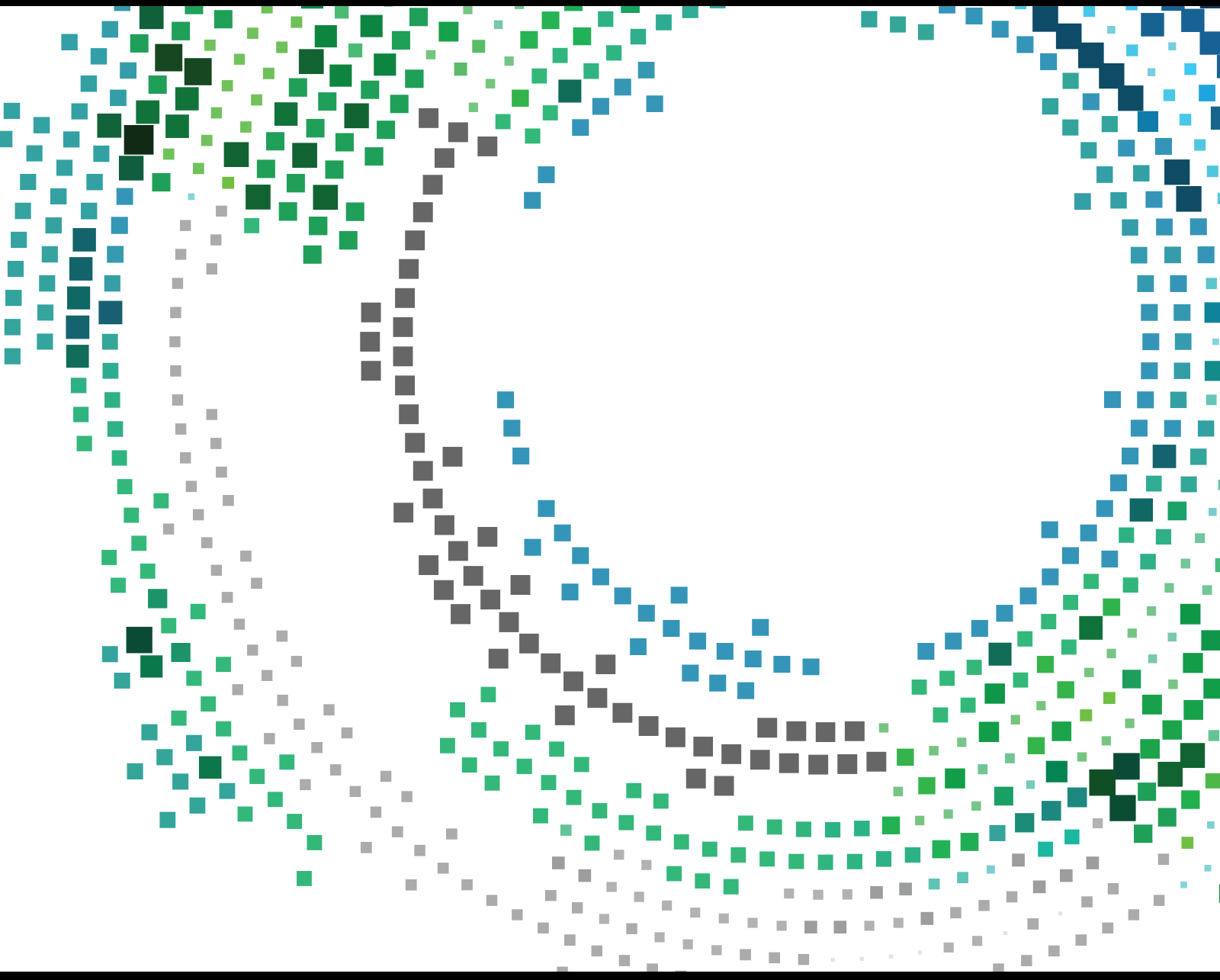
Mobile Information Systems

# Challenges of Deep Learning for Big Data Analysis in Mobile Computing Environments

Lead Guest Editor: Chin-Ling Chen
Guest Editors: Xingsi Xue and Aniello Castiglione



# Challenges of Deep Learning for Big Data Analysis in Mobile Computing Environments

Mobile Information Systems

# Challenges of Deep Learning for Big Data Analysis in Mobile Computing Environments

Lead Guest Editor: Chin-Ling Chen
Guest Editors: Xingsi Xue and Aniello Castiglione

Copyright © 2023 Hindawi Limited. All rights reserved.

