



# Text Classification on Social Network Platforms Based on Deep Learning Models

YA Chen<sup>1</sup>, Tan Juan<sup>2</sup>, and Hoekyung Jung<sup>3\*</sup>

<sup>1</sup>Professor, Department of Information Engineering, GuangXi Transport Vocational and Technical College, Naming Guangxi, China

<sup>2</sup>Professor, Weifang University of Science and Technology, Weifang 262700 Shandong, China

<sup>3</sup>Professor, Department of Computer Engineering, Paichai University, Daejeon 35345, Korea

## Abstract

The natural language on social network platforms has a certain front-to-back dependency in structure, and the direct conversion of Chinese text into a vector makes the dimensionality very high, thereby resulting in the low accuracy of existing text classification methods. To this end, this study establishes a deep learning model that combines a big data ultra-deep convolutional neural network (UDCNN) and long short-term memory network (LSTM). The deep structure of UDCNN is used to extract the features of text vector classification. The LSTM stores historical information to extract the context dependency of long texts, and word embedding is introduced to convert the text into low-dimensional vectors. Experiments are conducted on the social network platforms Sogou corpus and the University HowNet Chinese corpus. The research results show that compared with CNN + rand, LSTM, and other models, the neural network deep learning hybrid model can effectively improve the accuracy of text classification.

**Index Terms:** Big data, Deep learning model, Neural network, Social network platform, Text classification

## I. INTRODUCTION

With the development of the Internet and mobile social network platforms, the amount of text information has substantially increased. Although text information has great potential value, it is chaotic in the network because of the strong real-time characteristics of the network platform. However, effective information organization and management is lacking. Text classification can reduce information clutter by effectively organizing and managing text information, and as such, it has been widely used in information sorting, personalized news recommendation, spam filtering, and user intention analysis. This study established a hybrid model consisting of an ultra-deep convolutional neural network (UDCNN) and long-short-term memory network (LSTM), whereby word embedding is used to convert text into low-

dimensional vectors to improve the accuracy of word vectorized text classification. The classification effect of the model was verified through experiments [1]. Expenditure (item) recommendation functions using big data were added to the existing research fund management system, and the implications for future research directions were discussed.

## II. SYSTEM MODEL AND METHODS

### A. Word Embedding

If text is converted into a vector in a literal order and directly encoded, the vector dimension increases inordinately, and the characteristics of the dependency relationship between words and sentences in the natural language are

Received 6 November 2021, Revised 19 April 2022, Accepted 19 April 2022

\*Corresponding Author Hoekyung Jung (E-mail: [hkjung@pcu.ac.kr](mailto:hkjung@pcu.ac.kr), Tel: +82-42-520-5640)

Department of Computer Engineering, PaiChai University, Daejeon, 35345 Korea

Open Access <https://doi.org/10.56977/jicce.2023.21.1.9>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

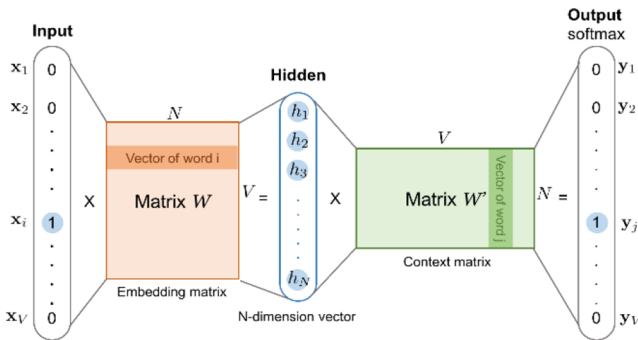
ignored. To solve this problem and enable the LSTM network to better utilize the context in natural language, this study combined word embedding with a hybrid model to convert the text into low-dimensional vectors. Consequently, the synonyms in the text before and after being converted into low-dimensional vector form are adjacent in vector space, thus allowing words synonymous in context to be aggregated.

**1) Related Definitions**

A simple word vector representation is a one-hot representation of a word through a very long vector. The size of dictionary D is the length of the vector. The vector component has only ones, whereas the other positions are all zeros and ones.. However, this type of word vector presents the problem of excessively high dimensions due to the huge amounts of data in deep learning. To address the high dimensionality problem of the one-hot expression and enable the vector to describe the connection between words, this study utilized the distribution expression to express the word vector [2].

