World Wide Web
https://doi.org/10.1007/s11280-021-00994-0

A distributed learning based sentiment analysis methods with Web applications

Guanghao Xiong[1] · Ke Yan[1,2] · Xiaokang Zhou[3,4]

Received: 22 August 2021 / Revised: 18 November 2021 / Accepted: 21 December 2021

Abstract

The main challenge of using deep learning (DL) for sentiment analysis tasks is that insufficient data leads to a decrement in classification accuracy. In addition, privacy issues are always concerned for sentiment data analysis. To tackle the above two mentioned problems, We propose a model based on the federated learning framework (Fed_BERT_MSCNN), which contains a Bidirectional Encoder Represent-ations from Transformers (BERT) module and a multi-scale convolution layer. It uses the BERT_MSCNN model for training on the data sets of multiple companies, and employs the federated learning framework to collect the model parameters of different distributed nodes. Finally, these model parameters are transmitted to the central node. The central node performs a weighted average of all model parameters, sending a set of common model parameters to the distributed nodes. According to the experimental results, the proposed model performs better than the state-of-the-art models in terms of accuracy, F1-score, and computational efficiency. In addition, we optimize the model parameters in order to practice in distributed computing models for web applications.

Keywords Sentiment analysis · Federated learning · Pre-training model · Feature extract · Distributed computing · Web applications

This article belongs to Topical Collection: *Special Issue on Resource Management at the Edge for Future Web, Mobile, and IoT Applications*

Guest Editors: Qiang He, Fang Dong, Chenshu Wu, and Yun Yang

✉ Ke Yan
keddiyan@gmail.com

✉ Xiaokang Zhou
zhou@biwako.shiga-u.ac.jp

[1] College of Information Engineering, China Jiliang University, Hangzhou, China

[2] National University of Singapore, Singapore, 4 Architecture Drive, Singapore 117566, Singapore

[3] Faculty of Data Science, Shiga University, Hikone 5228522, Japan

[4] RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo 1030027, Japan

Published online: 13 January 2022

1 Introduction

Since the emergence of the Web2.0 era, the Internet has been slowly changed from centralized computing to distributed and collaborated computing. New distributed learning technologies, such as cloud computing, edge computing and federated learning have been developed rapidly in recent years. Internet users have gradually moved from 'reading' web pages to 'co-constructing' the Internet. For example, people nowadays often post and share personal opinions, comment on e-commerce and film review websites. DoubleClick Inc. [23] conducted surveys on the clothing industry, sports and fitness products, travel industry network customers, and computer hardware equipment industries in the United States. And found that in the above-mentioned industries, nearly half of consumers will search the Internet for the introduction of related products and other consumer reviews of products before making a purchase decision. Internet product reviews are important for consumers' decision to purchase. Merchants nowadays obtain consumer feedback information from product review abstracts for optimizing marketing strategies [39]. However, the traditional method of resource management is time-consuming and laborious, so effective computerized mechanisms are needed to process the online reviews. Sentiment analysis, as an important research topic in the field of natural language processing (NLP), aims at extracting users' attitudes towards the product from the text comments generated by users' subjective consciousness and providing more accurate emotion classification results.

Sentiment analysis was originally based on the emotional dictionary method. This method is convenient in the classification process, which requires emotional dictionary searching and recurrence of words perform analysis (WPA), the dictionary building in the early stage needs high integrity. With the fast development of artificial intelligence (AI) and machine learning (ML) technologies [57], models such as SVM, decision trees, random forests, and naive Bayes have gradually been applied to automated sentiment analysis (ASA). ML methods have a complete theoretical foundation with better performance than traditional sentiment dictionaries. Nowadays, with the development of natural language processing technology, deep learning (DL) network models based on traditional RNN and LSTM are also applied to sentiment analysis problems. However, both ML and DL methods heavily rely on the sufficiency of the training data. [2, 33].

According to our literature survey, there are three main challenges for the state-of-art methods in the field of sentiment analysis. First, the traditional method is time-consuming and labor-intensive. Secondly, insufficient data will reduce the accuracy of sentiment analysis tasks to affect the performance of the model. Last, data privacy is also a big issue when integrating multiple data sets using distributed and collaborative learning.

Distributed learning uses multiple computing nodes for deep learning, which aims to improve the ML and DL performance with a combination of training datasets at different locations. It is also a good solution for data privacy protection. Federated learning, as a particular distributed learning framework, enables deep learning models to train efficiently while ensuring data security and privacy based on distributed learning [56]. In this article, We propose a model based on the federated learning framework (Fed_BERT_MSCNN), which contains a Bidirectional Encoder Represent-ations from Transformers (BERT) module and a multi-scale convolution layer. It can effectively collect the review information of different merchants under the premise of protecting data privacy, extract the emotional polarity in the review information, and make an accurate analysis. The model is mainly composed of three parts. The first part is a federated learning framework, which is used to construct a common sentiment classification model for reviews of different companies; the second part is the pre-training model BERT which is used to convert the words that cannot be directly calculated in the input text into a vector or matrix that can be calculated, and these digitized vectors can better reflect the meaning of the corresponding words in the sentence; the third part is a feature extractor, which is used to extract features related to sentiment polarity from the text, and to recognize and classify each comment. Since the model proposed in this article processes its data in multiple computing nodes, the private data of each computing node will not be uploaded [35, 48, 50].

The method proposed in this paper has the following contributions. (1) The proposed federated learning framework makes it possible to train a high-quality centralized model while the training data is distributed on a large number of client models. With the federated learning protocol, the proposed method improves the classification accuracy and protects the privacy of client data; (2) The pre-training model BERT is introduced to convert the input text into a dynamic word vector instead of the previous static word vector [17, 19]. The context information is fully utilized in the conversion process, which can effectively solve the problem of the polysemous word [18]; (3) After the pre-training model, a convolutional neural network [58] of different scales is added as a feature extractor, which improves the expression quality of the feature space; (4) The Ranger optimizer combining RAdam and Lookahead is proposed, compared with the existing popular optimizers such as SGD, it has superior performance in terms of classification accuracy according to our experimental results.

The rest of the article is arranged as follows: Chapter 2 introduces related work; Chapter 3 introduces the method we proposed in detail; Chapter 4 shows the comparative experiment of the method; Chapter 5 summarizes this article and arranges future work.

2 Related work

The sentiment analysis task is a hot topic in the field of NLP, and many researchers have been exploring its research and application value. In sentiment analysis tasks, only convert the words or phrases of the input text into numbers, then they can be stored in the computer for numerical calculation, so word embedding is an indispensable part of the task. One-hot encoding was first proposed, but then it was discovered that its mechanism of action would bring dimensional disasters and semantic gaps. The Word2Vec model [3] proposed in 2013 has greatly promoted the development of the field of natural language processing, especially the application of deep learning. The core idea of Word2Vec is to use the neural network to train the word context to obtain the vectorized representation of the word. The training method includes CBOW and Skip-gram. But the Word2Vec model convert the words of the input text into a static word vector, which cannot solve the problem of polysemy of a word well [13]. Later, with the ElMo [41] and GPT [45] pre-training models proposed in 2018, word embedding went further and can better represent the meaning of the text. The emergence of the BERT [11] pre-training model in October 2018 is described as everything in the past is a prologue. As a breakthrough in the history of word embedding, it uses the encoder with a two-way transformer as feature extraction, and employs masked language model and next sentence prediction for training, making a significant contribution to the sentiment analysis task.

