real labels is shown in Table 3.

Table 3: Confusion matrix between pre classification results of SC algorithm and real labels

Real category	Prediction category			Recall/%
	Short text	Medium Text	Long Text	
Short text	1385	1335	0	91.1
Medium Text	98	2252	130	90.8
Long Text	0	95	905	90.5
Precision/%	93.4	86.9	87.6	-

From Table 3, it can be seen that the SC algorithm pre classification can accurately separate texts of different lengths, with a recall rate of over 90%. After calculation, the clustering purity is 89.6%, indicating that the SC algorithm achieves better clustering performance. First, to verify the learning capability of the TS algorithm, the NLPCC dataset was used to train the algorithm iteratively. The accuracy and F1 score changes during the iteration were compared and analyzed. The results are shown in Figure 7.



(a) Comparison chart of accuracy changes (b) Comparison chart of F1 value change

Figure 7: Comparison of accuracy changes and F1 values of each model

As shown in Figure 7(a), at the end of the iteration, the accuracy of the TS algorithm was the highest at 95.1%, while the TF-IDF algorithm achieved 92.1%, SVM 88.1%, and Bi-LSTM network 86.0%. Additionally, the TS algorithm started to converge at 40% of the iteration, earlier than the comparison algorithms. This indicates that the TS algorithm has the strongest learning ability and higher learning efficiency. As shown in Figure 7(b), with the increase in the iteration rate, the F1 score of the TS algorithm increased from 81.5% to 94.6%. The TF-IDF algorithm increased from 78.1% to 92.0%, SVM from 74.5% to 86.0%, and the Bi-LSTM network from 70.3% to 81.2%. This shows that the TS algorithm has stronger classification ability. Next, to verify the work efficiency of the TS algorithm, the text processing speed of each algorithm was compared. The results are shown in Figure 8.

