INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 14(1), 124-134, February 2019.

Text Classification of Public Feedbacks using Convolutional Neural Network Based on Differential Evolution Algorithm

S. Zhang, Y. Chen, X.L. Huang, Y.S. Cai

Shuai Zhang, Yong Chen
School of Information
Zhejiang University of Finance and Economics
No.18 Xueyuan Street, Xiasha, Hangzhou 310018, China
zhangshuai@zufe.edu.cn, chenyong@zufe.edu.cn

Xiaoling Huang*
School of International Education
Zhejiang University of Finance and Economics
No.18 Xueyuan Street, Xiasha, Hangzhou 310018, China
*Corresponding author: huangxl@zufe.edu.cn

Yishuai Cai
School of Information
Zhejiang University of Finance and Economics
No.18 Xueyuan Street, Xiasha, Hangzhou 310018, China
caiyishuai@zufe.edu.cn

Abstract: Online feedback is an effective way of communication between government departments and citizens. However, the daily high number of public feedbacks has increased the burden on government administrators. The deep learning method is good at automatically analyzing and extracting deep features of data, and then improving the accuracy of classification prediction. In this study, we aim to use the text classification model to achieve the automatic classification of public feedbacks to reduce the work pressure of administrator. In particular, a convolutional neural network model combined with word embedding and optimized by differential evolution algorithm is adopted. At the same time, we compared it with seven common text classification models, and the results show that the model we explored has good classification performance under different evaluation metrics, including accuracy, precision, recall, and F1-score.

Keywords: public feedback, deep learning, text classification, convolutional neural network, differential evolution algorithm.

1 Introduction

In recent years, online feedback has played an increasingly important role in linking government and citizens. People can directly express their problems and opinions to the government departments by submitting their feedbacks online. It will help the government to better carry out their work. However, how to deal with hundreds of public feedbacks a day efficiently and deliver them to the appropriate departments is a big challenge for government administrators. Therefore, it is very necessary to explore a method to realize the automatic classification of public feedbacks, and to help government administrators improve their working efficiency and reduce their workload.

With the development of information technology, the field of text mining has attracted more and more scholars' attention [14, 20]. Feedback classification belongs to the text classification, which is an important branch of text mining. Text classification includes text segmentation, stopword removal, word vector representation, feature selection and classification [14]. In order to improve the accuracy of classification, several studies have improved the selection of text features and classifier [5,35]. Meanwhile, with the development of deep learning technology, convolutional neural network (CNN) [17] has shown excellent performance in solving many problems such as visual recognition [1], image recognition [10], and text classification [32].

In this study, we apply CNN to classify public feedbacks and assign them corresponding classes of labels, indicating which department should be responsible for the problems formulated in the public feedbacks. We obtained the public feedback data from a municipal government in China. The data was processed by distributed representation and word embedding method [15], and then using the CNN to automatically extract the deep text features and complete the classification of public feedbacks. To enhance the classification accuracy of the CNN, we adopt a heuristic optimization algorithm, differential evolution (DE) algorithm [25], for the selection and optimization of network parameters. In order to verify the effectiveness of the explored model, we compared the performance with other common text classification models, and the results show that the CNN has a good advantage in dealing with the classification of public feedbacks.

The remainder of this paper is organized as follow. Section 2 describes the related work on text classification and deep learning. Section 3 presents the content of the data, the way of text representation, the selected text classification model, the parameter optimization algorithm, and the corresponding performance evaluation metric. Section 4 analyzes and discusses the classification performance. In the last section of this paper, we made a summary and put forward some prospects for the future.

2 Related work

In recent years, data mining and analysis technology has been widely concerned and applied in various fields [6,7]. Among them, as an important part of data analytics, classification technology has been developed rapidly in text processing [8], image processing [34] and other fields. Specially, text classification is to classify text into predetermined categories according to the characteristic of the text, which has been applied in some fields. For example, Li et al. [19] proposed a hybrid classification model based on sentiment dictionary, support vector machine (SVM) and k-nearest neighbor (KNN) for sentiment classification of Chinese micro-blogs. Chau et al. [4] improved neural network algorithm to achieve multilingual text classification task. Sabbah et al. [27] proposed four modified frequency-based term weighting schemes, which is combined with common text classifiers such as SVM, KNN, naive Bayes (NB) and extreme learning machine, and tested in the text classification corpora. Liu and Peng [22] considered the statistics of positive and negative samples in the method of term frequency-inverse document frequency (TFIDF), and experiments were carried out in three real-world datasets. Nevertheless, these machine learning classification models only extract shallow text features, but cannot automatically and deeply extract text features hidden in context.