In the BERT model, the transformer encoder can only extract the global features of the sentence, the local features of the sentence also play an important role in the emotional polarity of the sentence. CNN-based deep learning methods use layers with convolution filters to extract local features and capture spatial local correlations, and have achieved fruitful results in related research in the image field [22, 26]. The emergence of the word vector model allows CNN to obtain local information in the text. In 2014 [46] first applied CNN to text sentiment classification and topic classification, and achieved good results. Wei et al. [53] used CNN to extract the text features of the source and target fields, transfer the CNN weights trained in the source field to the target field, and use a small amount of labeled target field data to fine-tune the CNN weights to achieve cross-domain text sentiment analysis. Kim et al. [24] proposed a CNN architecture with filters of different window sizes; Kalchbenner et al. [21] proposed a dynamic convolutional neural network to obtain local features of sentences.

In recent years, many deep learning models based on multi-task learning frameworks have emerged [11, 13, 20–22, 24, 26, 41, 45, 46, 53, 61, 67]. They can provide a convenient way to combine information of multiple tasks. However, large amounts of data often have higher requirements for computing resources, and this requires data from different locations of resources, resulting in privacy concerns [44]. In response to the pain points faced by the multi-task learning framework in the above sentiment analysis tasks, federated learning provides answers. Federal learning is a concept first proposed by Google Research in 2016 [25, 36], this technology can complete joint modeling without data sharing. Specifically, the own data of each data owner (individual/enterprise/institution) will not leave the local area. Through the parameter exchange method under the encryption mechanism in the federal system, a global sharing model is jointly established, and the built model only serves local targets in their respective regions [37]. Although there are some similarities between federated learning [7, 28, 47] and distributed machine learning [9, 16, 27, 30, 38], federated learning has its characteristics in terms of application fields, system design, and optimization algorithms.

We call the big data processing stage centered on the cloud computing model the era of centralized big data processing. The characteristics of this stage are mainly that the calculation and storage of big data are performed in a centralized manner in the cloud computing center [51, 60, 64] because the cloud computing center has relatively high Strong computing and storage capabilities. This resource-intensive big data processing method can save users a lot of expenses and create effective economies of scale. However, in the era of the Internet of Everything, the centralized processing model exposes its limitations due to the massive level of data generated by network edge devices. Scientists started developing new computing models to conduct in-depth research, such as microdata center [1], mobile edge computing [49], fog computing [10], Cloudlet [12], Haiyun computing of the Chinese Academy of Sciences [54], etc. The development of the application requirements of the Internet of Everything has given birth to an edge big data processing model, that is the edge computing model [50]. And the sentiment analysis model based on the federated learning framework proposed in this article is one of them. It can increase the processing capacity of performing task calculation and data analysis on the network edge device. And migrate some or all of the computing tasks of the original cloud computing model to the network edge device, reducing the computing load of the center and the pressure on the network bandwidth, increasing data processing efficiency in the era of the Internet of Everything.

3 The proposed system

We propose a model based on the federated learning framework (Fed_BERT_MSCNN), which contains a Bidirectional Encoder Represent-ations from Transformers (BERT) module and a multi-scale convolution layer. In this model, different companies use the same deep learning model to train their data and finally pass the training parameters to the central server. The central server averages all the parameters and then returns them to each company [14, 42, 59]. In the end, every company got a common deep learning model. The overall structure of the model is shown in Figure 1, which lists three examples of companies training.

From Figure 1, we can see that three different companies have used Model A, Model B, and Model C deep learning model structures. In this article, under the premise of considering accuracy and time, Model A, Model B, and Model C all use the same deep learning model. This model inputs the company's comment text to the pre-training model BERT for word embedding then passes the feature extractor MSCNN, and finally outputs the classification results through the softmax layer. The structure of the model is shown in Figure 2. The rest of this section will introduce the Fed_BERT_MSCNN model in detail.

3.1 Federal learning framework

In the previous chapter, it was mentioned that the deep learning model based on the multi-task learning framework can effectively increase the amount of data and improve the accuracy, but this method undoubtedly leads to uncontrollable data flow and the leakage of sensitive data. McMahan et al. first proposed the federated learning technology in 2016, which allows users to protect user privacy during machine learning, and to form training data sharing without the need for source data aggregation.

Federated learning is essentially a distributed machine learning technology [29], which mainly includes many clients and a central server. Clients (such as tablets, mobile phones, IoT devices) jointly train models under the coordination of a central server (such as a service provider). The client is responsible for training local data to obtain a local model, and the central server is responsible for the weighted aggregation of the local model to obtain a global model. After multiple iterations, a model that is close to the result of centralized machine learning is finally obtained, which effectively reduces many privacy risks caused by traditional machine learning source data aggregation [15].