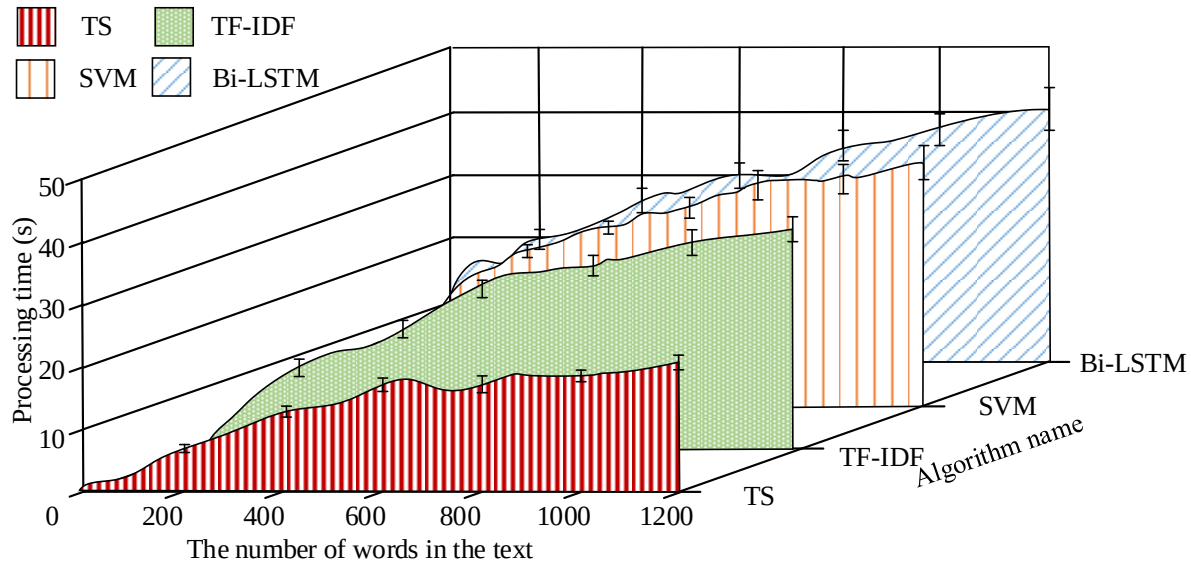


Figure 8: Comparison of processing time for texts of different lengths

As shown in Figure 8, all algorithms exhibited an increasing processing time as the text length increased. However, for all text lengths, the TS algorithm took less time than the comparison algorithms. When the text length was 600 characters, the processing time of the TS algorithm was 17s, while the TF-IDF algorithm took 20s, SVM took 23s, and the Bi-LSTM network took 30s. When the text length was 1200 characters, the processing time of the TS algorithm was 20s, while the TF-IDF algorithm took 29s, SVM took 30s, and the Bi-LSTM network took 40s. In conclusion, the TS algorithm exhibited high efficiency in keyword extraction for texts of various lengths. To verify the accuracy of keyword extraction by the TS algorithm, the recognition results of each algorithm were compared. The results are shown in Figure 9.

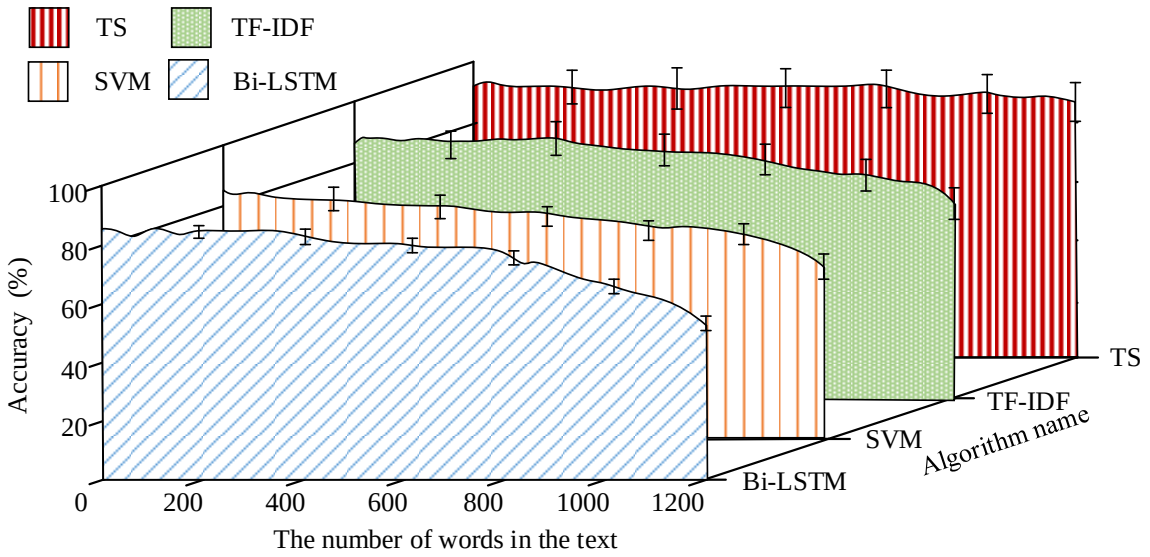


Figure 9: Comparison of keyword extraction accuracy for texts of different lengths

As shown in Figure 9, all algorithms showed a trend where accuracy decreased as text length increased. However, for all text lengths, the TS algorithm had a higher accuracy than the comparison algorithms. When the text length was 400 characters, the accuracy of the TS algorithm was 93.1%, while the TF-IDF algorithm's accuracy was 90.1%, SVM's accuracy was 82.3%, and the Bi-LSTM network's accuracy was 78.2%. When the text length was 1200 characters, the accuracy of the TS algorithm was 90.1%, while the TF-IDF algorithm's accuracy was 76.8%, SVM's accuracy was 63.2%, and Bi-LSTM's accuracy was 54.5%. In conclusion, the TS algorithm had a higher accuracy in keyword extraction for texts of different lengths.