At the same time, deep learning technology has been applied to the field of text classification. Li et al. [21] used deep belief networks for the classification of web spam. Sun et al. [29] used a deep neural network model based on restricted boltzmann machine optimization to realize the sentiment classification of micro-blog text; its experimental results show that this model is more suitable to deal with the classification of short-length text than other traditional classification models, such as NB or SVM. In addition, CNN is also a popular deep learning algorithm. Unlike other deep learning algorithms, CNN has the characteristics of weight sharing and local perception [13]. These two characteristics make CNN show its excellent performance in the ability of spatial feature extraction and high-dimensional data processing. Because the distribution of text has some spatial correlation, CNN has been widely used to solve the problem of text classification. For instance, Gando et al. [12] used a deep CNN model to achieve the automatic classification of illustrations in photographs. To solve the problem of data sparseness in the process of text classification, Wang et al. [32] proposed a model of hybrid word embedding clustering and CNN, and its effectiveness is verified by two open benchmarks. In addition, in order to improve the performance of the neural network, Ijjina and Chalavadi [18] proposed using genetic algorithm to optimize the weights of network parameters in order to improve image classification accuracy. To improve the efficiency of model fitting, Trivedi et al. [31] proposed to optimize the parameters of CNN fully connected layer by using genetic algorithm.

As far as we know, there are few studies on improving the work efficiency of government administrators in handling the classification of public feedbacks. Accordingly, to ease the burden of government administrators, we applied a CNN model to automatically classify public feedbacks by combining word embedding and optimizing network parameters with DE algorithm.

3 Method

In this section, we will describe in detail the text classification model based on CNN used in this study. The flow chart of the model, as shown in Figure 1, including three modules: text representation, classifier training, and performance evaluation.

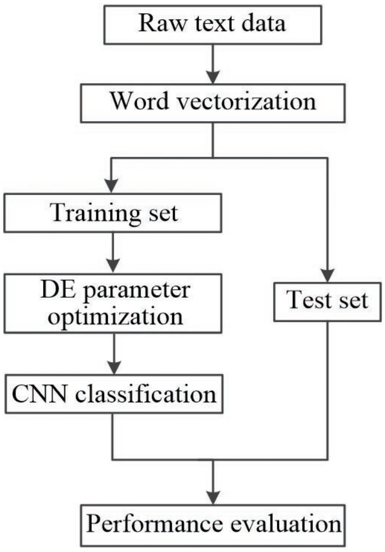

Figure 1: The flow chart of the text classification model

3.1 Data collection

In this paper, the dataset contains 4257 short-length texts, covering 22 categories, which is provided by a municipal government in China. The dataset is divided into three parts: training set, validation set, and test set, with the proportion of 80%, 10%, and 10%, respectively. The name of each category and the corresponding number of feedbacks are detailed in Table 1. Each feedback describes some suggestions or opinions written by the public to the relevant government departments. The meaning of each category is that the problem formulated in the feedback should be solved by the corresponding department in that category. In addition, it should be noted that each text has only one corresponding category.

Table 1: Names and feedback numbers of 22 categories

Category Name	**Number of Feedbacks**
Population and Family Planning Commission	42
Disabled Persons' Federation	29
Security Supervision Bureau	39
National Tax Bureau	43
City Management Enforcement Bureau	503
Electric Power Bureau	119
Housing Administration	382
Industrial and Commercial Bureau	152
Public Security Bureau	688
Sports Bureau	57
Bureau of Land and Resources	174
Environmental Protection Bureau	131
Commission for Discipline Inspection	98
Traffic Bureau	190
Education Bureau	401
Brigade Committee	66
Weather Bureau	56
Health Bureau	256
Pricing Bureau	507
Tobacco Bureau	112
Post Office	120
Court	92

3.2 Text representation and text classification model

In this section, we will describe the details of the text classification model. The structure of the model is shown in Figure 2.