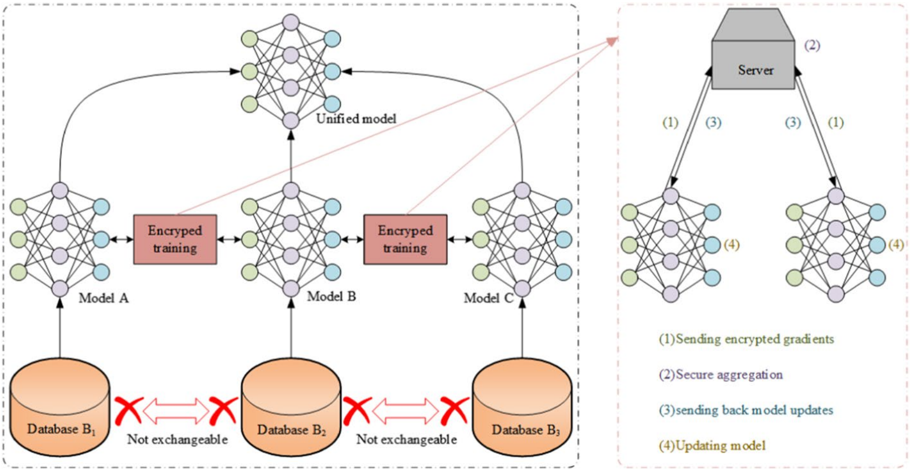

Figure 1 The overall framework of the Fed_BERT_MSCNN model

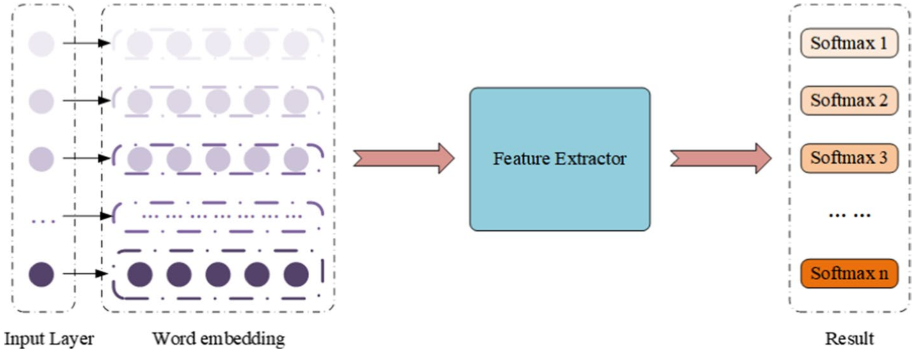

Figure 2 The overall framework of the BERT_MSCNN model

Figure 3 shows the life cycle of federated learning training and the multiple participants in the federated learning system. Specifically, the workflow is as follows. (1) The client downloads the global model from the server; (2) The client trains the local data to obtain the local model; (3) Each client uploads the local model update to the central server; (4) The server performs a weighted aggregation operation after receiving the data from all parties, obtaining the new global model; (5) Then evaluate the new global model; (6) If the evaluation is appropriate, deploy the new model to the remaining clients [52, 65].

A typical federated learning scenario is to upload only the updated gradient information of the model under the constraint that the local client device is responsible for storing and processing data, and train a single global model on tens of millions to millions of clients devices. The objective function F(w) of the central server is usually expressed as

$$ {}_{w}{}^{\min}F(w), F(w) = \sum_{k=1}^{m} \frac{n_k}{n} F_k(w) \tag{1} $$

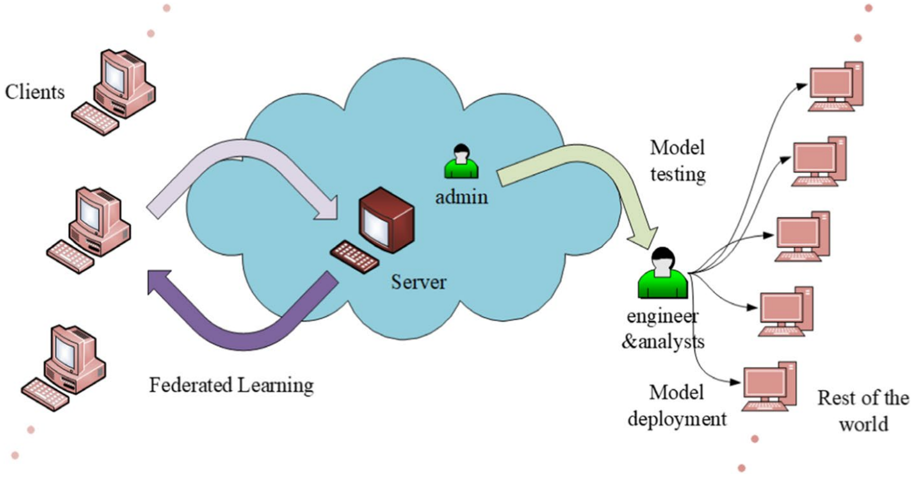

Figure 3 The training cycle of federated learning

$$F_k(w) = \frac{1}{n_k} \sum\nolimits_{i \in d_k} f_i(w) \tag{2}$$

In formula (1), m is the total number of client devices participating in training, n is the sum of all client data volumes, n_k is the data volume of the $\mathrm{k_{th}}$ client, and $F_k(w)$ is the local objective function of the $\mathrm{k_{th}}$ device. In formula (2), d_k is the local data set of the k-th client, and $f_i(w)=a(x_i, y_i, w)$ is the loss function generated by the model with parameter w on the instance (x_i, y_i) in the data set d_k. The sum of the loss functions generated by all instances in d_k divided by the total data volume of client k is the average loss function of the local client, and the loss function is inversely proportional to the model accuracy. Therefore, the objective function optimization of machine learning is usually to make the loss function reach the minimum.

3.2 BERT pre-training model

Word embedding is an inevitable step for NLP. It converts the input text into real-valued vectors to allow the computer to perform numerical operations. From the initial one-hot encoding to the current pre-trained model, the development and progress of word embedding are surprising. Figure 4 shows the Elmo and GPT proposed in recent years, and the BERT model we proposed in this article, they will be introduced in detail below.

The word embedding before Elmo is a static word vector after training. Regardless of the context of the new sentence, the word vector of this word will not change. As shown in Figure 4, the subsequent Elmo and GPT pre-training models use bidirectional LSTM and Transformer for word vector pre-training respectively, the word vector of the obtained word will change according to different context scenarios so that the dynamic word vector is obtained. The two-stage model of BERT and GPT is the same, the first is the language model pre-training, the second is the use of the fine-tuning mode to solve downstream tasks. The main difference from GPT is that a bidirectional language model similar to Elmo is used in the pre-training phase, another point is that the data scale of the language model is larger than that of GPT.

As shown in Figure 5, the BERT model has done a lot of details in terms of input. It uses WordPiece embedding as a word vector, and adds position embedding and sentence segmentation embedding. And there is a [CLS] vector before each text input, which will be used as a specific classification vector later. The calculation of position embedding is shown in formula (3)(4):

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{3}$$

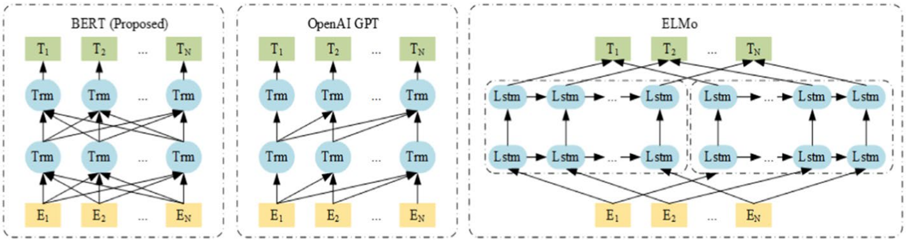

Figure 4 Structure of Elmo, GPT and BERT.

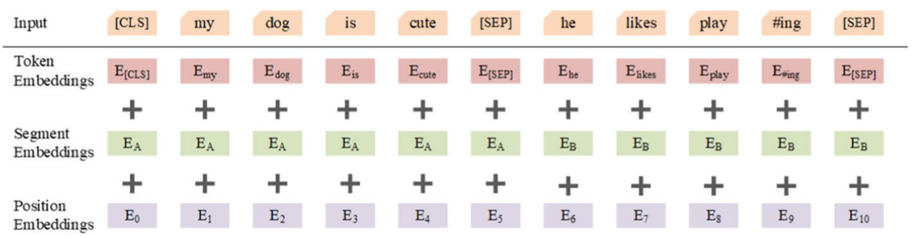

Figure 5 The input structure of the BERT model

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{4}$$

In the formula, pos is the position of the input vector; i is the dimension, and d_{model} is the dimension of the word vector. The text encoder converts the input X into a word vector of d_{model} dimension, which is used for subsequent Transformer encoder learning.