**2) Skip-gram Model**

The skip-gram model includes three layers: the input, projection, and output layers. A schematic is shown in Fig. 1.



**Fig. 1.** Skip-gram model.

The skip-gram model uses a word to predict the probability of the surrounding words. The middle word  $w_t$  is known, and the probability that the surrounding  $2n$  words  $w_{t-1}, w_{t-1+1}, \dots, w_{t-1+n}, w_{t+1}, \dots, w_{t+n}$  belong to a word in the dictionary is derived. The set  $w_{t-1}, w_{t-1+1}, \dots, w_{t-1+n}, w_{t+1}, \dots, w_{t+n}$  of words surrounding  $w_t$  represent the context of  $w_t$ , denoted as  $Context(w)$ . The model can calculate the conditional probability of the surrounding words  $c_i$  based on the middle word  $w_t$ , as

$$P(c_i | w_t) \tag{1}$$

where,  $c_i \in Context(w)$ . For a certain sentence S, the skip-gram model can calculate the probability that S is a natural language using the formula:

$$P(S) = \prod_{w \in S} \prod_{c_i \in Context(w)} (P(c_i | w_t)) \tag{2}$$

where  $P(S)$  represents the probability that sentence S is natural language, and  $w$  is the word in sentence S. The goal of model training is to maximize the probability of  $P(S)$ . For input text T, the expression for the probability of the text can be obtained as

$$P(T) = \prod_{S \in T} \prod_{w \in S} \prod_{c_i \in Context(w)} (P(c_i | w_t)) \tag{3}$$

To determine the maximum conditional probability, let the likelihood function of skip-gram be

$$L(\theta) = \prod_{S \in T} \prod_{w \in S} \prod_{c_i \in Context(w)} (P(c_i | w_t); \theta) \tag{4}$$

where  $\theta$  is the parameter to be estimated. The goal in solving the model is to determine the maximum value of the objective function. Therefore, the likelihood function is converted into a base likelihood function:

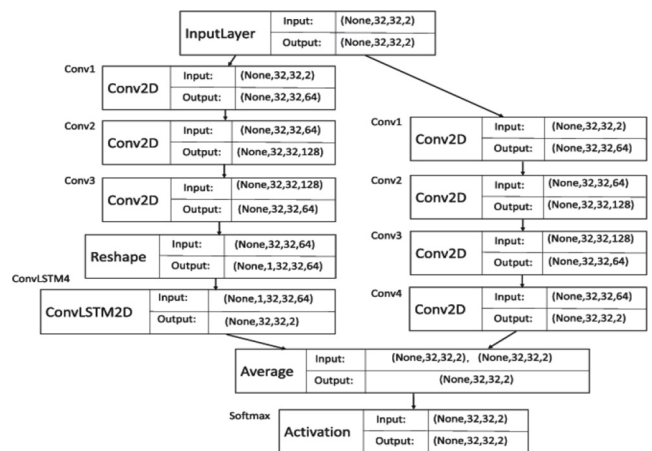
$$l(\theta) = lbL(\theta) = \frac{1}{V} \sum_{S \in T} \sum_{w \in S} \sum_{c_i \in Context(w)} lbP(c_i | w_t) \tag{5}$$

where V represents the dictionary size. This study used Jieba word segmentation, Word2Vec tools, and the skip-gram model to train the text and obtain word vectors. The obtained vectors are low-dimensional, and the vectors of synonyms are adjacent in vector space.

**B. UDCNN and LSTM Hybrid Model**

This study established a hybrid model combining UDCNN and LSTM, thereby using it to improve the accuracy of Chinese text classification. The structure of the model is shown in Fig. 2, where  $FC(I, O)$  represents a fully connected layer with input length I and output length O.

The word embedding layer contains 10 convolutional layers (Conv), and three fully connected layers (FC), thereby



**Fig. 2.** Hybrid model of UDCNN and LSTM.

forming a UDCNN network structure with a total of 14 layers. Concurrently, LSTM is combined with the UDCNN network structure to form a hybrid model. To optimize the memory usage of the deep network structure model, this study combined the VGG and ResNets models following these two rules in defining the UDCNN model structure [3]:

1) If the output vector remains unchanged after convolution, the number of convolution kernels and feature image size remain unchanged. 2) If the output vector is halved, the number of convolution kernels and feature image size are doubled.