This is a special issue published in “Mobile Information Systems.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

Alessandro Bazzi, Italy

## Academic Editors

Mahdi Abbasi, Iran
Abdullah Alamoodi, Malaysia
Markos Anastassopoulos, United Kingdom
Marco Anisetti, Italy
Claudio Agostino Ardagna, Italy
Ashish Bagwari, India
Dr. Robin Singh Bhadoria, India
Nicola Bicocchi, Italy
Peter Brida, Slovakia
Puttamadappa C., India
Carlos Calafate, Spain
Pengyun Chen, China
Yuh-Shyan Chen, Taiwan
Wenchi Cheng, China
Gabriele Civitarese, Italy
Massimo Condoluci, Sweden
Rajesh Kumar Dhanaraj, India
Rajesh Kumar Dhanaraj, India
Almudena Díaz Zayas, Spain
Filippo Gandino, Italy
Jorge Garcia Duque, Spain
Francesco Gringoli, Italy
Wei Jia, China
Adrian Kliks, Poland
Adarsh Kumar, India
Dongming Li, China
Juraj Machaj, Slovakia
Mirco Marchetti, Italy
Elio Masciari, Italy
Zahid Mehmood, Pakistan
Eduardo Mena, Spain
Massimo Merro, Italy
Aniello Minutolo, Italy
Jose F. Monserrat, Spain
Raul Montoliu, Spain
Mario Muñoz-Organero, Spain
Francesco Palmieri, Italy
Marco Picone, Italy
Alessandro Sebastian Podda, Italy
Maheswar Rajagopal, India
Amon Rapp, Italy
Filippo Sciarrone, Italy
Floriano Scioscia, Italy
Mohammed Shuaib, Malaysia
Michael Vassilakopoulos, Greece
Ding Xu, China
Laurence T. Yang, Canada
Kuo-Hui Yeh, Taiwan

# Contents

**Social Network Text Sentiment Analysis Method Based on CNN-BiGRU in Big Data Environment**
Youcheng Shan
Research Article (8 pages), Article ID 8920094, Volume 2023 (2023)

Hindawi
Mobile Information Systems
Volume 2023, Article ID 8920094, 8 pages
https://doi.org/10.1155/2023/8920094

*Research Article*

# Social Network Text Sentiment Analysis Method Based on CNN-BiGRU in Big Data Environment

**Youcheng Shan**

*School of Logistics and Supply Chain Management, Zhejiang Technical Institute of Economics, Hangzhou 310018, Zhejiang, China*

Correspondence should be addressed to Youcheng Shan; 260046@zjtie.edu.cn

Received 23 September 2022; Revised 26 October 2022; Accepted 24 November 2022; Published 31 January 2023

Academic Editor: Chin-Ling Chen

Copyright © 2023 Youcheng Shan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the serious colloquialism of social network texts and the sparse semantic features, this article proposes a CNN-BiGRU-based sentiment analysis method for social network texts in the big data environment. First, the dependency syntax tree is introduced to represent the dependency relationship between words to construct the word vector to represent the text. Then, sentiment features with different granularity are extracted by multiple convolution kernels of different sizes in a convolution neural network (CNN). These sentiment features are input into bidirectional gated recurrent unit (BiGRU) network for analysis to obtain deeper sentiment features. Finally, a certain number of neurons are discarded by the Dropout method, and sentiment types are classified by the Sigmoid activation function. The Weibo_senti_100k Weibo data set is used to demonstrate the proposed method. The results show that if the Dropout value is set to 0.25 and the Adam optimizer is selected, the analysis performance is the best. The accuracy, precision, recall, and AUC are about 94.09%, 95.13%, 92.87%, and 0.953, respectively, which has certain application value.

## 1. Introduction

With the rapid development of information technology and the Internet, people's daily life is also changing. Internet users can more freely publish information and express opinions on social networking platforms, which also provides convenience for the dissemination of public opinion [1]. Weibo and other mainstream social network media platforms are playing an increasingly important role in public opinion communication. In recent years, the related research on text information from social networks has attracted more and more attention, especially the sentiment analysis of the text. It is of great value to grasp the attitudes and views of netizens on the topic [2].

The hot topics on the Internet are closely related to the public sentiment in the real society. The topics pushed by the Internet in real time and the infinity of Internet users' comments and forwarding make the speed of public opinion propagation very fast. Therefore, for a topic, Internet users are easy to generate polarization of groups and even lead to group events on the Internet or in real life [3, 4]. The comment sentiment expressed by Internet users will not only affect the spread of public opinion on the whole topic but the users' sentiments will be affected by other users. Negative public opinion will drive the negative sentiments of users and promote the development of public opinion in a negative direction. Therefore, it is necessary to analyze the sentiment of social network text, that is, extract effective information from the massive social network data, and analyze the sentiment tendency of the public. The sentiment analysis results of netizens can guide and control the topic in a targeted way to curb the continuous fermentation of negative events [5]. How to mine meaningful information from the massive social network data, analyze the comment content, and mine the sentiment tendency of the comment text, have become the current research hotspots.

The analysis of sentiment tendency in the text can also be called opinion mining. It can be continuously trained through a large number of real comment data through machine learning or deep learning methods. The final model

can automatically and accurately judge the sentiment polarity expressed by the text [6]. At present, the deep learning method is excellent in sentiment classification. In essence, the deep neural network can effectively capture the high-level expression of data, and its expression ability is exponentially stronger than that of the shallow model. It can be used for sentiment analysis to more scientifically reflect the real semantics contained in the text, more carefully reflect each evaluation index of the topic, and more comprehensively help users and platforms make accurate and efficient judgments [7, 8].