According to Figure 6, the BERT model contains a total of 12 transformer encoder modules. After the text is converted into an input vector, it enters the transformer module for the masked language model and next sentence prediction pre-training task. As shown in Figure 6, a single Transformer module consists of a multi-head attention mechanism, a position-wise feedforward network, a residual connection, and normalization layer connecting these two layers. The multi-head attention mechanism is particularly important.

The self-attention mechanism requires every word in a sentence to calculate attention with other words in the sentence. It can extract the relationship between long-distance words and words, so this mechanism can well capture the context relationship of each word. We use the following formula to calculate the self-attention result.

$$\mathrm{X} \times W^{Q} = Q \tag{5}$$

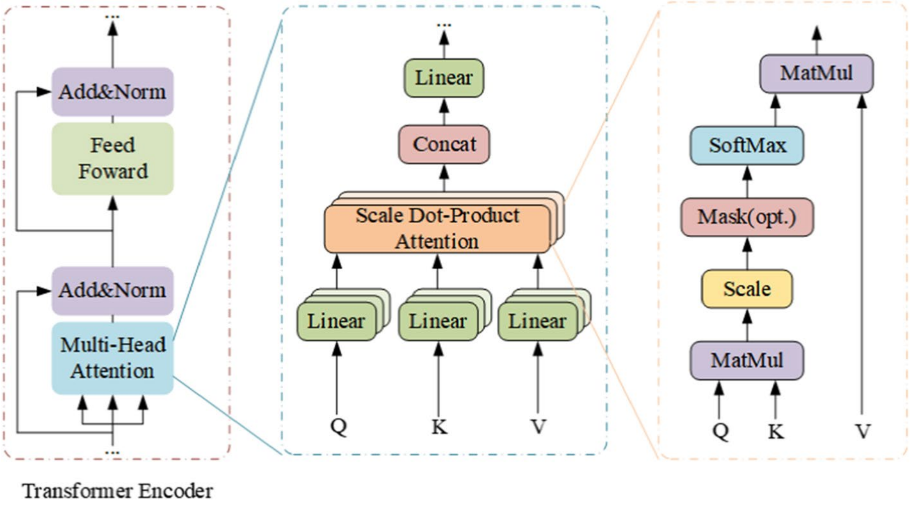

Figure 6 The structure of the transformer encoder

$$\mathrm{X} \times W^{K} = K \tag{6}$$

$$\mathrm{X} \times W^{V} = V \tag{7}$$

$$\mathrm{Z} = \mathrm{softmax}\left(\frac{Q \times K^{T}}{\sqrt{d_{model}}}\right) \times V \tag{8}$$

Combining Figure 6 with the formula, we can see the operation steps of the self-attention mechanism. First, the three matrices $\mathrm{W^Q}$, $\mathrm{W^K}$, and $\mathrm{W^V}$ are randomly initialized. And then the query matrix Q, the key matrix K, and the value matrix V are generated according to formulas (5), (6), (7). Finally we use the softmax function to calculate the weight of each input token according to formula (8). In this formula, $\mathrm{d_{model}}$ is the dimension of the vector, divided by $\sqrt{d_{model}}$ is to prevent the intramolecular product value from being too large.

In the above self-attention mechanism, it only uses a set of $\mathrm{W^Q}$, $\mathrm{W^K}$, and $\mathrm{W^V}$ to transform to obtain the Query, Keys, and Values matrix. The multi-head attention mechanism uses multiple sets of $\mathrm{W^Q}$, $\mathrm{W^K}$, and $\mathrm{W^V}$ to obtain multiple sets of Query, Keys, and Values matrices, and final multiple Z matrices are spliced together. The Transformer Encoder used in this article uses 8 different sets of $\mathrm{W^Q}$, $\mathrm{W^K}$, and $\mathrm{W^V}$.

$$H(Q, K, V) = \left(Z_1 \bigoplus Z_2 \bigoplus \cdots \bigoplus Z_n\right) W^{O} \tag{9}$$

Among them, $\bigoplus$ is the splicing operator, and $\mathrm{Z_i}$ is the representation of the *i*-th attention mechanism.

Position-wise feed-forward networks are calculated by two nonlinear fully connected layers and the nonlinear activation function Relu. Calculated as follows:

$$\mathrm{FFN}(x) = W_2 \bullet \mathrm{Relu}\left(W_1 x + b_1\right) + b_2 \tag{10}$$

$$\mathrm{LN}(x) = \sigma(x + \mathcal{F}(x)) \tag{11}$$

Finally, the input layer, multi-head attention mechanism, and position-wise feed-forward networks will be connected by residual connection and layer normalization. The calculation is shown in Eq. (11). Among it, $\mathcal{F}(x)$ represents the output of the next layer; σ represents the normalization of the layer; $\mathrm{LN}(x)$ represents the output of the layer.

3.3 Feature extractor

As shown in Figure 7, the word embedding from the BERT pre-training model is used as the input of the convolutional layer [31, 34, 63], and then a one-dimensional convolution containing the filter vector is used to slide on the sequence and detect features at different positions. In the structure, the multi-scale CNN is composed of three convolutional layers, and the size of each convolution kernel is different, which is used to extract text features of different scales from word embeddings. We will introduce the principles of multi-scale CNN in detail in this section.

In this paper, considering the grammar, phrase, context and other factors in each enterprise data, it is decided to use the convolutional layer with the convolution kernel window of 1, 2, and 3 to further extract the features. We use $x_{i:i+j}$ to represent the connection of words $\mathrm{x_i}$, $\mathrm{x_{i+1}}$, ..., $\mathrm{x_{i+j}}$, and then the convolution kernel performs sliding window calculation

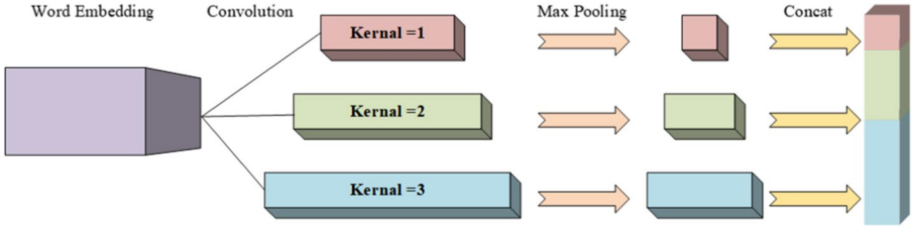

Figure 7 The structure of the multi-scale CNN

with a window of size h to obtain a new feature. For example, the new feature calculated on $x_{i:i+h-1}$ can be expressed as Eq. (12), Where $b \in \mathbb{R}$ is the bias and f is the activation function ReLU. And after we apply the convolution operation to all possible word combinations $\{x_{1:h}, x_{2:h+1}, \cdots, x_{n-h+1:n}\}$, we can get another new feature is shown as Eq. (13).