With rule 2, the increase in depth can effectively improve the classification effect in the VGG and ResNet models. However, the increase in depth significantly increases the demand for memory. According to the design guidelines of convolutional neural networks, the spatial size of the convolution and convolution data should be gradually reduced during the convolution process. Therefore, in this study, the output vector was halved in the proposed hybrid model. To reduce the memory pressure while ensuring network structure capacity and avoid losing excessive information, in this study, the model doubles the number of convolution kernels and the size of the feature image when the output vector is halved.

The hybrid model has a total of 14 layers. The first layer is the word-embedding layer. The input text sequence is expanded into a sequence of word vectors and used as the input of the convolutional layer. The UDCNN network structure after the word embedding layer consists of the first and second convolutional layers, which comprise 64 convolution kernels of size three. A pooling operation is performed on the convolution result, essentially connecting the two convolution layers, to obtain 128 convolution kernels of size three. Subsequently, three pooling operations are performed, thus connecting two convolutional layers for each pooling operation. The pooling operation is repeated, and the three fully connected layers are connected to obtain the classification result.

In Fig. 2, the UDCNN and LSTM hybrid model includes five pooling operations: the first three average the output pooling value, whereas the last two obtain the maximum value of the pooling operation. The convolutional layer between every pair of pooling operations in the model is called a convolutional block. The UDCNN network structure of the second convolution block is detailed in Fig. 3.

To prevent overfitting, the dimensionality of the features was reduced; the memory usage of the mixed model was optimized; the downsampling factor was set to 2 during each average pooling operation; and the output vector was halved. According to the previous two rules, the number of convolution kernels and the size of each convolution block's feature image were progressively changed from 64 to 128, 256, and 512, and k-max was performed after the fourth and fifth con-

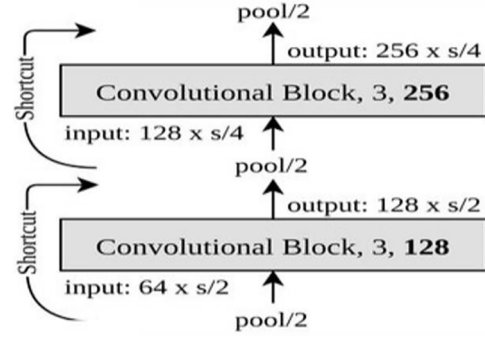


Fig. 3. The second convolution block in the UDCNN network structure

volution blocks. After each selection of  $k$  local optimal eigenvalues for the sampling area, redundant features were discarded to guarantee the generation of a fixed-dimensional eigenvector [4]. Additionally, three fully connected layers were set up after the last maximum pooling operation. Finally, the classification result was obtained using the SoftMax function. In SoftMax regression, the probability of classifying  $x$  into category  $j$  is

$$P(y^i = j | x^i; \theta) = \frac{e^{\theta_j^T x^i}}{\sum_{i=1}^k e^{\theta_j^T x^i}} \quad (6)$$

In the case of multiple layers in the network structure model, to increase the convergence speed and reduce the learning cycle, the activation function ReLU is set in the UDCNN convolutional and fully connected layers, using the formula:

$$F(x) = \max(0, x) \quad (7)$$

In UDCNN, the introduction of batch standardization operations and shortcut connections provides a solution for the problem of decreasing accuracy with increasing depth in traditional convolutional neural networks. It is precisely the increase in the depth of the UDCNN network that effectively improves the ability of extracting text features. Therefore, the UDCNN has a very positive effect on image processing and speech recognition. However, because natural language has expressions such as inversion and preposition, the current text may have a strong context-dependent relationship with the previous text. Some previous historical information may be needed for in-text training, and the UDCNN does not have the ability to retain historical information. Because of the characteristics of natural language and the shortcomings of UDCNN in this respect, this study combined LSTM and UDCNN to form a hybrid model. The LSTM unit structure is shown in Fig. 4 [5,6].