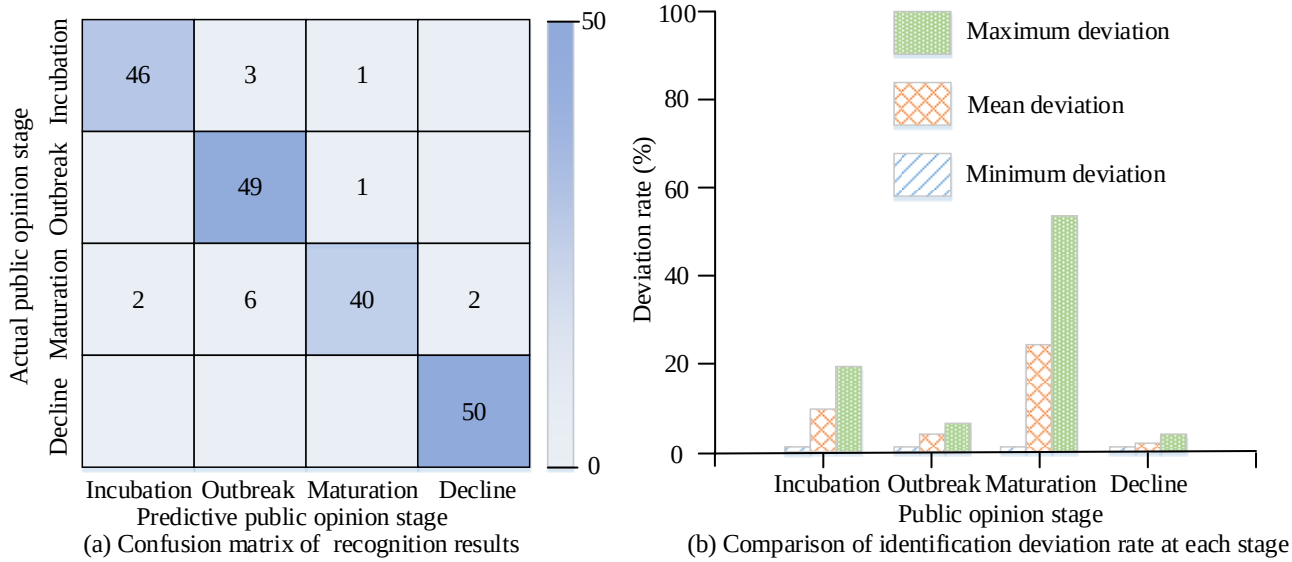
4.2 Practical application test of TSTL model

The 2019 NBA and China controversy event-related data was used as the dataset. The data mainly comes from Weibo. This study collected relevant information from public social media platforms through web crawling technology, strictly following the platform's terms of use and laws and regulations during the collection process. The total monitored public opinion information was 2538. Firstly, perform data preprocessing by using text similarity clustering methods to remove a large amount of duplicate or highly similar content. Using keyword filtering and sentiment analysis techniques, select data that is directly related to NBA China and has clear emotional tendencies. Useless characters, HTML tags, etc., were removed, and encoding issues and typos in the text were corrected. Divide the dataset into a training set (70%), a validation set (15%), and a testing set (15%). Determine hyperparameters through grid search. Set the time step of TDNN to 3, initialize the forget gate bias to 1.0, set Dropout to 0.2, set L2 regularization to 0.01, set the initial learning rate to 0.001, set the batch size to 32, and set the training epochs to 100. Choose the Adam optimizer, set the initial learning rate to 0.001, and decay by 0.1 times every 10 epochs. If the performance of the validation set does not improve for 10 consecutive epochs, stop training and use 5-fold cross validation to ensure model stability and reliability. Compare the complete TSTL model with TDNN-LSTM, TextRank-LP-TDNN-LSTM, TS-TDNN, and TS-LSTM. The results of the ablation experiment are shown in Table 4.

Table 4: Results of ablation experiment

Model	Accuracy/%	F1/%	Cohen's Kappa
TDNN-LSTM	90.22	89.82	84.63
TextRank-LP-TDNN-LSTM	93.46	92.57	86.59
TS-TDNN	87.49	86.04	84.25
TS-LSTM	89.24	88.62	84.15
TSTL	95.18	94.61	90.45

From Table 4, it can be seen that the indicators of the complete TSTL model perform the best, with an accuracy of 95.18%. This indicates that the proposed improvement strategies effectively enhance the model's public opinion prediction performance. Use TF-IDF, Bi-LSTM, and SVM as comparison models. To verify the TSTL model's ability to recognize network public opinion risk states, 200 sets of public opinion information from the 2019 NBA period were extracted. This included 50 sets each from the incubation period, outbreak period, maturation period, and decline period, as shown in Figure 10.



