2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics

A Deep CFS Model for Text Clustering

Bin Lv, Weiliang Hou, Guohua Liu
Beijing Thunisoft Corporation Limited
Beijing, Chian
lb, houwl, liuguohua@thunisoft.com

Jing Gao, Xu Yuan, Peng Li, Zhikui Chen
School of Software Technology
Dalian University of Technology
Dalian, China
gaojing, david, zkchen@dlut.edu.cn
lipeng2015@mail.dlut.edu.cn

***Abstract*—With the fast development of the Internet technology, the court text information is collected from various fields at an unprecedented speed, such as Weibo and Wechat. This big court text information of high volume poses a vast challenge for the judge making reasonable decisions based on the vast cases. To cluster the reasonable assistant cases from the vast cases, we propose a deep CFS model for the text clustering, which can cluster the court text effectively, in this paper. In the proposed model, a robust deep text feature extractor is designed to improve the cluster accuracy , in which an ensemble of deep learning models are used to learn the deep features of the text. Furthermore, the CFS algorithm is conducted on the extracted deep text features, to discover the non-spherical clusters with the automatic find of the cluster centers. Finally, the proposed deep cluster model is evaluated on two typical datasets and the results show it can perform better than compared models in terms of the cluster accuracy.**

***Index Terms*—deep learning model, text clustering, the deep CFS model.**

I. Introduction

With the fast development of the Internet technology, the court text information is collected from various fields at an unprecedented speed, such as Weibo and Wechat [1] [2]. This big court text information contains many useful knowledge which can assist the judge to make decisions effectively and efficiently. Furthermore, it can help the junior judges to acquaint the experience [3-5]. However, it is a vast challenge for the judges to extract the similar court knowledge from those big court cases to make reasonable decisions based on those assistant knowledge [6]. Specially, there is no unified standard which can be used to distinguish the similar cases. On the other hand, the law permits the judges to make decisions based on their personal knowledge. Those two reasons lead the judges who have different law knowledge, case experience and value to make the different decisions for the same case. Furthermore, the fast increase of the number and the complexity of the case supplies the judge with more similar cases, which further makes it more difficult to assist the judge to make reasonable decisions [7]. Thus, how to effectively distinguish those complicated cases requires novel models and algorithms.

Clustering, an important method of the knowledge discovery, attempts to divide the unlabeled objects into different groups based on the similarity between the objects such that the objects with high similarity are classified to the same group while the objects with low similarity are in the different groups [8] [9]. Many clustering algorithms are introduced and those algorithms can be divided into several classes [10-13]. Specially, the distance-based clustering algorithms, such as the K-means and K-medoids assign the objects to the clustering center with the smallest distance[14] [15]. Typically, this kind of algorithms optimize the sum of the distance between the object and the pre-assigned clustering center. Although they achieve the state-of-the-art performance in various domains, such as the image segmentation and the text recognition, those algorithms cannot well cluster the non-spherical data. Another kind of clustering algorithms is based on the distribution of the objects [16]. The distribution-based methods aims to use an ensemble of well-designed probability distribution functions to represent the objects. However, the distribution-based algorithms cannot produce the high clustering accuracy without the well-fitting distribution functions. To solve the above problems, the density-based clustering algorithms are proposed [17]. The density-based algorithms can cluster arbitrary shape data, in which the similar objects of the same clustering center are located in the region of the high density and surrounded by the regions of the low density [18]. Although the density-base clustering algorithms can detect arbitrary shape data, it is a difficult to choose a reasonable cut-off density which has a vital effect on the cluster accuracy. To address the non-trivial selection of the cut-off density of the density-based algorithm, Alex proposed the CFS algorithm (Clustering by fast search and find of density peaks) [19].

In CFS, both the distance and density bases are adopted to detect the underlying patterns contained in the objects [19]. Specially, the distance basis is used to define the local density of the object, and two arbitrary objects with a short distance are in the same region with a high density. Then, based on the local density, the clustering distance is introduced, which can detect the clustering centers of the large local density and the large clustering distance. The CFS algorithm can cluster the arbitrary shape data objects with the automatic detection of the clustering centers and achieve the state-of-the-art performance in various application. However, there are still some drawbacks in CFS, such as the selection of the cut-off distance and the selection of the density peaks. To cluster the unlabeled objects where there are more than one density peaks in a cluster, Zhang et al. proposed an extended CFS (ECFS) algorithm [20]. In ECFS, the clustering process is composed of the stages, e.i., the pre-clustering and the merging-clustering.