The first part of the structure is the operation of text representation. In view of the fact that most words are meaningful to Chinese text classification as well as the ability of CNN to process high dimensional data, this study will not remove the stop words like general text classification. To reduce the dimension disaster of text representation, we will use word embedding method [15] to reduce the dimension of word vector space. Assuming that the vector dimension of each word is M, by using distributed representation method [15], map each word to a dictionary of common words with a capacity of 5000. The corresponding matrix size of a sentence will be 5000*M. However, we assume that a sentence has up to N words, and by word embedding we can reduce the matrix size to N*M, of which N is far less than 5000.

The second part of the structure is the operation of text classification. First, the above-mentioned word vector matrix is used as input of CNN model. By using 256 different convolution kernels and padding operation [23] for feature extraction, we can get 256 corresponding feature maps that have the same size as that before the convolutional operation. Second, in order to reduce the number of parameters of the network and prevent the overfitting issues, we took a max-pooling operation [23] on the mapping results. It should be mentioned that the max-pooling operation preserves the salient features extracted and discard the less important features, which can facilitate the fitting of the model. Then, by linking the results of the pooling with the neurons in the fully connected layer and using the rectified linear unit incentive function [24],

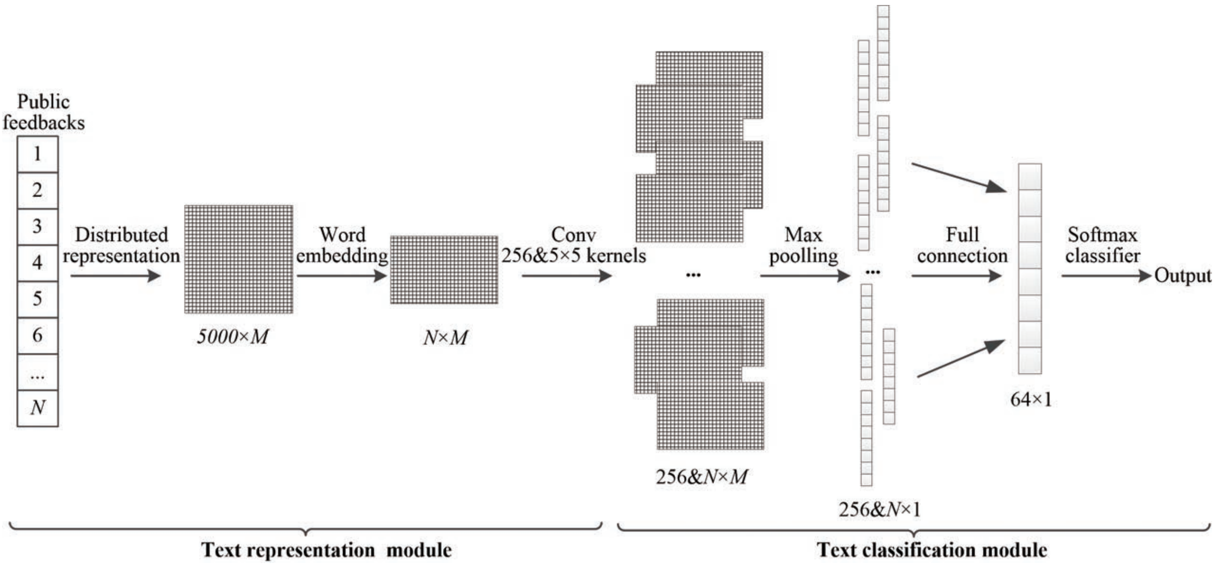

Figure 2: The structure of CNN classification model

the output vector is obtained. Finally, the output vector is put into the Softmax classifier [28] to get the classification results, which is the probability of classifying into each category. In this study, the size of the convolution kernels we use is 5*5, and the loss function is cross-entropy [2].