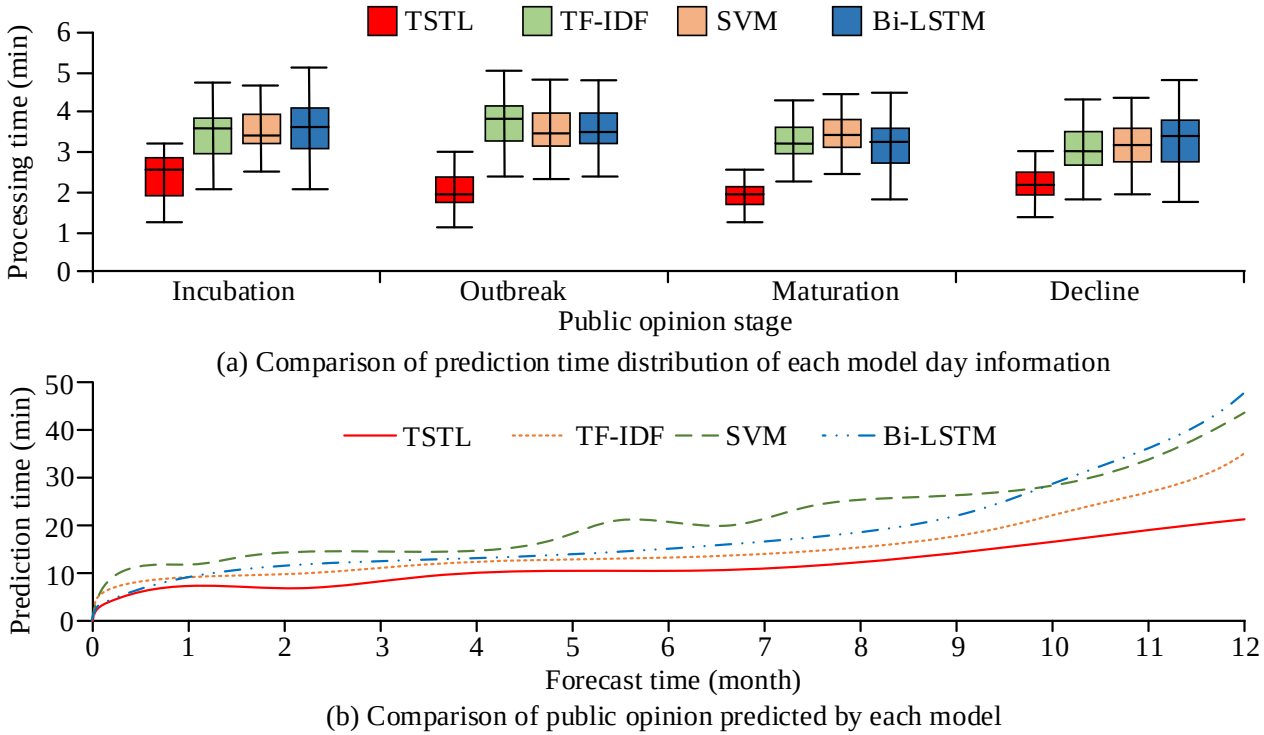
(a) Confusion matrix of recognition results

(b) Comparison of identification deviation rate at each stage

Figure 10: Confusion matrix and relative deviation rate of recognition results

As shown in Figure 10(a), in the data from the four different stages, the TSTL model had a recognition accuracy of 100% for the decline period data, 80% for the maturation period, 98% for the outbreak period, and 92% for the incubation period. The average accuracy was 92.5%. As shown in Figure 10(b), the maximum deviation for the maturation period data was 55.2%, with an average of 21.0%, both higher than the other stages. In

contrast, the maximum deviation for the decline period was only 3.2%, with an average of 1.5%, both lower than the other stages. This indicates high model accuracy in identifying public opinion stages, with highest recognition accuracy during the decline period. The TSTL model runs on GPU and utilizes 24 threads to accelerate data preprocessing and model inference. The Bi LSTM model runs on GPU and is single-threaded. The TF-IDF model and SVM model run on the CPU, single-threaded. The average GPU utilization rates of the TSTL model and Bi LSTM model are 85% and 70%, respectively. The average CPU utilization rates of TF-IDF and SVM models are 60% and 55%, respectively. To verify the work efficiency of the TSTL model, the processing time for 200 sets of data was compared across models, and the time spent for predicting future public opinion states was also compared. The results are shown in Figure 11.