Three gates control the update and deletion of historical

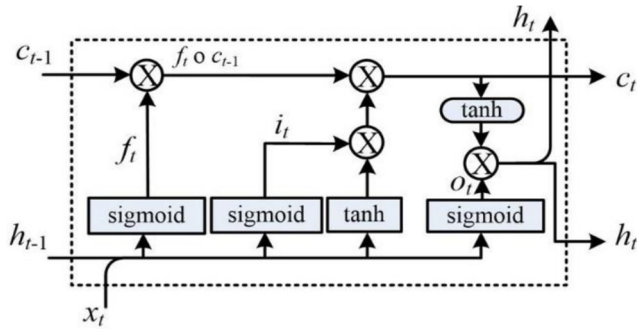


Fig. 4. LSTM unit structure.

information: the input, output, and forget gates. The input gate is used to control the current unit state input; the output gate controls the output of the current LSTM cell; and the forget gate is used to control the historical information stored in the cell in the last round. The model remembers that the input, output, and forget gates at time  $t$  are  $i_t, o_t, f_t$ , respectively, to update the state of the neuron using the following calculation method.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{8}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{9}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{10}$$

The above analysis suggests that the three gates do not provide additional information but only play a role in limiting the amount of information while ensuring that each LSTM remote unit remembers historical information, which can compensate for the shortcomings of the RNN network. Simultaneously, the three gates only play a filtering role, and thus, the activation function uses a sigmoid. The ultra-deep layers of the UDCNN can effectively extract the features of the text vectors. The memory unit of LSTM retains historical information during the training process according to natural language characteristics before and after dependence, thus compensating for the shortcomings of the UDCNN. Therefore, this study proposed a hybrid model of UDCNN and LSTM for text classification, thereby effectively improving the classification.

In Fig. 2, before the fully connected layer FC (4096, 2048), the merge fusion layer in the Keras framework is used to fuse UDCNN and LSTM. The merge layer can provide a series of methods for fusing two layers or two models. The output result of this method is an object with the same structure as that of the layer obtained by merging using the above code, and it can be used as the output of a normal layer. Finally, three fully connected layers are connected, and SoftMax is used to output the classification results. At this point, UDCNN and LSTM are integrated to form a hybrid model, as shown in Fig. 2 [7].

### III. EXPERIMENTAL RESULTS

#### A. Experimental Data

##### 1) Corpus of Training Word Vectors

For knowledge mining, in general, knowledge base corpus training is used to obtain word vectors; however, improved text classification requires wider coverage that reflects current Internet hotspots. This study used a WeChat public account publication with multiple fields. The corpus is a balanced Chinese corpus with 8 million articles and 65 billion words, which can be used for training to obtain high-quality word vectors.

##### 2) Text Classification Corpus

In this study, the Sogou corpus was used to test the text classification effect of the proposed UDCNN and LSTM hybrid model. The Sogou corpus comprises network-wide news data provided by the Sogou social network platform laboratory. The dataset was generated from several news sites, including Sina, Netease, Tencent, and Phoenix News from June to July 2012, comprising domestic, international, sports, entertainment, and social news from 18 channels, with URL and text information. Given that the dataset is in XML format, before the experiment, a script was used to parse the data in the news title and content into corresponding categories. During processing, each article was saved as a text file for news headlines and news content, and then each text was segmented using the jieba word segmentation tool. After processing was completed, the size of the dataset was 1.43 G, and the processed dataset was used as the training and test corpus for text classification. Owing to the large amounts of complete experimental data and limitations of experimental equipment, this study selected 12 categories, and from each category, parts of the data were randomly extracted for the experiment. The text category and quantity distribution of the Sogou corpus used in the experiment are presented in Table 1, where 90% was used as the training set and the remaining 10% as the test set [8].

Table 1. Sogou corpus text category and quantity

Category	Quantity	Category	Quantity
Car	2925	Education	2295
Finance	2588	Media	2128
Culture	2685	Physical education	2990
Public welfare	383	Tourism	2546
Health	2542	Female	2532
IT	4496	Entertainment	4383

The University HowNet text classification corpus, organized and provided by the university, was divided into 20 categories, including more than 9,000 documents. The distri-

bution of its categories and the number of texts are presented in Table 2.

**Table 2.** Types and quantities of texts in the University HowNet Corpus

Category	Quantity	Category	Quantity
Art	742	Mineral	34
Literature	34	Transport	59
Education	61	Surroundings	1218
Philosophy	45	Agriculture	1022
History	468	Economic	1601
Space	642	Legal	52
Energy	33	Medicine	53
Electronic	28	Military	76
Communication	27	Political	1026
Computer	1358	Movement	1254

As the number of texts in categories, such as law and minerals was too small, categories and texts with fewer than 100 texts were deleted during this experiment.