3.3 Parameter optimization

As we know, the structure of neural network and its training parameters have a great influence on the performance of network. The parameters include the size of convolution kernel, the number of neurons in the fully connected layers, the learning rate, etc. Therefore, in this paper, we will use the differential evolution algorithm [25] to optimize the parameters of convolution neural network. DE is a parallel direct search algorithm proposed by Price in 1996, which has strong global optimization capability. It includes four steps: initializing the population, mutation, crossover, and selection [25] . The implementation steps of the algorithm are shown in Table 2. A chromosome [25] consists of six genes, the number of neurons in the fully connected layer, the number of convolution filters, the size of batch, the values of dropout [23], and the learning rate, respectively. Among them, the first four values are integers and the last two values are decimal. In addition, the fitness function of DE algorithm is the error evaluation function of CNN during training.

3.4 Evaluation metric

In this study, we use four metrics to evaluate the effectiveness of the model used, including accuracy, precision, recall, and F1-score. The calculation of these metrics is shown in Equations (1-4).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

Table 2: The pseudocode of the DE algorithm

```
Initialize population size $P$, chromosome length $L$, crossover rate $CR$, and chromosomes
While (termination condition is not satisfied)
    For $i$ in range(0, $P$)
        Random selection of three chromosomes $x_a$ ,$x_b$, $x_c$, then perform the mutation operation and get the new chromosome $v_i$
        For j in range(0, $L$)
            If rand(0,1) $<$ $CR$ or rand(0,$L$) $==$ $j$
                Perform the crossover operation
            Else
                Do not perform crossover operation
            End if
            Get the new chromosome $u_i$
        End for
    End for
    For $i$ in range(0, $P$)
        If fitness($u_i$)$>$ fitness($x_i$)
            Update chromosome $x_i$
        Else
            Not update chromosome $x_i$
        End if
    End for
End while
The optimal fitness of the chromosome as the final output
```

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Where TP represents the numbers of correctly classified as the target category, TN represents the number of correctly classified as another category, FP represents the numbers of misclassified as target category, and FN represents the numbers of misclassified as another category. We can draw from the formula that the value of all the metrics is between 0 and 1, and the closer the value is to 1, the better the classification performance of the model is.

4 Computational experiments

This section will evaluate the performance of the model. We compare the CNN model based on DE optimization (CNN-DE) with the common text classification models to verify the effectiveness of the model. All experiments were developed under the Python version 3.6. The core configuration of the computer includes Inter Core i7-8700k processors with a 3.7GHz basic frequency and 32G running memory.

In order to verify the effectiveness of the CNN-DE model, this paper compares it with the commonly used seven text classification models in the same dataset. Among them, six models are combined by the basic classifier and TFIDF method. The classifier includes NB [33] decision tree (DT) [26], random forest (RF) [3], SVM [30], KNN [9], and gradient boost decision tree (GBDT) [11]. TFIDF is a common word frequency based vector representation method. DT, RF, and GBDT are based on tree structure for classification and forecasting, which has the advantages including simple structure and fast training speed. SVM is a supervised learning algorithm by constructing the maximal margin hyperplane, which is often used to solve the multi-dimensional nonlinear classification problem. KNN is an unsupervised learning algorithm, which can make classification according to the preset number of classes. In addition, the long short-term memory network (LSTM) [16] model does not adopt the TFIDF method, and the features are extracted automatically directly. LSTM is a deep neural network algorithm with memory function, which has often been used to solve the problem of sequence forecasting and classification. For ease of description, the comparison models are recorded as TFIDF-NB, TFIDF-DT, TFIDF-RF, TFIDF-SVM, TFIDF-KNN, TFIDF-GBDT, and LSTM, respectively.

In addition, all of these models are developed in a Spyder environment, where the TFIDF-NB, TFIDF-DT, TFIDF-RF, TFIDF-SVM, TFIDF-KNN, and TFIDF-GBDT models are implemented using the Sklearn toolkit, the CNN-DE and LSTM are implemented using open source deep learning framework Tensorflow 1.8.0.