However, the social network text has serious problems such as colloquialism, sparse semantic features, and too simple and refined, the analysis method based on deep learning still has some shortcomings. Therefore, a CNN-BiGRU-based social network text semantic analysis method in the big data environment is proposed. The innovations of the proposed method are summarized as follows:

(1) In view of the problems that recurrent neural networks are prone to disappear gradients and cannot grasp the nonlinear relationship over a long-time span, which leads to long-term dependence, the proposed method uses the bidirectional gated recurrent unit (BiGRU) network to obtain the hidden vector representation of text and the dependency between words with large time steps.

(2) In order to improve the accuracy of sentiment analysis, the proposed method combines the advantages of convolutional neural network (CNN) and BiGRU. CNN can extract local features and semantic information of text, and BiGRU can handle dependency information between words with large distance. Therefore, the CNN-BiGRU model is proposed, which has strong application ability.

The remaining chapters of this article are arranged as follows: the second chapter introduces the related research on emotion analysis; the third chapter introduces the emotion analysis model based on CNN-BiGRU; the fourth chapter is the experimental part, which designs a contrast experiment to verify the performance of the proposed model; and the fifth chapter is the conclusion.

## 2. Related Works

Sentiment analysis is mainly to identify a kind of positive or negative sentiment of the target object, and the goal of text sentiment analysis is to clarify the attitude of the reviewer towards the commented object [9]. Text sentiment analysis technology is to divide unstructured subjective sentences according to their different sentiment tendencies. At present, there are three kinds of methods for text sentiment analysis: Based on a sentiment dictionary, based on machine learning, and based on the neural network [10, 11]. The analysis method based on the sentiment dictionary constructs the sentiment dictionary according to the different contexts of the text and makes rules to judge the sentiment tendency. For example, Reference [12] proposed a hybrid method based on dictionary technology and fuzzy classification technology to analyze the sentiment of the Twitter text. Using the UCINET tool for social network analysis, and combining artificial neural network to rank users, the sentiment of tweet content is divided into seven categories, which effectively realizes text sentiment analysis. In order to overcome the challenges brought by nonstandard languages, Reference [13] proposed two unsupervised dictionary-based sentiment classification methods. By using a large amount of encoded information in expressions, combined with a classifier based on weakly supervised neural networks, unknown words are recognized through word embedding in specific fields, and sentiment analysis is realized on the basis of sentiment classification. However, the non-high-frequency unit classifies the sentiment of the pseudo context through network expansion, which leads to the poor generalization ability of the dictionary-based analysis method and the difficulty in constructing the sentiment dictionary.

The method based on machine learning is to train a large number of labeled text data to obtain a pretraining model, so as to predict unknown text for sentiment classification. For example, Reference [14] proposed a hierarchical fusion cross-modal complementary network for multi-modal sentiment network analysis. The feature extraction module from text and image was used to learn the attention features of text and image generated by the image-text generator to form a hierarchical fusion framework, which could fully integrate different modal features and accurately analyze the sentiment of text and image. Reference [15] proposed a method to extract relevant sentiment information from location-based social networks and used a new scale to classify the information to achieve sentiment analysis of Twitter text data. However, the traditional machine learning methods have some problems such as high dimension and sparse features.

Compared with traditional machine learning methods, the neural network performs better in natural language processing. It maps features to low dimensional word vectors with context information and solves the problems of high dimension, sparse, and no consideration of the correlation between features in traditional text representation models. Reference [16] proposed a hybrid deep learning model for fine-grained sentiment prediction in multimodal data. By combining the advantages of a deep learning network and machine learning, the two specific symbol systems, text and visual image, were used in multimodal fusion to realize context sentiment analysis. Reference [17] proposed a new text semantic recognition method based on a hybrid neural network model structure for the polysemy phenomenon and topic confusion of Weibo text. It used the latent semantic relations in different language contexts and the cooccurrence statistical features between words in Weibo, and feed the output of CNN to the LSTM filter to achieve accurate Weibo sentiment analysis. Reference [18] proposed a sentiment analysis method using CNN and a bidirectional encoder to analyze the relevant corpus collected from Twitter using a three-layer convolution framework. It conducted experiments on the corpus of 17000 tweets. The results demonstrate that it has a high accuracy.

However, due to the increasing traffic of social networks, the text information has serious problems such as colloquialism, sparse semantic features, and being too simple and refined. The ability of sentiment analysis methods based on deep learning alone to deal with such problems needs to be improved. Therefore, a CNN-BiGRU-based social network text sentiment analysis method in a big data environment is proposed.

## 3. Aspect Based Sentiment Analysis Model Using CNN-BiGRU

*3.1. Model Framework.* Gated recurrent unit (GRU) solves the problem of gradient disappearance in recurrent neural network and has the ability to learn long-term dependence. BiGRU is an improvement on GRU. It is mainly composed of two GRUs in opposite directions. In this way, not only the previous information can be obtained, but also the influence of the following information on the current world can be considered [19]. CNN can extract text features and semantic information by convolution operation a using convolution kernel. The effect of a single CNN or BiGRU model in dealing with text sentiment analysis is not very ideal. Considering the advantages of CNN and BiGRU network models in sentiment analysis tasks, the CNN-BiGRU model is constructed by combining the two neural networks to solve the problems in text sentiment analysis of social networks. The model structure is shown in Figure 1.