(a) Comparison of prediction time distribution of each model day information

(b) Comparison of public opinion predicted by each model

Figure 11: Comparison of processing time and prediction time

As shown in Figure 11(a), regardless of the public opinion stage, the TSTL model's processing time was generally lower than the comparison models. The time distribution for the outbreak, maturation, and decline periods had lower dispersion compared to the comparison models. As shown in Figure 11(b), under different future prediction durations, the TSTL model required less time than the comparison models. For instance, when predicting for 12 months, the TSTL model took 21 minutes, the TF-IDF model took 32 minutes, the SVM model took 43 minutes, and the Bi-LSTM model took 47 minutes. This indicates that the TSTL model was faster in both current data processing and future public opinion prediction. Finally, to verify the accuracy of the model in actual applications, the predicted changes in attention from each model were compared with the actual changes in attention. The results are shown in Figure 12.

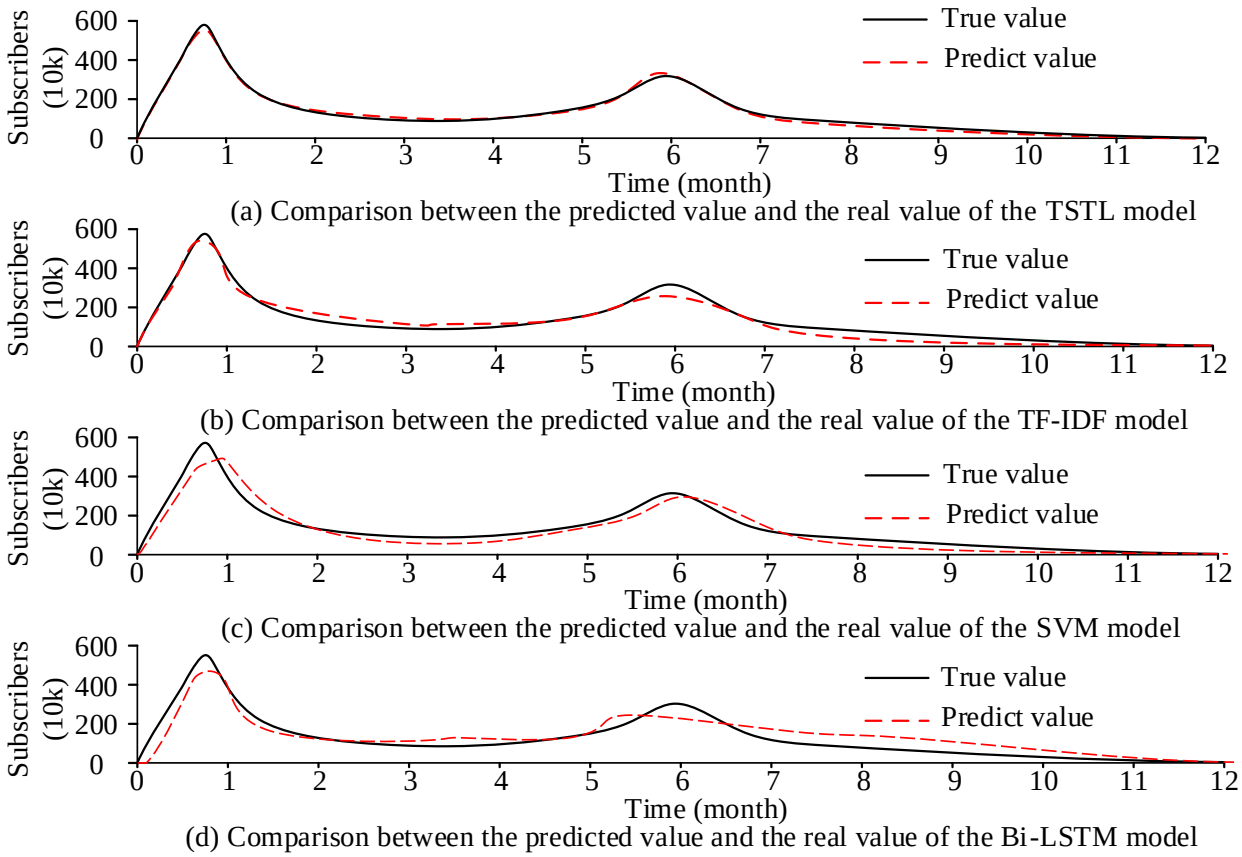


Figure 12: Comparison of predicted attention and actual attention

As shown in Figure 12(a), the predicted attention of the TSTL model was closely aligned with the ground truth, with only slight deviations at 1 month and 6 months after the public opinion event occurred. As shown in Figure 12(b), the TF-IDF model's prediction had significant deviations at 1 month and 6 months, with slight deviations at months 2-4 and 7-9. As shown in Figures 12(c) and 12(d), the predictions of the SVM and Bi-LSTM models had large deviations from the actual results, with the first predicted outbreak occurring 0.1-0.2 months later than the actual event, and the predicted attention scale was clearly smaller than the actual scale. In conclusion, the TSTL model had superior prediction capability for future public opinion compared to the TF-IDF, SVM, and Bi-LSTM models. To further validate the proposed model's superiority and generalization, this study used ICE-2024 international event data for testing, including 8426 data points. Compare the proposed TSTL model with the currently advanced Robust Optimized BERT Approach (RoBERTa), Multi Task Learning Deep Feedforward Sequential Memory Network (MTL-DFMN), and Modern Temporary Convolutional Network (ModernTCN). Accuracy, mean square error (MSE), and mean absolute error (MAE) are used as evaluation metrics. Repeat the experiment 10 times to obtain stable statistical results, and use paired t-test to evaluate whether the performance differences between models are statistically significant. Set the significance level α to 0.05, and if $p<0.05$, it is considered significant. The performance comparison results of the four models are shown in Table 5.

Table 5: Performance comparison of four models

Model	Accuracy/%	MSE	MAE	p (vs. ModernTCN)
RoBERTa	93.052±1.761	0.028±0.006	0.156±0.075	<0.05
MTL-DFSMN	91.680±2.042	0.032±0.011	0.178±0.082	<0.05