## B. Experiment Preparation and Experiment Environment

### 1) Word Vector Pre-training

This experiment used Gensim's Word2Vec for word vector pre-training, and the training corpus was the WeChat official account article described in the previous section. The word segmentation tool Jieba was used to add a dictionary of 500,000 entries, and new word discovery was turned off. The dictionary was pieced together from multiple dictionaries on the Internet, and a few words were deleted. The training model was the skip-gram described in Section 2. The model has 352,196 words consisting of Chinese and basic common English words. The vector dimension was set to 256, training window size to 10, and minimum word frequency to 64. Further, herein, 10 iterations were performed, including a high-quality word vector [9].

### 2) Model Building Environment

In the hybrid model shown in Fig. 2, the network layer, activation function, loss function, and optimizer can be regarded as independent modules during the training process. The model was constructed using Keras's API.

### 3) Experimental Environment and Parameter Settings

The experimental environment is presented in Table 3. The number of iterations was set to 30.

In the experiment, the loss function was set as categorical cross-entropy. To solve the problem of premature termination in the learning process, the optimizer was set to RMS Prop, which uses an attenuation coefficient to introduce a

**Table 3.** Experimental environment

Software and hardware	Configuration
CPU	Xeon E5-1620 v3
RAM	DDR4, 8 GB
GPU	Nvidia Quadro K2200, 4 GB
Operating system	Windows 8.1
Development environment	Anaconda 4.3.0 Theano 0.8.0

certain amount of attenuation into each round, and the parameter update rules are indicated in (11) and (12):

$$E_t(g^2) = 0.9E_{t-1}(g^2) + 0.1g(\theta_t) \quad g(\theta_t) \quad (11)$$

$$\theta_{t+1} = \theta_t - [diag(E_t(g^2)) + dI]^{-\frac{1}{2}} g(\theta_t) \quad (12)$$

where  $t = 0, 1 \dots$  represents the number of iterations and  $g^2$  represents the vector of the gradient square. The adjustable parameter was 0.9;  $\odot$  is the element product operator, which represents the corresponding positions of two matrices or vectors. By multiplying the elements of  $diag(v)$ , a diagonal matrix was generated according to vector  $v$ , where  $d$  is a very small integer (approximately  $10^{-8}$  in value), and  $I$  is the identity matrix.

### 4) Experimental Evaluation Standards

To evaluate the effect of the mixed model proposed in this study on text classification, the common text classification metric of field-precision was used to test the model. The mixing matrix was established according to the classification results presented in Table 4.

**Table 4.** Mixed matrix of classification results

Whether it belongs to this category after classification	The original text belongs to this category	The original text does not belong to this category
Belong to this category after classification	A	B
Not in this category after classification	C	D

Generally, accuracy is used as the main metric when evaluating classification performance and calculated as follows.

$$R = \frac{A+D}{A+B+C+D} \times 100\% \quad (13)$$

## C. Research Fund Rejection Detection System

### 1) Comparison between UDCNN and other CNN Models

In this experiment, the text classification effect of the 14-

layer UDCNN model, proposed in this study, was compared with other CNN models. In Table 5, the experimental corpus used by the model is the Sogou corpus; evidently, ConvNet (event) and ConvNet (event + bigram + trigram) were both improved convolutional neural networks. ConvNet (event) uses event features in the text for convolution, whereas ConvNet (event + bigram + trigram) uses additional two three-element phrase information for convolution. These two methods clarify the feature source during feature extraction. Nonetheless, other feature information in the same text can be easily ignored, with focus only on the event feature.

**Table 5.** Comparison of classification effect between UDCNN and other CNN models

Model	Accuracy
ConvNet(event) model	93
ConvNet (event + bigram + trigram) model	95.1
Lg. w2vConv model	95.61
Sm. w2vConv model	95.46
UDCNN model	97.93

The UDCNN model is similar to the convolutional neural network used in image processing and speech recognition. It increases the convolutional network structure (14 layers) and sets the convolution kernel size to three. However, only increasing the network depth causes the gradient to disappear and the accuracy to decrease; therefore, this study introduced a shortcut connection and batch standardization into UDCNN to solve this problem. Finally, the depth advantage of the UDCNN network was used along with a small convolution kernel to extract text features effectively.