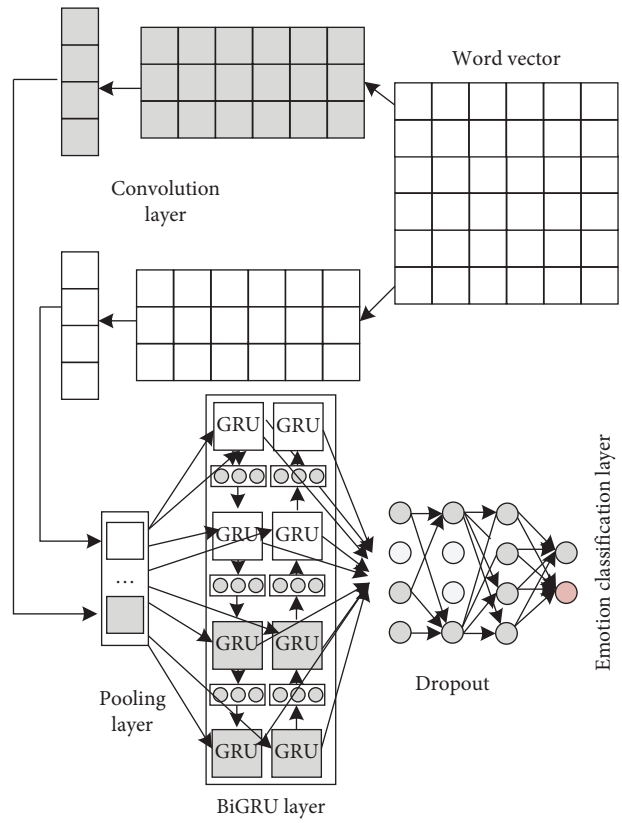



Figure 1: The structure of the CNN-BiGRU model.

*3.2. Word Vector Represents Text.* The word vector embedding layer includes sentence embedding and aspect word embedding. First, given a sentence $s = [w_1, w_2, \cdots, w_i, \cdots, w_n]$, where $n$ is the length of the words in the text, and $q = [w_i, w_{i+1}, \cdots, w_{i+m-1}]$ is the aspect word, where $m$ is the length of the aspect word. The unstructured text is mapped into continuous word vector representation by the Glove model, and then sentence embedding $[u_1; u_2; \cdots; u_n] \in R^{n \times d}$ and aspect word embedding $[u_i; u_{i+1}; \cdots; u_{i+m-1}] \in R^{m \times d}$ are obtained by looking up the embedding matrix $M^{u \times d}$, where the size of the vocabulary is represented by $u$, and the embedding dimension is represented by $d$.

When an aspect word appears in a sentence, the word closer to the aspect word will have a greater impact on the sentiment polarity of the aspect word [20]. The proposed method introduces a dependency syntax tree to represent the dependency between words. For example, in the sentence "*This food tastes good but the price is a bit expensive*," the target word is *price*, and the open source library spacey is used to build the dependency syntax tree. Each word is taken as a node, and Floyd is used to calculate the shortest distance from each node to the first target node, which is taken as the position index. The shortest distance index sequence is expressed as $P = [4, 3, 2, 3, 3, 0, 1, 4, 3, 2]$, and its corresponding position embedding is obtained by searching the position embedding matrix $D \in R^{d \times n}$, which is randomly initialized and updated during the training process.

*3.3. CNN Layer.* CNN layer mainly includes the convolution layer and pooling layer.

*3.3.1. Convolution Layer.* Convolution layer is mainly used to extract the features of the text and obtain the local semantic information of the sentence. The $r \times k$ filter used in the convolution layer performs convolution on the sentence matrix to extract local features $f_i$, and the calculation is as follows:

$$f_i = \delta(\kappa \cdot x_{i:i+c-1} + b), \tag{1}$$

where $x_{i:i+r-1}$ is the vector with total $c$ lines from $i$ to $i + r - 1$ in the sentence matrix, $\kappa$ is the convolution kernel, $b$ is the offset, and $\delta$ is the rectified linear unit (ReLU).

Each convolution kernel will extract a part of the features, and the filter will slide from top to bottom in the whole sentence matrix according to the set step size to obtain the local feature set $F$. The calculation of $F$ is as follows:

$$F = \{f_1, f_2, \cdots, f_{n-c+1}\}. \tag{2}$$

*3.3.2. Pooling Layer.* The pooling layer performs feature dimension reduction [21]. The proposed method uses the max-pooling method to extract the largest feature from the local feature set obtained by the convolution layer to replace the entire local feature $\widehat{f}_i$, and the calculation is as follows:

$$\widehat{f}_i = \max\{f_1, f_2, \cdots, f_{n-c+1}\}. \quad (3)$$

*3.4. BiGRU Layer.* The text vector is composed of a word vector and a position embedding vector. BiGRU is used to obtain aspect words and context information, so as to obtain a hidden layer vector representation [22]. GRU performance is similar to long and short-term memory networks, but it has fewer parameters and lower computational complexity. The GRU network has two gate structures of update gate $z_t$ and reset gate $r_t$. $z_t$ is used to indicate that the cell unit receives information in the previous time step. The larger the value, the more information in the previous time step is memorized. $r_t$ is used to indicate the degree of ignoring the information in the previous time step. The more the value, the more the information is forgotten. At a certain time, the hidden state of GRU is calculated as follows:

$$\begin{aligned} r_t &= \sigma(\omega_t x_t + \varpi_t h_{t-1}), \\ z_t &= \sigma(\omega_z x_t + \varpi_z h_{t-1}), \\ h_t &= \tanh(\omega x_t + \varpi(r_t \odot h_{t-1})), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot h_t, \end{aligned} \quad (4)$$

where $h_{t-1}$ represents the hidden state at a previous time, $x_t$ represents input sequence information, $\omega$ and $\varpi$ are weight matrices, and $\sigma$ is the sigmoid function.