The performance evaluation results of CNN-DE and other comparison models are shown in Figure 3. Table 3 lists the detailed evaluation results. We can find that the CNN-DE model outperforms with the other seven models under the same text dataset according to the four performance evaluation metrics used. The accuracy, precision, recall, and F1-score of CNN-DE model are 0.829, 0.830, 0.829, and 0.825, respectively. More specifically, according to the values of the accuracy, precision and recall, the CNN-DE model has improved 15.1%, 12.3%, and 15.1%, respectively compared to the TFIDF-DT model, and the F1-score of the CNN-DE model is 14.7% better than the TFIDF-KNN model. It is worth noting that the values of precision and recall of CNN-DE model are not only relatively high, but also the difference between the two is small compared with other models. It shows that the CNN-DE model has a good balance in dealing with text classification problems. This is because the CNN-DE model not only makes use of the word frequency information of text, but also automatically extracts the abstract semantic information, and then excavates the deeper text features to achieve better classification performance. At the same time, the DE algorithm optimizes the network parameters of CNN and further improves the performance of the model. However, for other comparative models, the classifiers can only extract the shallow text feature information, and the deep high-dimensional feature cannot be obtained. On the other hand, the word vector representation method based on TF-IDF is prone to cause dimension disaster problem, thus reducing the classification efficiency. In addition, the parameters of the models have been adjusted by repeated experiments, and there is still room for further improvement.

Furthermore, the evaluation results of CNN-DE model for each category are shown in Table 4, where 'Support' represents the number of test samples for each category. We can observe that the CNN-DE model has good classification performance under sufficient sample size. For example, the Post Office category has a test sample of 14 feedbacks, with a F1-score of 0.93.

Table 3: Detailed information on the evaluation results of each model

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
CNN-DE	0.829	0.830	0.829	0.825
LSTM	0.716	0.730	0.716	0.715
TFIDF-NB	0.728	0.762	0.728	0.717
TFIDF-DT	0.678	0.707	0.678	0.682
TFIDF-RF	0.739	0.783	0.739	0.744
TFIDF-SVC	0.700	0.829	0.700	0.710
TFIDF-KNN	0.683	0.720	0.683	0.678
TFIDF-GBDT	0.711	0.779	0.711	0.726

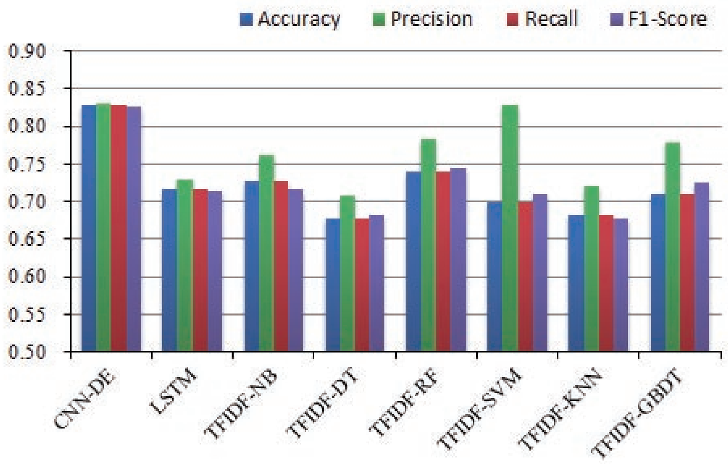

Figure 3: Performance comparison of the text classification models

Table 4: The evaluation results for each category of the CNN-DE model

Category Name	**Precision**	**Recall**	**F1-score**	**Support**
Population and Family Planning Commission	0.71	0.50	0.59	10
Disabled Persons' Federation	1.00	1.00	1.00	4
Security Supervision Bureau	1.00	1.00	1.00	1
National Tax Bureau	1.00	1.00	1.00	2
City Management Enforcement Bureau	0.85	0.89	0.87	53
Electric Power Bureau	1.00	0.81	0.90	16
Housing Administration	0.90	0.88	0.89	40
Industrial and Commercial Bureau	0.61	0.85	0.71	13
Public Security Bureau	0.72	0.81	0.76	59
Sports Bureau	0.00	0.00	0.00	3
Bureau of Land and Resources	0.80	0.84	0.82	19
Environmental Protection Bureau	0.86	0.86	0.86	22
Commission for Discipline Inspection	0.86	0.86	0.86	7
Traffic Bureau	0.77	0.48	0.59	21
Education Bureau	0.91	0.85	0.88	34
Tourism Committee	0.80	1.00	0.89	8
Weather Bureau	1.00	0.80	0.89	5
Health Bureau	0.81	0.84	0.82	25
Pricing Bureau	0.92	0.94	0.93	48
Tobacco Bureau	0.78	0.70	0.74	10
Post Office	0.93	0.93	0.93	14
Court	0.69	0.75	0.72	12