ModernTCN	92.638±1.850	0.029±0.008	0.163±0.077	<0.05
TSTL	95.125±1.043	0.021±0.005	0.131±0.062	-

From Table 5, it can be seen that the proposed TSTL model still demonstrates good performance in the larger ICE-2024 international event data, with an average accuracy of 95.125%, MSE of 0.021, and MAE of 0.131, which is significantly better than the comparison model ($p<0.05$). The results indicate that the proposed TSTL

model demonstrates superior performance and good generalization in public opinion analysis tasks.

5 Discussion

To address low efficiency and poor predictive ability in current keyword extraction and online public opinion evolution research methods, the study proposed a network public opinion evolution research model based on TextRank-LP. This model uses an optimized TextRank-LP for keyword extraction and integrates TDNN and LSTM networks as the data analysis module to perform time-series feature analysis of online public opinion. In the process of keyword extraction, the SC algorithm classifies the text based on the text feature vector, reducing the weight propagation bias in the keyword extraction process, and the classified text has clearer information structure, improving keyword extraction speed. In addition, by using features such as sentence length as classification criteria, texts within the same cluster have similarity in sentence structure, further improving the accuracy of TextRank-LP in weight assignment. The experimental results show that the accuracy of TF-IDF, SVM, and Bi-LSTM algorithms is 92.1%, 88.1%, and 86.0%, respectively. The accuracy of the TS keyword extraction algorithm proposed in the study can reach 95.1%, which is higher than the comparison algorithm, and it begins to converge at an iteration completion rate of 40%. When the text length is 1200 words, the processing time of the TS algorithm is 20 seconds, which is 9 seconds shorter than the TF-IDF algorithm. This verifies the positive impact of SC pre-classification on keyword extraction. In practical testing of the constructed TSTL model, the recognition accuracy for public opinion stages was 92.5%. The model demonstrated superior processing efficiency. In addition, when analyzing the temporal variation relationship of keywords, TDNN is mainly used to extract local time series features and capture short-term dependencies through convolution operations. LSTM networks have unique memory units and gating mechanisms, which can effectively handle dependencies over long time spans. In the TSTL model, the combination of TDNN and LSTM enables the model to grasp both the local variation characteristics of keywords and the global temporal relationships. The experimental results show that the TSTL model can more accurately predict the future development of public opinion in public opinion prediction experiments, and can accurately predict network public opinion attention 12 months ahead within 21 minutes. This is due to the strong sensitivity of LSTM to sequences, which can better grasp the temporal changes of keywords.