The output $h = [\overrightarrow{h}, \overleftarrow{h}]$ of the BiGRU encoder combines the forward hidden layer $\overrightarrow{h} = [\overrightarrow{h}_1, \overrightarrow{h}_2, \cdots, \overrightarrow{h}_n]$ and the backward hidden layer $\overleftarrow{h} = [h_1, h_2, \cdots, h_n]$, where $n$ is the length of the sentence.

*3.5. Dropout Layer.* CNN-BiGRU model has too many parameters and insufficient training samples, which is prone to overfitting. Therefore, the Dropout method is adopted to randomly discard a certain number of neurons with a certain probability in the training process. Only a part of neurons are allowed to participate in the training and parameter learning of the model, so as to ensure that the model cannot rely on some local features excessively in the process of training and learning parameters. It can improve the training efficiency and generalization ability of the model [23, 24]. The neural network model using the Dropout method is shown in Figure 2.

The specific workflow of the Dropout method is as follows:

(1) Temporarily delete hidden layer neurons with a certain probability from the network.

(2) The input is propagated forward in the network and then back through the same network. After each batch size is executed, the corresponding weights and offsets are updated according to the stochastic gradient descent algorithm.

(3) Recovery of discarded neurons; at this time, the discarded neuron parameters remain the same, and the neuron parameters that have not been discarded have been updated.

(4) Repeat steps (1), (2), and (3) continuously.

The $h$ obtained from the BiGRU layer is processed by Dropout to obtain $\widetilde{h}$, and the mathematical expression is as follows:

$$\widetilde{h} = \text{Dropout}(h). \quad (5)$$

*3.6. Sentiment Classification Layer.* After the Dropout layer, the output representation is forwarded to the fully connected layer through the activation function. This layer maps the output of the previous layer to the required output size. This layer also learns to retain the relevant information required for the sentiment prediction of the target and forgets the irrelevant data. If the polarity probability of an aspect exceeds a threshold, the aspect is assigned to the corresponding sentiment class [25, 26]. The last task of fine-grained sentiment analysis is to classify the sentiment polarity. The model accepts $y_i$ as input features. The final predicted sentiment polarity of the aspect target is the label with the highest probability.

For sentiment classification, the proposed method uses a sigmoid activation function, which can output values between 0 and 1. The calculation is as follows:

$$\varphi = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)). \quad (6)$$

At the same time, the proposed model is trained with L2 regularization to minimize the cross-entropy loss.

## 4. Experiments and Analysis

The experiments are conducted based on the deep learning framework TensorFlow. The specific experimental environment is shown in Table 1.

In addition, the model parameter settings in the experiment are shown in Table 2.

*4.1. Data Set.* In order to verify the effectiveness of the proposed model, the Weibo_senti_100k Weibo data set of the open-source Chinese natural language processing data set on the ChineseNLPCorpus project on GitHub was selected for the experiment. Weibo_senti_100k belongs to Sina Weibo comment text with sentiment annotation, with about 50000 positive and negative comments, respectively.

Weibo_senti_100k Weibo data set is preprocessed to remove various punctuation marks, HTML tags, "#" tags, "@" in user names, and so on. Then, use the Jieba word segmentation tool to segment words, and remove the stop words according to the stop words list of the Harbin Institute of Technology. Finally, in order to train and test the text sentiment analysis model, the preprocessed data set is divided into 80000 and 20000 items, respectively according to the ratio of the test set and a training set of 1 : 4. The sentiment categories are divided into positive and negative categories.

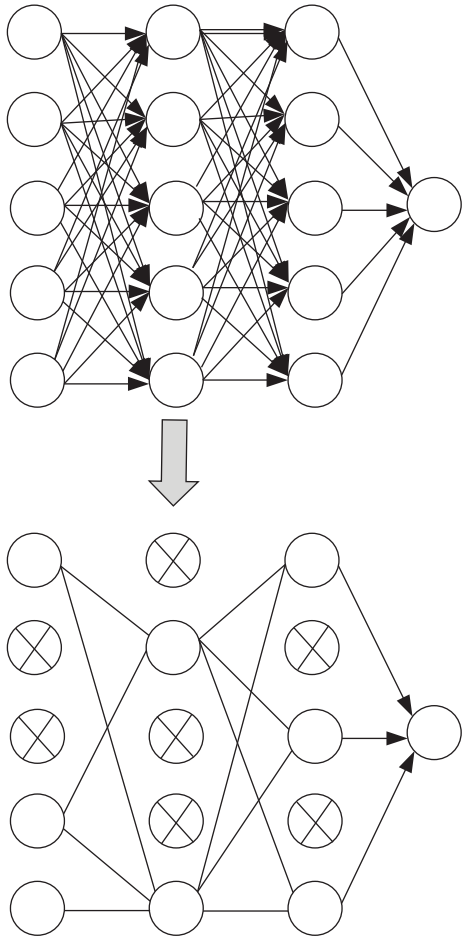


Figure 2: Neural network model using dropout.