5 Conclusion and future work

In order to help government administrator improve the efficiency of handling public feedbacks and reduce the burden of work, we applied the CNN text classification model to automatically classify feedbacks. The model consists of text representation and text classification. Before training CNN classifier, to avoid the problem of dimension disaster caused by the excessive size of text input vectors, we used the word embedding method to reduce dimension. To improve the fitting ability of CNN, we used the DE algorithm to optimize the network parameters of CNN. To verify the effectiveness of the model, we compared the CNN-DE model with other seven common text classification models in terms of accuracy, precision, recall, and F1-score. The experimental results show that the CNN-DE model has better performance under each evaluation metric.

Meanwhile, this study still has some limitations that need to be addressed in future work. For example, the amount of data used in the model training is limited, and it has a certain influence on the fitting of the model. In the future, we will use a richer set of data to train model to improve its accuracy. In addition, we will improve the DE algorithm and explore more effective parameter optimization methods to optimize the CNN structure, such as genetic algorithm.

Acknowledgment

This work was supported by National Social Science Foundation of China (No. 17BGL047), National Natural Science Foundation of China (No. 51875503, No. 51475410).

Bibliography

[1] Bai, D.D.; Wang, C.Q.; Zhang, B.; et al. (2018); Sequence searching with CNN features for robust and fast visual place recognition, *Computers & Graphics*, 70, 270–280, 2018.

[2] Bishop, C.M. (1995); *Neural Networks for Pattern Recognition*, Oxford University Press, UK, 1995.

[3] Breiman, L. (2001); Random forests, *Machine Learning*, 45(1), 5–32, 2001.

[4] Chau, R.N.; Yeh, C.S.; Smith, K.A. (2005); A neural network model for hierarchical multilingual text categorization, *In Proceedings of the 2nd International Symposium on Neural Networks*, May 30–June 1, Chongqing, China, 238–245, 2005.

[5] Chen, J.N.; Huang, H.K.; Tian, S.F.; et al. (2009); Feature selection for text classification with Naive Bayes, *Expert Systems with Applications*, 36(3), 5432–5435, 2009.

[6] Dai, Y.; Wu, W.; Zhou, H.B.; et al. (2018); Numerical simulation and optimization of oil jet lubrication for rotorcraft meshing gears, *International Journal of Simulation Modelling*, 17(2), 318–326, 2018.

[7] Dai, Y.; Zhu, X.; Zhou, H.; et al. (2018); Trajectory tracking control for seafloor tracked vehicle by adaptive neural-fuzzy inference system algorithm, *International Journal of Computers Communications & Control*, 13(4), 465–476, 2018.

[8] Du, C.; Huang, L. (2018); Text classification research with attention-based recurrent neural networks, *International Journal of Computers Communications & Control*, 13(1), 50–61, 2018.

[9] Duda, R.O.; Hart, P.E. (1973); *Pattern Classification and Scene Analysis*, Wiley, USA, 1973.

[10] Ferreira, A.; Giraldi, G. (2017); Convolutional neural network approaches to granite tiles classification, *Expert Systems with Applications*, 84, 1–11, 2017.

[11] Friedman, J.H. (2001); Greedy function approximation: a gradient boosting machine, *Annals of Statistics*, 29(5), 1189–1232, 2001.

[12] Gando, G.; Yamada, T.; Sato, H.; et al. (2016); Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs, *Expert Systems with Applications*, 66, 295–301, 2016.

[13] Goodfellow, I.; Bengio, Y.; Courville, A. (2016); *Deep Learning*, The MIT Press, 2016.