$$c_i = f\left(\boldsymbol{w}_h \bigotimes \boldsymbol{x}_{i:i+h-1} + b\right) \tag{12}$$

$$\mathbf{c}_h = \left[c_1, c_2, \cdots, c_{n-h+1}\right] \tag{13}$$

After the convolutional layer in CNN is generally a pooling layer, it can reduce the dimensionality of the output matrix. The most commonly used pooling is average pooling and maximum pooling. It can be seen from Figure 7 that in this article, we have chosen the maximum pooling to get the most obvious features in each convolution kernel. After pooling, we stitch the feature vectors obtained by the convolution kernels of different window values to obtain the final value. The calculation formula is as follows:

$$\hat{c}_h = \max\left\{\mathbf{c}_h\right\} \tag{14}$$

$$\hat{C} = \left[\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_d\right] \tag{15}$$

After the feature extractor CNN is the final output layer, different comment texts are input into the softmax layer for sentiment analysis to obtain the final accuracy. During the training process, the network parameters are continuously updated to optimize the loss value of the model. Its final calculation method is shown in Eq. (16), where p_i^j is the correct label of a set of data; $\hat{p}_i^j$ is the predicted distribution, C is the number of categories.

$$L(\hat{p}, p) = -\sum_{i=1}^{N} \sum_{j=1}^{C} p_i^j \log\left(\hat{p}_i^j\right) \tag{16}$$

4 Experimental process and results

4.1 Data set and evaluation index

As shown in Table 1, the data sets used in this experiment come from user reviews of apparel, toys, electronics, movies, videos, and baby products companies. Since this article is a two-category task, these user comments are processed to include sentiment tags

Table 1 Data set case

Data set type	Example	Label
Electronics	great product but is only $ 30 at iriver.com 's stor	1
	i dont like this mouse , i brought , and never work , its useles	0
Apparel	recipient was very satisfied with this blanket as pb are his initials	1
	a red star ! ? ! ? i bet this wo n't sell well in eastern europe .	0
Toys	these make meals a lot more fun for children ... i know my son loves them	1
	fisher price is selling the same item for only $ 33 . $ 139.99 has to be a mistake	0
Video	this is an excellent documentary of shangri-la and its elusive transcendental nature	1
	i love norm macdonald and this is the dumbest movie of all tim	0
Baby	great product - i heard from other mommies that this was the pump to get ; i agree	1
	rent a hospital grade medalia pump . you wont be sorr	0
MR	it's a feel-good movie about which you can actually feel good .	1
	a decidedly mixed bag .	0

Table 2 Data set statistics

Data set type	Training set		Test set		Total
	Positive	Negative	Positive	Negative	
Electronics	705	693	298	302	1998
Apparel	690	710	315	285	2000
Toys	694	706	307	293	2000
Video	694	706	313	287	2000
Baby	700	600	297	303	1900
MR	678	722	306	294	2000

(Negative or Positive). Among these data sets, the data sets of apparel, toys, electronics, videos, and babies were collected by Blitzer et al. [5], and Pang et al. [40] collected other movie data set.

As shown in Table 2, each company provided approximately 2,000 user reviews, totaling more than 12,000. Since the amount of data in each company is small, The original dataset is divided into a training set and a validation set with a ratio of 7:3. The numbers of positive and negative samples in the training and validation set are almost balanced.

In the experiments carried out in this article, we used different comparative experimental methods for the data sets of different companies and adopted the same two evaluation indicators, namely the most commonly used accuracy and F1-score.

4.2 Model comparison

To verify the effect of the sentiment analysis model based on the federated learning framework proposed in this paper, a comparison study has been conducted with modern popular deep learning models, including BERT_CNN, BERT_LSTM, BERT_BiLSTM, BERT_MSCNN and their models based on the federated learning framework. All experiments use the six data sets mentioned above, which can ensure the fairness of comparative experiments. In this comparison experiment, the accuracy comparison is shown in Table 3. In

Table 3, the first four methods are deep learning models that do not use the federated learning framework, while the others are deep learning models based on the federated learning framework. Because they all use the BERT pre-training model for word embedding, it will be omitted later.

In Table 3, the classification accuracy of the deep learning model based on the federated learning framework is the highest in each task.

4.3 Self-comparison of the model

To prove the effectiveness of the word embedding pre-training model BERT and the two-way transformer proposed in this article, this section will propose a new word embedding model glove for comparison. The latter feature extractor is the same as that of Fed_BERT_MSCNN. For other settings, refer to the first section of this chapter.

Figure 8 shows the comparison between f1-score and accuracy between Glove_MSCNN, BERT_MSCNN, Fed_Glove_MSCNN, and Fed_BERT_MSCNN proposed in this article. It can be seen that regardless of whether federated learning is added, the word embedding using the BERT model is more compact and higher than the f1-score and accuracy of the Glove model on the six data sets. So the final average accuracy will be higher. From the figure, we can also see that the accuracy and f1-score of Fed_Glove_MSCNN and Fed_BERT_MSCNN after federated learning has been partially improved compared with the previous Glove_MSCNN and BERT_MSCNN. It once again proves that the federated learning proposed in this paper not only protects the privacy of data but also the improvement of the effect of deep learning models is also obvious.

4.4 The impact of convolutional neural networks

To verify the role of the feature extractor after the pre-training model BERT, this section decided to add the Fed_BERT model and Fed_BERT_MSCNN with different convolution kernel windows to compare with the model proposed in this article. In this section, we use accuracy as the evaluation criterion, the experimental results are as follows.

It can be seen from Figure 9 that for the Fed_BERT model without adding a convolutional neural network as a feature extractor, the model proposed in this article achieved a lead in all tasks, and achieved a leading effect of 1.5% in the final average accuracy rate. For other Fed_BERT_MSCNN models with window values of 1, 2, 3, 4, 5, (2, 3, 4), (3, 4,

Table 3 Comparison with accuracy of existing deep learning models

Data set type	No_Federated				Federated			
	CNN	LSTM	BiLSTM	MSCNN	Fed_CNN	Fed_LSTM	Fed_BiLSTM	Proposed
Electronics	0.845	0.875	0.853	0.855	0.853	0.882	0.867	0.867
Apparel	0.875	0.865	0.877	0.888	0.88	0.87	0.88	0.907
Toys	0.825	0.84	0.868	0.867	0.82	0.832	0.868	0.872
Video	0.74	0.74	0.781	0.768	0.747	0.752	0.783	0.778
Baby	0.872	0.88	0.845	0.855	0.881	0.882	0.848	0.857
MR	0.885	0.875	0.847	0.856	0.89	0.878	0.849	0.875
Average	0.840	0.846	0.845	0.848	0.845	0.849	0.849	0.859