**2) Influence of the Mixture of UDCNN and LSTM on Text Classification**

In this experiment, the Sogou corpus was classified using the UDCNN, LSTM, and the hybrid UDCNN and LSTM models. The classification effect comparison is presented in Table 6; evidently, the effect of mixing the UDCNN and LSTM was better than that of a single UDCNN or LSTM model. This is because UDCNN neglects the convolution operation of the word vector corresponding to the text. The context-dependency relationship in an article, and although a single LSTM model can use the gates in the model to store and control information and improve context information, the number of layers is not as deep as the UDCNN model, and the feature extraction capacity of word vectors is insufficient. Therefore, this study combines the deep layer of the UDCNN model with the small convolution kernel of the LSTM model to save context information, which can improve accuracy [10].

**Table 6.** Comparison of the classification effects of the three models

Model	Accuracy (%)
UDCNN model	97.93
LSTM model	91.07
UDCNN and LSTM hybrid model	98.96

**3) Comparison of the effects of this mixed model and other classification models**

In this experiment, the UDCNN and LSTM hybrid model and other classification models were used to compare the classification effects of the Sogou corpus and University HowNet corpus. The results are presented in Tables 7 and 8, respectively, where one of the comparison models has increased the attention mechanism, and some of the others use a certain distribution to initialize the input randomly. However, these methods improved only in a single aspect. For classification, the improvement of the effect was limited, and the use of the UDCNN and LSTM hybrid model takes advantage of the UDCNN super-deep convolution combined with the advantages of the LSTM model to save context information. The fusion of these two can significantly improve the accuracy of text classification. On the Sogou corpus, the accuracy rate of the CLKNN model in the literature reached 96.5%, and on the University HowNet corpus, the accuracy rate of the LSTM model reached 91.30%, whereas the accuracy rates of the UDCNN and LSTM hybrid model on the Sogou corpus and the University HowNet corpus reached 98.96% and 93.10%, respectively, which are significantly compliant [11].

**Table 7.** Comparison of the classification effect of different models in the Sogou corpus

Model	Accuracy (%)
Attention-based LSTM model	92.18
Combination of positive and negative sequence attention-based LSTM model	94.81
LSTM model	95.18
BoW model	92.85
CLKNN model	96.5
C-LSTM model	94.6
CNN + Skip-gram model	91.34
MT-LSTM model	94.4
LSTM model	95.6
UDCNN and LSTM hybrid model	98.96

**Table 8.** Comparison of classification effects of different models in the University HowNet Corpus

Model	Accuracy (%)
CNN + rand model	89.41
LSTM model	91.3
Labeled-LDA (allocation) model	90.4
UDCNN and LSTM hybrid model	93.1

#### IV. DISCUSSION AND CONCLUSIONS

This study used word embedding to convert the text into low-dimensional vectors and ensures that the vectors of words with similar context are adjacent in vector space. After word vectorization, UDCNN and LSTM were combined into a hybrid model to classify the text. The hybrid model combines the advantages of the UDCNN super-deep convolution with the advantages of the LSTM model to save context information and effectively improve the accuracy of text classification in the feature extraction process. However, the proposed UDCNN and LSTM hybrid model focuses more on the entire text operation. In an actual text, the text category can be obtained based on a certain central paragraph or article keywords. Therefore, in future, the use of keywords and attention mechanism will be introduced into the mixed model to further improve the efficiency and accuracy of text classification.

#### ACKNOWLEDGMENTS

This study was supported by “the Young and Middle-aged Promotion Project of GuangXi Education Department, China (Grant: 2019KY348)” and by “fine-grained Image Classification in vehicle application technology research (Grant: 2020KY24019).” This study was also supported by the China Scholarship Council (No. 202008450033).