978-1-5386-7975-3/18/$31.00 ©2018 IEEE
DOI 10.1109/Cybermatics_2018.2018.00054

In the pre-clustering stage, the CFS algorithm is conducted on the unlabeled data to produce the candidate cluster set. Differently, the cut-off distance of ECFS is more smaller to group the objects of the same local density in the cluster into sperate classes. In the merging-clustering stage, the similar classes are be merged into a class by measuring the similarity of the classes. ECFS can cluster the data with more density peaks, but the final number of the clustering center still needs to be assigned. To address the above problems, Jing et al. proposed an improved CFS algorithm [21]. Specially, to improve the robustness of CFS, the standard deviation is employed to compute the cut-off distance. Furthermore, the "bump" strategy is used to automatically detect the number of the class. At the same time, the merging clustering and splitting clustering are used to justify the cluster results. Those improved CFS algorithm can effectively improve the performance of the CFS algorithm, but they ignore the noise contained in the data, which has great effect on the clustering results.

This paper investigates a new method to discover the hidden patterns of the big court text to assist judges make decisions. To achieve that purpose, a deep CFS algorithm is proposed for the text classification. Specially, a robust deep text feature extractor is designed to improve the cluster accuracy by filtering the noises of the raw input. It is initialized by the layer-wise unsupervised learning strategy that the initial parameters of each layer are trained by the basic auto-encoder of the equal neurons re-constructing the input. Furthermore, an ensemble of deep learning models learn the features of the text by adopting the dropout method to improve the learning efficiency. Moreover, to group the big court text, the CFS algorithm is used to cluster the text. It can classify the non-spherical clusters with the automatical find of the cluster centers by computing the local density and the density-based distance. Finally, the proposed deep cluster algorithm is evaluated on a typical dataset and a real court text dataset. The results show it can perform better than compared models in terms of the cluster accuracy.

Thus, the contributions can be summarized into the three-fold aspects:

- To assist judges make decisions based on the big court text, a deep CFS algorithm is proposed for the text classification, in which the robust deep text feature extractor and the typical CFS algorithm are combined to discover the hidden patterns of the big court text. And extensive experiments are conducted to evaluate the performance of the proposed model.
- To learn the deep robust features of the text, a robust deep text feature extractor is designed by the layer-wise unsupervised learning and the dropout learning. In detail, the layer-wise unsupervised learning is used to obtain the initial parameters of each layer by re-constructing the input of each layer. After that, an ensemble of deep learning models are used to improve the learning of the text features.
- To group the big court text, the CFS algorithm is used to cluster the text which can discover the non-spherical clusters with the automatical find of the cluster centers, based on the distance between the data point.

The rest of this paper is organized as follows. Section II describes the robust deep text feature extractor. Section III introduces the deep CFS algorithm. The experiments of the proposed model is demonstrated in Section IV. Section V concludes the whole paper.

II. The Deep CFS Algorithm

In this section, we first introduce the robust deep text feature extractor which can effectively capture the intrinsic representation of the text contained noises. Then, the deep CFS model is introduced based on the above deep representation of the text, which can cluster the court text.

A. *The Robust Deep Text Feature Extractor*

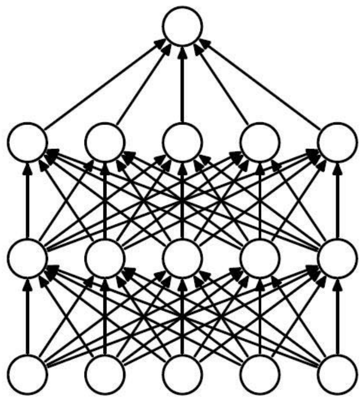
The features play an vital role in various text applications, such as the text classification and the text clustering [22] [23]. To obtain a set of good features of the text is a difficult work by the experts based on their experience [24] [25]. Especially, there are few text extractors designed for the court text. To address this problem, we introduce a deep text feature extractor based the deep learning which is an effective feature learning technique achieving the state-of-the-art performance in the images and audio domains [26]. Furthermore, to train a robust deep text feature extractor, the dropout technique is adopted, which uses the geometric average of the ensemble of standard deep text feature extractors as the output.

The dropout technique proposed by Hinton is an effective method that prevents the supervised neural network from over-fitting [27-29]. In dropout, each neuron is dropped randomly with a probability at each training phase, which amounts to sampling an ensemble of thinned deep models with different architectures trained on the different data. In the dropout, the geometric mean is used to approximate the predictions of those thinned deep models. Thus, the dropout method are adopted in the proposed text feature extractor, as shown in Fig. 1. The steps of proposed text feature extractor is composed of the feed-forward, back-propagation and prediction computations.

a) The feed-forward computations.