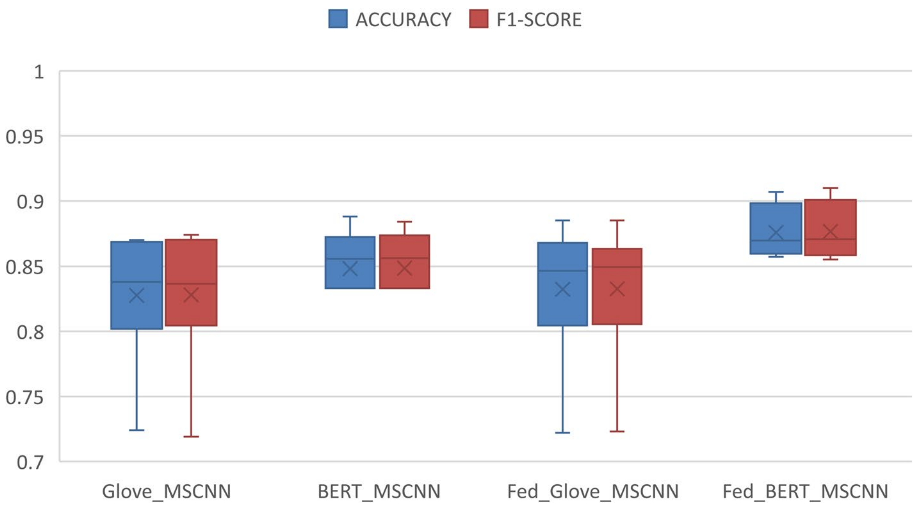

Figure 8 Comparison of the effects of the Glove model and the BERT model

5), the improvement effect on some data sets may not be It is particularly obvious, but the averaged accuracy rate has been improved by about 0.5%-1%.

4.5 Comparison of optimizers

In this article, we propose to use the Ranger optimizer. The optimizer combines the advantages of the LookAhead setting bidirectional exploration and RAdam for the rectification function. The Apparel dataset is used for the comparative study. The Fed_BERT_MSCNN model proposed in this article is compared with the popular SGD and Adadelta optimizers in Figure 11.

The comparison result is shown in Figure 10, the Range optimizer has a higher starting point than SGD and Adadeltaloss. As the epoch increases, the loss of the three optimizers is gradually decreasing. In terms of accuracy, the upward trend of SGD and Adadelta is the same as their loss decline, and the final accuracy rates are divided into 86.8% and 81.5% respectively; the Ranger optimizer, which we use, has a low starting point, it rose sharply to 85.3% in the third epoch, although there were some fluctuations in the subsequent epochs, the final accuracy rate reaches 89.2%.

5 Conclusion and discussion

In this article, We propose a model based on the federated learning framework (Fed_BERT_MSCNN), which contains a Bidirectional Encoder Represent-ations from Transformers (BERT) module and a multi-scale convolution layer. The BERT pre-training model is added to convert the text vector. Compared with the existing pre-training models, the two-way transformer in the BERT pre-training model trains the text vectors and extracts the emotional polarity in the text in a more sophisticated way. A multi-scale

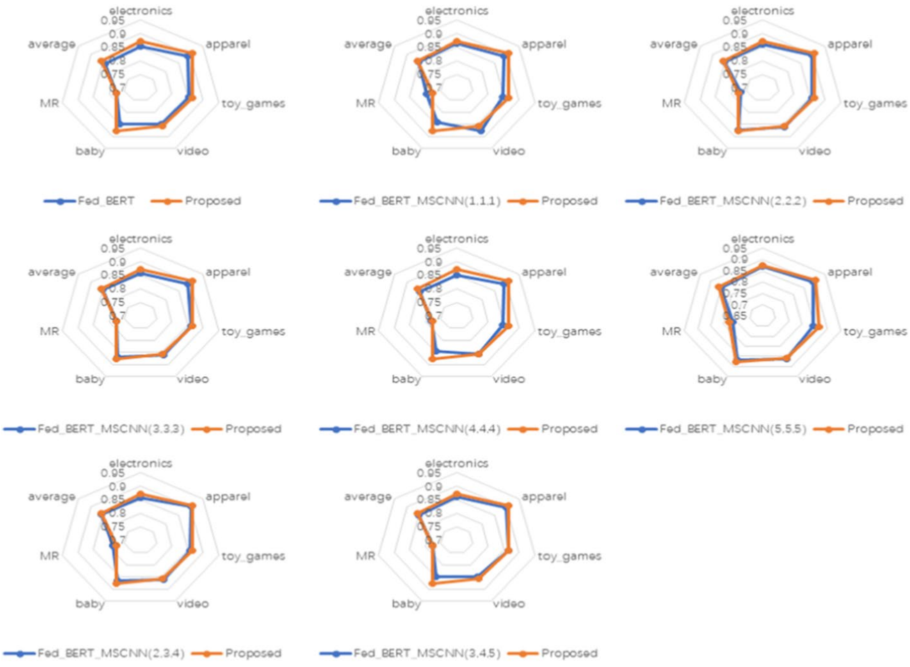

Figure 9 Comparison of models with different convolution kernel windows and models proposed in this article