Data bias may have a significant impact on the performance and results of the model. If there are deviations in the data collection process, the model may not be able to fully reflect the actual situation of the entire public opinion, resulting in deviations in predicting the development of public opinion and identifying key features. The complexity and diversity of language can also lead to semantic biases in data. Therefore, in practical applications, continuous monitoring and evaluation of data quality are necessary to ensure the reliability and effectiveness of the model. The TextRank-LP-based network public opinion evolution model has good scalability and can handle high-capacity multi-platform text streams. In practical applications, distributed computing and data preprocessing techniques can meet high-capacity multi-platform text stream processing requirements. Meanwhile, by optimizing the architecture of the model, such as adopting more efficient neural network structures and optimization algorithms, the running speed and efficiency of the model can also be improved, enabling it to better adapt to complex network environments. In addition, the model is constructed based on TextRank-LP and TDNN-LSTM networks, and has strong generalization ability, thus exhibiting certain advantages in transferability across event types. Specifically, the TextRank-LP algorithm does not rely on domain specific knowledge and mainly focuses on the structure of the text and the importance of words when extracting keywords, making it applicable to different types of events. The TDNN-LSTM network can adapt to changes in different time series characteristics, thus accurately grasping the evolution law of public opinion in different types of events. Although there are differences in the data characteristics of different event types, the model can adapt to these differences by adjusting parameters and training processes.

6 Conclusion

In conclusion, the TextRank-LP and TDNN-LSTM network-based model for keyword extraction and evolution of public opinion in emergency events demonstrated high accuracy and efficiency, meeting the accuracy and efficiency requirements for emergency event public opinion research. The proposed model has broad prospects in practical applications, not only for monitoring public opinion, but also for promoting business development. For example, when monitoring social media public opinion during the release of a brand’s new product, the model can quickly extract key information, thereby helping the enterprise to understand the stage of public opinion in real time and adjust market strategies in a timely manner. However, the actual network environment is much more complex than the laboratory, and the attention-based linear modeling method used in this study may not fully capture subtle nonlinear features of public opinion changes when dealing with complex nonlinear public opinion evolution laws. In addition, the current model only uses sentence length as a text classification criterion, which may be limited when dealing with multilingual and text

containing a large number of informal expressions and slang. Therefore, in future research, cyclic attention mechanisms can be further introduced to better handle complex nonlinear features, such as Transformer models. Additionally, multilingual pre-trained models can be used for initialization to effectively capture common features of different languages, such as mBERT.

Funding

This paper is one of the research outcomes of the Jiangxi Province Higher Education Humanities and Social Sciences Research Project titled "Research on the Practice and Innovation of Network Education Work in Higher Vocational Colleges from the Perspective of Matrix Communication", with the project number: MKS23212.

References

[1] Bergquist M, Nilsson A, Harring N, Jagers S C. Meta-analyses of fifteen determinants of public opinion about climate change taxes and laws. Nature Climate Change, 2022, 12(3): 235-240. https://doi.org/10.1038/s41558-022-01297-6

[2] Zhu K. Detection of negative online public opinion among college students based on STOA optimization algorithm and dissemination trend data. Informatica, 2024, 48(17): 153-170. https://doi.org/10.31449/inf.v48i17.6385

[3] Baek C, Kang J, Choi S S. Online unstructured data analysis models with KoBERT and Word2vec: a study on sentiment analysis of public opinion in Korean. International Journal of Fuzzy Logic and Intelligent Systems, 2023, 23(3): 244-258.

[4] Munthe M. Perancangan Aplikasi Silogisme Artikel Bahasa Batak Dengan Menerapkan Metode Textrank. Journal of Computing and Informatics Research, 2022, 1(2): 34-37.