Table 1: Experimental operation environment.

| Experimental environment | Environment configuration |
|---|---|
| Operating system | Ubuntu 18.04 |
| Memory size | 32G |
| Deep learning framework | TensorFlow 1.12.0 |
| Programming language | Python 3.7 |

Table 2: Parameter setting of the model.

| Parameter | Value |
|---|---|
| Word vector dimension | 768 |
| BiGRU neuron | 256 |
| Learning rate | 0.001 |
| Batch size | 96 |
| Number of iterations | 100 |

*4.2. Evaluation Index.* In the experiment, the accuracy, the area under ROC (receiver operating characteristic) curve (AUC), the precision, the recall, and $F1$ score are used as the evaluation indexes of the proposed model. Let TP be the true positive case, TN be the true negative case, FP be the false positive case, and FN be the false negative case.

(1) Accuracy: accuracy is an index used to evaluate the classification model. Generally speaking, the accuracy refers to the proportion of correct results predicted by the proposed model, which is calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{7}$$

(2) AUC (area under curve): the area enclosed by the coordinate axis under the ROC curve. The ROC curve shall be calculated before calculating AUC. The ROC curve is to draw the corresponding true positive rate (TPR) and false positive rate (FPR) results of the vertical axis with different cut-off points in the two-dimensional coordinate system. The obtained curve is the ROC curve. TPR and FPR are calculated as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{TN} + \text{FP}}. \end{aligned} \tag{8}$$

(3) Precision: the precision index is the proportion of samples identified as positive categories that are indeed positive categories. Precision is defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{9}$$

(4) Recall: recall refers to the proportion of all positive category samples correctly identified as a positive category. The definition of recall is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{10}$$

(5) $F1$ score: in order to give consideration to the precision and the recall, neither the precision nor the recall is preferred, so the derivative $F1$ score of the harmonic average of the two can be used. An important characteristic of the harmonic mean: only when both are large or small, the harmonic mean will be large or small, and as long as one is small, the result will be greatly lowered. $F1$ score is defined as follows:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

*4.3. Selection of Dropout Value.* There are too many parameters in CNN-BiGRU model, and it is easy to produce an overfitting phenomenon in the training process. In order to solve this problem, the Dropout method is used to randomly discard the number of neurons with specific probability values in the training process. Only the remaining neurons are allowed to participate in the training and parameter learning of the model. During the training process, the model

cannot rely excessively on some local feature information, so as to improve the training efficiency and generalization ability of the model. In the experiment, the Dropout values are set to 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, and 0.5 for analysis and verification. It explores the impact of different Dropout values on the sentiment analysis results. The specific experimental results are shown in Table 3.

According to the experimental results in Table 3, when the Dropout value is 0.1, the overall performance of the model is relatively poor, and the accuracy is 89.24%. This may be because the Dropout value is too small, the number of randomly discarded neurons is small, and there are too many parameters in the model. When the Dropout value is increased to 0.25, the model performance reaches the best from the three evaluation indexes of accuracy, precision, and $F1$, which are 89.95%, 90.07%, and 90.54%, respectively. After the overall performance of the model reaches the optimum, continue to increase the Dropout value. Almost all the evaluation indexes begin to show a downward trend, and the performance of the model begins to decline gradually. This may be because the number of discarded neurons is too large, resulting in too few parameters. It is difficult for the model to analyze the data well, resulting in the overall performance degradation. Therefore, in general, the Dropout value is set to 0.25.

*4.4. Selection of Optimizer.* The essence of the model training process is to minimize the loss function. After determining the loss function, the optimizer needs to be used for gradient optimization. The optimization target is the parameters in the model. The parameters in the training CNN-BiGRU model include all the parameters in CNN and BiGRU. In the experiment, five commonly used optimizers including SGD (stochastic gradient descent), AdaGrad, Adadelta, RMSProp (root mean square prop), and Adam are selected to optimize the model. Explore the impact of different optimizers on the model performance. The specific experimental results are shown in Table 4.

According to the experimental results in Table 4, the Adam optimizer has the best effect, with accuracy, precision, and $F1$ values of 88.98%, 88.86%, and 90.27%, respectively. This is mainly because Adam can adaptively adjust the learning rate parameters, and combines the advantages of Adadelta, RMSProp, and Momentum, and Adam is quite robust to the selection of super parameters. Therefore, the optimizer selects Adam.

*4.5. Comparison with Other Methods.* According to the model trained on the training set, the test set is analyzed. In order to demonstrate the analytical performance of the proposed method on the test set, it is compared with the methods in References [12, 17]. The results are shown in Figure 3.