[14] Gupta, V.; Lehal, G.S. (2009); A survey of text mining techniques and applications, *Journal of Emerging Technologies in Web Intelligence*, 1(1), 60–76, 2009.

[15] Hinton, G.E. (1986); Learning distributed representations of concepts, *In Proceedings of the 8th Annual Conference of the Cognitive Science Society*, August 15–17, Hillsdale, Canada, 1–12, 1986.

[16] Hochreiter, S.; Schmidhuber, J. (1997); Long short-term memory, *Neural Computation*, 9(8), 1735–1780, 1997.

[17] Hubel, D.H.; Wiesel, T.N. (1962); Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *The Journal of Physiology*, 160(1), 106–154, 1962.

[18] Ijjina, E. P.; Chalavadi, K.M. (2016); Human action recognition using genetic algorithms and convolutional neural networks, *Pattern Recognition*, 59, 199–212, 2016.

[19] Li, F.F.; Wang, H.T.; Zhao, R.C.; et al. (2017); Chinese micro-blog sentiment classification through a novel hybrid learning model, *Journal of Central South University*, 24(10), 2322–2330, 2017.

[20] Li, N.; Wu, D.D. (2010); Using text mining and sentiment analysis for online forums hotspot detection and forecast, *Decision Support Systems*, 48(2), 354–368, 2010.

[21] Li, Y.C.; Nie, X.Q.; Huang, R. (2018); Web spam classification method based on deep belief networks, *Expert Systems with Applications*, 96, 261–270, 2018.

[22] Liu, L.; Peng, T. (2014); Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF, *Journal of Information Science and Engineering*, 30(5), 1463–1481, 2014.

[23] Mou, L.C.; Ghamisi, P.; Zhu, X.X. (2018); Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, 56(1), 391–406, 2018.

[24] Nair, V.; Hinton, G.E. (2010); Rectified linear units improve restricted boltzmann machines, *In Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, June 21–24, Haifa, Israel, 807–814, 2010.

[25] Price, K.V. (1996); Differential evolution: a fast and simple numerical optimizer, *In Proceedings of the North American Fuzzy Information Processing Society*, June 19–22, New York, USA, 524–527, 1996.

[26] Quinlan, J.R. (1987); Simplifying decision trees, *International Journal of Man-machine Studies*, 27(3), 221–234, 1987.

[27] Sabbah, T.; Selamat, A.; Selamat, M.H.; et al. (2017); Modified frequency-based term weighting schemes for text classification, *Applied Soft Computing*, 58, 193–206, 2017.

[28] Socher, R.; Perelygin, A.; Wu, J.; et al. (2013); Recursive deep models for semantic compositionality over a sentiment treebank, *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, October 18–21, Seattle, USA, 1631–1642, 2013.

[29] Sun, X.; Li, C.C.; Ren, F.J. (2016); Sentiment analysis for Chinese microblog based on deep neural networks with convolutional extension features, *Neurocomputing*, 210, 227–236, 2016.

[30] Suykens, J.A.; Vandewalle, J. (1999); Least squares support vector machine classifiers, *Neural Processing Letters*, 9(3), 293–300, 1999.

[31] Trivedi, A.; Srivastava, S.; Mishra, A.; et al. (2018); Hybrid evolutionary approach for devanagari handwritten numeral recognition using convolutional neural network, *Procedia Computer Science*, 125, 525–532, 2018.

[32] Wang, P.; Xu, B.; Xu, J.M.; et al. (2016); Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification, *Neurocomputing*, 174, 806–814, 2016.

[33] Zhang, W.; Zhang, Z.; Chao, H.C.; et al. (2018); Kernel mixture model for probability density estimation in Bayesian classifiers, *Data Mining and Knowledge Discovery*, 32(3), 675–707, 2018.

[34] Zhang, W.; Zhang, Z.; Qi, D.; et al. (2014); Automatic crack detection and classification method for subway tunnel safety monitoring, *Sensors*, 14(10), 19307–19328, 2014.

[35] Zhou, Y.; Li, Y.W.; Xia, S.X. (2009); An improved KNN text classification algorithm based on clustering, *Journal of Computers*, 4(3), 230–237, 2009.