#### REFERENCES

- [ 1 ] A. Bashar, “Survey on evolving deep learning neural network architectures,” *Journal of Artificial Intelligence*, vol. 1, no. 2, pp. 73-82, 2019. DOI: 10.36548/jaicn.2019.2.003.
- [ 2 ] M. Zeng, F. Zhang, F. X. Wu, Y. Li, J. Wang, and M. Li, “Protein-protein interaction site prediction through combining local and global features with deep neural networks,” *Bioinformatics*, vol. 36, no. 4, pp. 1114-1120, Feb. 2020. DOI: 10.1093/bioinformatics/btz699.
- [ 3 ] A. Alarifi, A. Tolba, Z. Al-Makhadmeh, and W. Said, “A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks,” *The Journal of Supercomputing*, vol. 76, no. 6, pp. 4414-4429, Jun. 2020. DOI: 10.1007/s11227-018-2398-2.
- [ 4 ] S. Hassanpour, N. Tomita, T. DeLise, B. Crosier, and L. A. Marsch, “Identifying substance use risk based on deep neural networks and Instagram social media data,” *Neuropsychopharmacology*, vol. 44, no. 3, pp. 487-494, Feb. 2019. DOI: 10.1038/s41386-018-0247-x.
- [ 5 ] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, “Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion,” *Information Processing & Management*, vol. 56, no. 4, pp. 1245-1259, Jul. 2019. DOI: 10.1016/j.ipm.2019.02.018.
- [ 6 ] L. Xiang, G. Guo, J. Yu, V. S. Sheng, and P. Yang, “A convolutional neural network-based linguistic steganalysis for synonym substitution steganography,” *Mathematical Biosciences and Engineering*, vol. 17, no. 2, pp. 1041-1058, 2020. DOI: 10.3934/mbe.2020055.
- [ 7 ] C. C. Li and B. Liu, “MotifCNN-fold: protein fold recognition based on fold-specific features extracted by motif-based convolutional neural networks,” *Briefings in bioinformatics*, vol. 21, no. 6, pp. 2133-2141, Nov. 2020. DOI: 10.1093/bib/bbz133.
- [ 8 ] P. M. Shakeel, A. Tolba, Z. Al-Makhadmeh, and M. M. Jaber, “Automatic detection of lung cancer from biomedical data set using discrete AdaBoost optimized ensemble learning generalized neural networks,” *Neural Computing and Applications*, vol. 32, no. 3, pp. 777-790, Feb. 2020. DOI: 10.1007/s00521-018-03972-2.
- [ 9 ] H. Sadr, M. M. Pedram, and M. Teshnehlab, “A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks,” *Neural Processing Letters*, vol. 50, no. 3, pp. 2745-2761, Dec. 2019. DOI: 10.1007/s11063-019-10049-1.
- [ 10 ] X. Deng, Y. Li, J. Weng, and J. Zhang, “Feature selection for text classification: A review,” *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3797-3816, Feb. 2019. DOI: 10.1007/s11042-018-6083-5.
- [ 11 ] N. Jiang, F. Tian, J. Li, X. Yuan, and J. Zheng, “MAN: mutual attention neural networks model for aspect-level sentiment classification in SloT,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2901-2913, Apr. 2020. DOI: 10.1109/JIOT.2020.2963927.



**Ya Chen**

was born in 1986. She is a Senior Engineer and a member of CCF. She received her B.S in 2009 from the School of Computer Science and Engineering of Guangxi Normal University, China, and her M.S. in 2018 from Wuhan University with a major is Computer Technology. Since 2020, she has been with the Department of Computer Engineering at Pai Chai University. Her current research interests include Internet of Things, machine learning, big data, pattern recognition and artificial intelligence. (ya4453565@gmail.com)



**Tan Juan**

received her B.S degree from the Department of Computer Science of Liaocheng University. In 2005, she received her M.S. degree from the Department of Information Engineering of Ocean University in China. From 2005 to 2020, she worked in the Department of Software Technology at Weifang University of Science and Technology in China, where she taught courses in computer science. She has been working in the Computer Engineering Department of Peicai University since 2020. Her current research interests include artificial intelligence, big data, deep learning, and blockchain.



**Hoekyung Jung**

received his M.S. degree in 1987 and Ph. D. degree in 1993 from the Department of Computer Engineering of Kwangwoon University, Korea. From 1994 to 1995, he worked for ETRI as a researcher. Since 1994, he is with the Department of Computer Engineering at Paichai University, where he works as a professor. His current research interests include multimedia document architecture modeling, information processing, embedded systems, machine learning, big data, and IoT.