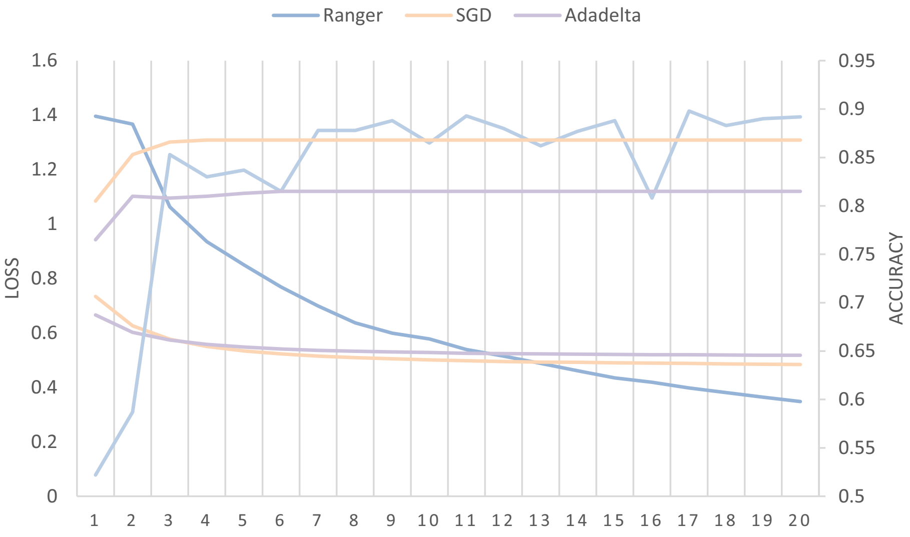

Figure 10 Comparison between different optimizers

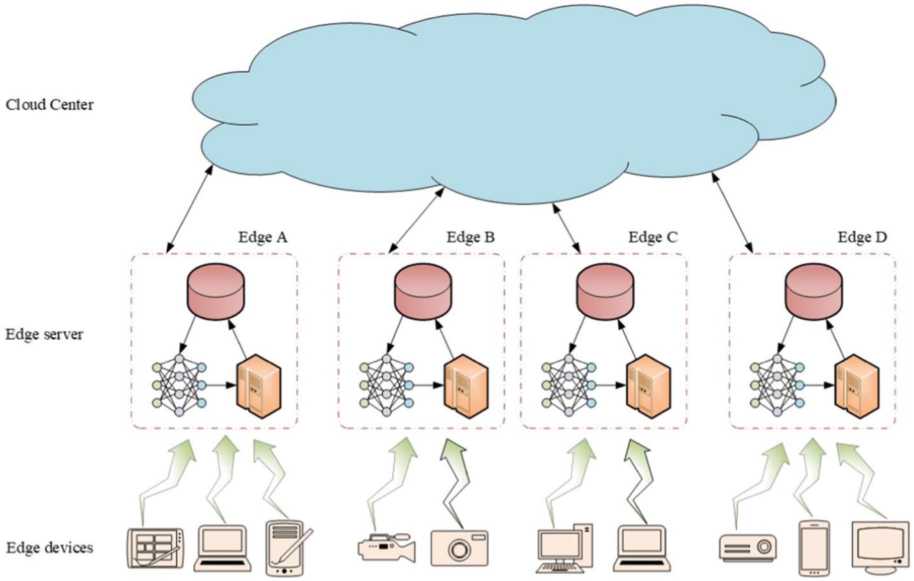

Figure 11 Future vision of distributed computing based on federated learning

convolutional neural network is utilized as a feature extractor, which can extract local features in sentences and construct the feature vector space of the text together with BERT.

Facing the prospects, in the era of big data, how to realize data sharing, promote the collision and integration of multi-source data, and maximize the release of data value under the premise of ensuring data security and privacy has become the main challenges for academia and industry [6, 43]. Federated learning, as an emerging technology to deal with this challenge, has broad application prospects for distributed computing, including edge computing and etc., as shown in Figure 11.

With the popularization of smartphones and mobile Internet applications [8], a large amount of data is generated at the client side of the device [4, 66]. Distributed computing enables the calculation to occur on the local device, and private data [55, 62] needs to be sent to the cloud. Federated learning, as the operating system of edge computing, provides a protocol specification for collaboration and sharing between all parties. It allows edge devices to cooperate and train an optimal global machine learning model without sending source data to cloud devices. In the future, with the further development of the Internet of Things [32], artificial intelligence and edge computing will move forward in the direction of integration. Of course, in addition to the application of the federated learning framework to distributed computing, further improvements of the BERT pre-training model is required, to improve the effectiveness of the model in sentiment analysis tasks or apply the model for other related tasks in NLP.

Declarations

Conflicts of interests The authors declare that they have no competing interests.

References

1. Armbrust, M., Fox, A., Griffith, R., et al.: A view of cloud computing[J]. Communications of the ACM. **53**(4), 50–58 (2010)
2. Azrour, M., Mabrouki, J., Guezzaz, A., et al.: New enhanced authentication protocol for internet of things[J]. Big Data Mining and Analytics. **4**(1), 1–9 (2021)
3. Bengio, Y., Ducharme, R., Vincent, P., et al.: A neural probabilistic language model[J]. The journal of Machine Learning Research. **3**, 1137–1155 (2003)
4. Bi, R., Liu, Q., Ren, J., et al.: Utility aware offloading for mobile-edge computing[J]. Tsinghua Science and Technology. **26**(2), 239–250 (2020)
5. Blitzer, J., Dredze, M., Pereira, F., et al.: Boom-Boxes and Blenders: domain adaptation for sentiment classification[C]//Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), Pereira. 2007, 447.
6. Calero, C., Mancebo, J., García, F., et al.: 5Ws of green and sustainable software[J]. Tsinghua Science and Technology. **25**(3), 401–414 (2019)
7. Chen, Y., Sun, X., Jin, Y.: Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation[J]. IEEE Transactions on Neural Networks and Learning Systems. **31**(10), 4229–4238 (2019)
8. Chen, J., Cai, T., He, W., et al.: A blockchain-driven supply chain finance application for auto retail industry[J]. Entropy. **22**(1), 95 (2020)
9. Dai, W., Kumar, A., Wei, J., et al.: High-performance distributed ML at scale through parameter server consistency models[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 29(1) (2015)
10. Dastjerdi, A.V., Gupta, H., Calheiros, R.N., et al.: Fog computing: Principles, architectures, and applications[M]//Internet of things. Morgan Kaufmann. 61–75 (2016)
11. Devlin, J., Chang, M. W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805 (2018)
12. Gai, K., Qiu, M., Zhao, H., et al.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing[J]. Journal of Network and Computer Applications. **59**, 46–54 (2016)
13. Guo, H.Y.: Text classification based on word vector and topic vector[D]. Wuhan: Huazhong University of Science and Technology. 4–24 (2016)
14. He, Q., Yan, J., Yang, Y., et al.: Chord4s: A p2p-based decentralised service discovery approach[C]//2008 IEEE International Conference on Services Computing. IEEE, 1: 221-228 (2008)
15. He, Q., Zhou, R., Zhang, X., et al.: Keyword search for building service-based systems[J]. IEEE Transactions on Software Engineering. **43**(7), 658–674 (2016)
16. Ho, Q., Cipar, J., Cui, H., et al.: More effective distributed ml via a stale synchronous parallel parameter server[C]//Advances in neural information processing systems. 1223-1231 (2013)
17. Hu, M., Ji, Z., Yan, K., et al.: Detecting anomalies in time series data via a meta-feature based approach[J]. IEEE Access. **6**, 27760–27776 (2018)
18. Hu, M., Feng, X., Ji, Z., et al.: A novel computational approach for discord search with local recurrence rates in multivariate time series[J]. Information Sciences. **477**, 220–233 (2019)
19. Ji, Z., Wang, B., Deng, S.P., et al.: Predicting dynamic deformation of retaining structure by LSSVR-based time series method[J]. Neurocomputing. **137**, 165–172 (2014)
20. Jin, N., Wu, J., Ma, X., et al.: Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification[J]. IEEE Access. **8**, 77060–77072 (2020)
21. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188 (2014)
22. Karpathy, A., Toderici, G., Shetty, S., et al.: Large-scale video classification with convolutional neural networks[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 1725-1732 (2014)
23. Kelt, J.: Search before the purchase: Understanding buyer search activity as it builds to online purchase[J]. DoubleClick, February (2005)
24. Kim, Y.: Convolutional neural networks for sentence classification [C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1746-1751. (2014) https://doi.org/10.3115/v1/D14-181
25. Konečný, J., McMahan, H. B., Yu, F.X., et al.: Federated learning: Strategies for improving communication efficiency[J]. arXiv preprint arXiv:1610.05492, (2016)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems. **25**, 1097–1105 (2012)