[5] Wang L, Zhang Y, Yuan J, Hu K, Cao S. FEBDNN: fusion embedding-based deep neural network for user retweeting behavior prediction on social networks. Neural Computing and Applications, 2022, 34(16): 13219-13235. https://doi.org/10.1007/s00521-022-07174-9

[6] Gusra D T, Tan K M, Changniago R. Peringkas Berita Otomatis Berbasis Website dengan Metode Text Rank. Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika, 2024, 6(2): 139-144. https://doi.org/10.53580/naratif.v6i2.300

[7] Sihombing J J, Arnita A, Al Idrus S I, Niska D Y. Implementation of text summarization on indonesian scientific articles using textrank algorithm with TF-IDF web-based. Journal of Soft Computing Exploration, 2024, 5(3): 310-319. https://doi.org/10.52465/joscex.v5i3.475

[8] Benhafid Z, Selouani S A, Amrouche A, Sidi Yakoub M. Attention-based factorized TDNN for a noise-robust and spoof-aware speaker verification system. International Journal of Speech Technology, 2023, 26(4): 881-894. https://doi.org/10.1007/s10772-023-10059-4

[9] Qiao L, Gao Y, Xiao B, Bi X, Li W, Gao X. HS-Vectors: Heart sound embeddings for abnormal heart sound detection based on time-compressed and frequency-expanded TDNN with dynamic mask encoder. IEEE Journal of Biomedical and Health Informatics, 2022, 27(3): 1364-1374. https://doi.org/10.1109/JBHI.2022.3227585

[10] Mohbey K K, Meena G, Kumar S, Lokesh K. A CNN-LSTM-based hybrid deep learning approach for sentiment analysis on Monkeypox tweets. New Generation Computing, 2024, 42(1): 89-107. https://doi.org/10.1007/s00354-023-00227-0

[11] Lilhore U K, Simaiya S, Dalal S, Damaševičius, R. A smart waste classification model using hybrid CNN-LSTM with transfer learning for sustainable environment. Multimedia Tools and Applications, 2024, 83(10): 29505-29529. https://doi.org/10.1007/s11042-023-16677-z

[12] Wang X, Kong M, Chen J, Wang X, Pei Z. Modeling topic evolution in public opinion events: an unsupervised spatio-temporal graph attention approach. Applied Intelligence, 2024, 54(20): 9706-9722. https://doi.org/10.1007/s10489-024-05684-8

[13] Zhang Y, Su Y, Zhao G, Guo X. Research on network public opinion propagation model of major epidemics under cross-infection of double emotions. Journal of System Simulation, 2023, 35(12): 2582-2593. 10.16182/j.issn1004731x.joss.22-0815

[14] Ding X, Zhang X, Fan R, Xu Q, Hunt K, Zhuang J. Rumor recognition behavior of social media users in emergencies. Journal of Management Science and Engineering, 2022, 7(1): 36-47. https://www.nstl.gov.cn/paper_detail.html?id=c46eeba3b92e70627118e1df86faf648

[15] Yuan J, Shi J, Wang J, Liu W. Modelling network public opinion polarization based on SIR model considering dynamic network structure. Alexandria Engineering Journal, 2022, 61(6): 4557-4571. https://doi.org/10.1016/j.aej.2021.10.014

[16] Cai M, Luo H, Meng X, Cui Y, Wang W. Influence of information attributes on information dissemination in public health emergencies. Humanities and Social Sciences Communications, 2022, 9(1): 1-22. https://doi.org/10.1057/s41599-022-01278-2

[17] Hernawan Y F, Adikara P P, Wihandika R C. Peringkasan Artikel Berbahasa Indonesia Menggunakan TextRank dengan Pembobotan BM25. Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK), 2022, 9(1): 61-68. https://doi.org/10.25126/jtiik.202293765

[18] Dong Y. Particle Swarm Optimization in Gene Expression Profile Clustering. Informatica, 2024, 48(16): 77-88. https://doi.org/10.31449/inf.v48i16.6360

[19] He W, Xia D, Liu J, Ghosh U. Research on the dynamic monitoring system model of university network public opinion under the big data environment. Mobile Networks and Applications, 2022, 27(6): 2352-2363. https://doi.org/10.1007/s11036-021-01881-8

[20] Zhang Y, Gao Y, Zhao Z. Research on Operation and Anomaly Detection of Smart Power Grid Based on Information Technology Using CNN+ Bidirectional LSTM. Informatica, 2025, 49(7): 157-164. https://doi.org/10.31449/inf.v49i7.7037