It can be seen from Figure 3 that the proposed method performs well in any evaluation index compared with the other two methods, and its accuracy, precision, recall, and AUC are about 94.09%, 95.13%, 92.87%, and 0.953, respectively. Since the BiGRU model adopted in the proposed method can better learn the context information in the social network text and is more conducive to learning the text sentiment, its accuracy, precision, and AUC are improved by about 2.16%, 2.65%, and 0.081 compared with the LSTM model in Reference [17]. In addition, the combination of CNN and LSTM models in Reference [17] can better obtain the text sentiment types. Compared with the use of artificial neural networks only in Reference [12], its performance has been significantly improved, and the AUC has increased by about 0.157. Overall, the proposed method has the best performance in text sentiment analysis, which also verifies the effectiveness of the CNN-BiGRU model.

Table 3: Experimental results of different dropout values.

| Dropout | Accuracy (%) | Precision (%) | Recall (%) | $F1$ (%) |
|---|---|---|---|---|
| 0.10 | 89.24 | 87.53 | 90.77 | 89.12 |
| 0.15 | 89.56 | 87.12 | 91.04 | 89.04 |
| 0.20 | 89.74 | 89.71 | 90.87 | 90.29 |
| 0.25 | **89.95** | **90.07** | 91.01 | **90.54** |
| 0.30 | 89.52 | 87.65 | 90.53 | 89.07 |
| 0.35 | 89.05 | 88.06 | 91.88 | 89.93 |
| 0.40 | 88.78 | 86.92 | **92.39** | 89.57 |
| 0.45 | 88.59 | 89.88 | 88.20 | 89.03 |
| 0.50 | 88.33 | 87.75 | 89.89 | 88.81 |

Bold marks the highest value.

Table 4: Experimental results of different optimizers.

| Optimizers | Accuracy (%) | Precision (%) | Recall (%) | $F1$ (%) |
|---|---|---|---|---|
| SGD | 84.56 | 81.43 | 89.75 | 85.39 |
| AdaGrad | 87.93 | 84.74 | 89.07 | 87.86 |
| Adadelta | 88.17 | 86.85 | 89.81 | 88.30 |
| RMSProp | 88.55 | 85.22 | **92.04** | 88.49 |
| Adam | **88.98** | **88.86** | 91.72 | **90.27** |

Bold marks the highest value.

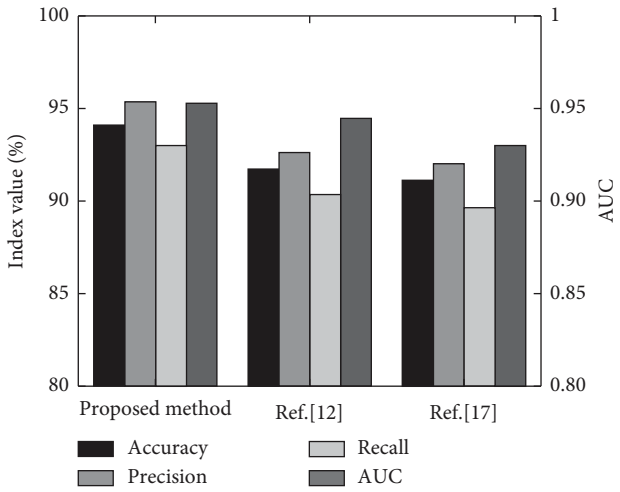



Figure 3: Text sentiment analysis results of different methods.

## 5. Conclusion

In the face of the explosive growth of text resources in social networks, how to effectively use text data and mine potential

value is of great significance. Therefore, a CNN-BiGRU-based social network text sentiment analysis method in a big data environment is proposed to accurately obtain the sentiment polarity. The dependency syntax tree is used to represent the dependency relationship between words, and it is input into the CNN-BiGRU model to learn and obtain various sentiment features. At the same time, a certain number of neurons are discarded by the Dropout method to improve the analysis efficiency. Finally, the sigmoid activation function is used to complete the classification of sentiment types. Experiments based on Weibo_senti_100k Weibo data set shows as follows:

(1) The proposed method adopts the Dropout method and Adam optimizer to improve its analysis performance. At the same time, the Dropout value is set to 0.25, and its $F1$ score reaches 90.54%. The optimization effect is remarkable.

(2) The proposed CNN-BiGRU model has good sentiment analysis performance. Its accuracy, precision, recall, and AUC are about 94.09%, 95.13%, 92.87%, and 0.953, respectively, which is superior to other comparative methods.

The text information is diversity, but the proposed method only classifies the polarity of text sentiment into positive and negative, without more fine-grained identification and deep mining of text sentiment, such as dividing into three categories (positive, neutral, and negative) or more categories such as "pleasure, anger, sorrow, and joy." Therefore, more detailed research will be conducted in the next work.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work was supported by the High Level Vocational Professional Group Construction Project of China (2019), the National Level Teaching Innovation Team of Vocational Education Teachers (2019), and the Industry University Cooperation and Collaborative Education Project of Zhejiang Province (2020).

## References

[1] N. Khan, A. Khalique, T. A. Khan, and I. Hussain, "Sentiment analysis of social network feeds for threat protection," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 7, pp. 439–445, 2020.

[2] M. M. Ibaez, R. R. Rosa, and L. Guimares, "Sentiment analysis applied to analyze society's emotion in two different context of social media data," *Inteligencia Artificial*, vol. 23, no. 66, pp. 66–84, 2020.