27. Li, M., Andersen, D. G., Park, J. W., et al.: Scaling distributed machine learning with the parameter server[C]//11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14). 583-598 (2014)
28. Li, T., Sanjabi, M., Beirami, A., et al.: Fair resource allocation in federated learning[J]. arXiv preprint arXiv:1905.10497 (2019)
29. Liang, W., Hu, Y., Zhou, X., et al.: Variational Few-Shot Learning for Microservice-Oriented Intrusion Detection in Distributed Industrial IoT[J]. IEEE Transactions on Industrial Informatics. (2021)
30. Lin, Y., Han, S., Mao, H., et al.: Deep gradient compression: Reducing the communication bandwidth for distributed training[J]. arXiv preprint arXiv:1712.01887 (2017)
31. Liu, Y., Pei, A., Wang, F., et al.: An attention-based category-aware GRU model for the next POI recommendation[J]. International Journal of Intelligent Systems. (2021)
32. Mabrouki, J., Azrour, M., Fattah, G., et al.: Intelligent monitoring system for biogas detection based on the internet of things: Mohammedia, morocco city landfill case[J]. Big Data Mining and Analytics. **4**(1), 10–17 (2021)
33. Mahmud, M.S., Huang, J.Z., Salloum, S., et al.: A survey of data partitioning and sampling methods to support big data analysis[J]. Big Data Mining and Analytics. **3**(2), 85–101 (2020)
34. Malek, Y.N., Najib, M., Bakhouya, M., et al.: Multivariate deep learning approach for electric vehicle speed forecasting[J]. Big Data Mining and Analytics. **4**(1), 56–64 (2021)
35. Mao, Y., You, C., Zhang, J., et al.: A survey on mobile edge computing: The communication perspective[J]. IEEE Communications Surveys & Tutorials. **19**(4), 2322–2358 (2017)
36. McMahan, H. B., Moore, E., Ramage, D., et al.: Federated learning of deep networks using model averaging[J]. arXiv preprint arXiv:1602.05629, (2016)
37. McMahan, B., Moore, E., Ramage, D., et al.: Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR. 1273–1282 (2017)
38. Niu, F., Recht, B., Ré, C., et al.: Hogwild!: A lock-free approach to parallelizing stochastic gradient descent[J]. arXiv preprint arXiv:1106.5730 (2011)
39. O'Connor, B., Balasubramanyan, R., Routledge, B. R., et al.: From tweets to polls: Linking text sentiment to public opinion time series[C]//Fourth international AAAI conference on weblogs and social media. (2010)
40. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales[J]. arXiv preprint cs/0506075 (2005)
41. Peters, M., Neumannm, M., Iyyer, M., et al.: Deep Contextualized Word Representations[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). (2018)
42. Qi, L., Wang, X., Xu, X., et al.: Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing[J]. IEEE Transactions on Network Science and Engineering. (2020)
43. Qi, L., Hu, C., Zhang, X., et al.: Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment[J]. IEEE Transactions on Industrial Informatics. **17**(6), 4159–4167 (2020)
44. Qiu, M., Dai, H.N., Sangaiah, A.K., et al.: Guest editorial: Special section on emerging privacy and security issues brought by artificial intelligence in industrial informatics[J]. IEEE Transactions on Industrial Informatics. **16**(3), 2029–2030 (2019)
45. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving language understanding by generative pre-training[J]. (2018)
46. Rakhlin, A.: Convolutional neural networks for sentence classification[J]. GitHub, (2016)
47. Rehak, D., Dodds, P., Lannom, L.: A model and infrastructure for federated learning content repositories[C]//Interoperability of web-based educational systems workshop. 143 (2005)
48. Satyanarayanan, M.: The emergence of edge computing[J]. Computer. **50**(1), 30–39 (2017)
49. Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A.: A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective[J]. Computer Networks. **107496**, (2020)
50. Shi, W., Cao, J., Zhang, Q., et al.: Edge computing: Vision and challenges[J]. IEEE internet of things journal. **3**(5), 637–646 (2016)
51. Tan, X., Zhang, J., Zhang, Y., et al.: A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network[J]. Tsinghua Science and Technology. **26**(1), 36–47 (2020)
52. Wang, F., Zhu, H., Srivastava, G., et al.: Robust collaborative filtering recommendation with user-item-trust records[J]. IEEE Transactions on Computational Social Systems. (2021)
53. Wei, X., Lin, H., Yu, Y., et al.: Low-resource cross-domain product review sentiment classification based on a CNN with an auxiliary large-scale corpus[J]. Algorithms. **10**(3), 81 (2017)

54. Xu, Z.W.: Cloud-Sea Computing Systems: Towards Thousand-Fold Improvement in Performance per Watt for the Coming Zettabyte Era[J]. Journal of Computer Science and Technology. **029**(002), 177–181 (2014)
55. Xu, Y., Zhang, C., Zeng, Q., et al.: Blockchain-enabled accountability mechanism against information leakage in vertical industry services[J]. IEEE Transactions on Network Science and Engineering. (2020)
56. Yan, K., Shen, W., Jin, Q., et al.: Emerging privacy issues and solutions in cyber-enabled sharing services: From multiple perspectives[J]. IEEE Access. **7**, 26031–26059 (2019)
57. Yuan, Y., Huang, J., Yan, K.: Virtual Reality Therapy and Machine Learning Techniques in Drug Addiction Treatment[C]//2019 10th International Conference on Information Technology in Medicine and Education (ITME). IEEE. 241–245 (2019)
58. Yuan, Y., Huang, J., Ma, X., et al.: Children's Drawing Psychological Analysis using Shallow Convolutional Neural Network[C]//2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). IEEE. 692–698 (2020)
59. Zhang, X., Dou, W., He, Q., et al.: LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis[C]//2017 IEEE 33rd International Conference on Data Engineering (ICDE). IEEE, 983-994 (2017)
60. Zhang, W., Chen, X., Jiang, J.: A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems[J]. Tsinghua Science and Technology. **26**(1), 95–111 (2020)
61. Zhang, J., Yan, K., Mo, Y.: Multi-Task Learning for Sentiment Analysis with Hard-Sharing and Task Recognition Mechanisms[J]. Information. **12**(5), 207 (2021)
62. Zhang, C., Xu, Y., Hu, Y., et al.: A blockchain-based multi-cloud storage data auditing scheme to locate faults[J]. IEEE Transactions on Cloud Computing. (2021)
63. Zhou, X., Liang, W., Kevin, I., et al.: Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data[J]. IEEE Transactions on Emerging Topics in Computing. (2018)
64. Zhou, X., Xu, X., Liang, W., et al.: Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT[J]. IEEE Internet of Things Journal. (2021)
65. Zhou, X., Xu, X., Liang, W., et al.: Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS[J]. IEEE Transactions on Industrial Informatics. (2021)
66. Zhou, X., Yang, X., Ma, J., et al.: Energy Efficient Smart Routing Based on Link Correlation Mining for Wireless Edge Computing in IoT[J]. IEEE Internet of Things Journal. (2021)
67. Zonglin, L.I.U., Meishan, Z., Ranran, Z., et al.: Multi-task learning model for legal judgment predictions with charge keywords[J]. Journal of Tsinghua University (Science and Technology). **59**(7), 497–504 (2019)