[3] A. Safruddin, A. Hermawan, and A. P. Wibowo, "Mplementation of backpropagation neural network in sentiment analysis on twitter to public figures," *Compiler*, vol. 9, no. 2, pp. 101–108, 2020.

[4] Y. Liu, D. Liu, and Y. Chen, "Research on sentiment tendency and evolution of public opinions in social networks of smart city," *Complexity*, vol. 2020, no. 7, Article ID 9789431, 13 pages, 2020.

[5] H. Yuan, Y. Song, J. Hu, and Y. Ma, "Design of festival sentiment classifier based on social network," *Computational Intelligence and Neuroscience*, vol. 2020, no. 2, Article ID 8824009, 10 pages, 2020.

[6] J. Zhao, D. Zeng, Y. Xiao, L. Che, and M. Wang, "User personality prediction based on topic preference and sentiment analysis using LSTM model," *Pattern Recognition Letters*, vol. 138, no. 2, pp. 397–402, 2020.

[7] Y. Madani, M. Erritali, J. Bengourram, and F. Sailhan, "Social network analysis: transforming a black and white approach into a grey approach using fuzzy logic system," *Journal of Information Technology Research*, vol. 13, no. 3, pp. 142–155, 2020.

[8] P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan, and R. Sharma, "Effect of negation in sentences on sentiment analysis and polarity detection," *Procedia Computer Science*, vol. 185, no. 1, pp. 370–379, 2021.

[9] C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "Hybrid deep learning models for sentiment analysis," *Complexity*, vol. 2021, no. 3, Article ID 9986920, 16 pages, 2021.

[10] P. Sánchez-Holgado and C. Arcila-Calderón, "Supervised sentiment analysis of science topics: developing a training set of tweets in Spanish," *Journal of Information Technology Research*, vol. 13, no. 3, pp. 80–94, 2020.

[11] Q. Yao, R. Y. M. Li, L. Song, and M. J. C. Crabbe, "Construction safety knowledge sharing on Twitter: a social network analysis," *Safety Science*, vol. 143, pp. 105411–105418, 2021.

[12] M. M. Madbouly, S. M. Darwish, and R. Essameldin, "Modified fuzzy sentiment analysis approach based on user ranking suitable for online social networks," *IET Software*, vol. 14, no. 3, pp. 300–307, 2020.

[13] K. Kobs, A. Zehe, A. Bernstetter et al., "Emote-controlled: obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular Twitch.tv channels," *ACM Transactions on Social Computing*, vol. 3, no. 2, pp. 1–34, 2020.

[14] C. Peng, C. Zhang, X. Xue, J. Gao, H. Liang, and Z. Niu, "Cross-modal complementary network with hierarchical fusion for multimodal sentiment classification," *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 664–679, 2022.

[15] S. Dass, V. K. Kannan, and K. Shyamala, "Sentiment severity on location-based social network (LBSN) data of natural disasters," *International Journal of Recent Technology and Engineering*, vol. 8, no. 5, pp. 6–12, 2020.

[16] A. Kumar, K. Srinivasan, W. H. Cheng, and A. Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Information Processing and Management*, vol. 57, no. 1, Article ID 102141, 2020.

[17] M. Ling, Q. Chen, Q. Sun, and Y. Jia, "Hybrid neural network for Sina weibo sentiment analysis," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 983–990, 2020.

[18] P. Mishra, "Sentiment analysis by novel hybrid method BE-CNN using convolutional neural network and BERT," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5320–5326, 2020.

[19] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A. M. Al-Zoubi, I. Aljarah, and H. Faris, "An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter," *Knowledge-Based Systems*, vol. 192, no. 15, Article ID 105353, 2020.

[20] D. Kandasamy, P. Dhandayudam, V. Pirya, and S. Sudhakar, "Deep learning sentiment analysis for recommendations in social applications," *International Journal of Scientific and Technology Research*, vol. 9, no. 1, pp. 3812–3815, 2020.

[21] A. A. Kharlamov, A. N. Raskhodchikov, and M. Pilgun, "Smart city data sensing during COVID-19: public reaction to accelerating digital transformation," *Sensors*, vol. 21, no. 12, pp. 3965–3969, 2021.

[22] N. Garg, "Annotated corpus creation for sentiment analysis in code-mixed Hindi-English (Hinglish) social network data," *Indian Journal of Science and Technology*, vol. 13, no. 40, pp. 4216–4224, 2020.

[23] V. Roseline and G. H. Chellam, "Sentiment classification using PS-POS embedding with bilstm-crf and attention," *International Journal of Future Generation Communication and Networking*, vol. 13, no. 4, pp. 3520–3526, 2020.

[24] M. S. Do, G. Choi, J. W. Hwang et al., "Research topics and trends of endangered species using text mining in Korea," *Journal of Asia-Pacific Business*, vol. 13, no. 4, pp. 518–523, 2020.

[25] V. Roseline and D. Herenchellam, "PS-POS embedding target extraction using CRF and BiLSTM," *International Journal of Advanced Science and Technology*, vol. 29, no. 3, pp. 10984–10995, 2020.

[26] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, no. 8, pp. 58080–58093, 2020.