For a dropout neural network with H fully-connected layers, $h \in \{1, 2, \cdots, n_H\}$ denotes the index of the layer with h_0 and h_{n_H} representing the input layer and output layers, respectively. Let $z^{(h)}$ be the input vector of the layer h. $a^{(h)}$ denotes the output of the layer h. $W^{(h)}$ and $b^{(h)}$ are the weight matrix and the bias vector of the hidden layer h. The feed-forward pass of a dropout neural network is described in the following:

$$\begin{array}{l} s_i^{(\mathrm{h})} \sim Bernoulli\,(p) \\ \widetilde{\mathbf{a}}^{(l)} = \mathbf{s}^{(\mathrm{h})} * \mathbf{a}^{(l)} \\ z_j^{(l+1)} = \mathbf{w}_j^{(l+1)} \widetilde{\mathbf{a}}^{(l)} + b_j^{(l+1)} \\ a_j^{(l+1)} = f\left(z_i^{(l+1)}\right) \end{array}, \tag{1}$$



(a) The standard neural network (b) The dropout neural network

Fig. 1. The robust deep text extractor.

where $\mathbf{s}^{(\mathrm{h})}$ is the mask vector in which each element $s_i^{(\mathrm{h})}$ subjected to the Bernoulli distribution. The masked output vector $\widetilde{\mathbf{a}}^{(l)}$ can be obtained by the element-wise product of the mask vector and the standard activation vector. In other words, it is the thinned output vector in which the standard activation is set to be the zero with the probability of p.

b) The back-propagation computation. In the dropout back-propagation, the loss only passes through the selected architecture. The details of the dropout back-propagation are described as follow:

Step 1. $l = n_H$ is the output layer. Compute the loss term as follows:

$$\Delta\delta^{n_H} = (Y - A) \otimes f^{`}\left(Z^{n_H}\right). \tag{2}$$

δ^{n_H}, Y and A are the vector form of the loss term, the expected label and the activation, respectively.

Step 2. $l = n_{H-1}, \cdots, 3, 2$ is the hidden layer of the network. Compute the loss term in the following form:

$$\Delta\delta^{l} = W^{\mathrm{T}}\delta^{l+1} \otimes f^{`}\left(Z^{l}\right) \otimes S^{l}, \tag{3}$$

where W denotes the weight, S^l is the mask vector and $\otimes$ represents the element-wise product.

c) The predictions computation.

In the prediction phase, the whole architecture without dropout are used to compute the prediction value. And the weights of the model are multiplied by a "scale-up" factor to emulate the behavior of the ensemble of those thinned sub-networks. Thus, the output of the test phase is as follows:

$$\mathrm{O}_{itest}^{(l)} = a\left(\sum\nolimits_{j=1}^{s_{l-1}} p w_{ij}^{(l-1)} x_j^{(l)} + b_i^{(l)}\right). \tag{4}$$

s_{l-1} denotes the number of the neuron of the hidden layer $l-1$, X_i is the mask factor, and p represents the scale-up factor which guarantees that the expected output of each nodes under random dropout is the same as the output produced during pre-training.

To further improve the training efficiency of the robust deep text feature extractor, the pre-learning method is used to initialize the weights and biases of each layer before the dropout learning [30]. The pre-learning uses the unsupervised strategy to train each layer, in which each layer aims to reconstruct the input training objects by the the back-propagation algorithm. Thus, the details of the training of the robust deep text extractor are demonstrated in the Algorithm 1.

B. The CFS Algorithm

After obtaining the robust deep text features, the CFS algorithm is used to cluster the text into the corresponding categories by the fast search and find of density peaks. Specifically, the clusters are surrounded by the boundary with the lower local density and the cluster center is far from other cluster centers in the CFS algorithm. For example, given the deep representation of the dataset $X = \{x_1, x_2, \cdots x_n\}$, two quantities are computed to classify the text object into categories. The first quantity is the local density which aims to measure the distribution of the text objects. The local density is expressed as:

$$r_i = \sum_k x\left(s_{ij} - s_c\right), \tag{5}$$

where r_i denotes the local density of the test object i, and $x\left(s_{ij} - s_c\right)$ is computed as follows:

$$x\left(s_{ij} - s_c\right) = \left\{ \begin{array}{ll} 1 & if s_{ij} - s_c \leq 0 \\ 0 & otherwise \end{array} \right. \tag{6}$$

where s_{ij} represents the difference between the text object i and the text object j and the s_c is the cutoff distance which denotes the maximum difference in one class.

The second quantity is the distance between the text object and the other text object with a higher local density. It is expressed as:

$$\delta_i = \min_{j: r_j > r_i}\left(d_{ij}\right). \tag{7}$$

For the text object i with highest local density, δ_i is set to be the maximum distance between the text object i and other text object. Generally, the cluster center are objects with the large δ_i.

Algorithm 1: The deep text feature learning algorithm

Require: text objects $\left\{\left(X^i, Y^i\right)\right\}$, iterator, learning rate η, threshold
Ensure: weights and biases $\theta = \{W, b\}$
for $layer l = 2, 3 \cdots L$ **do**
compute the feed forward process of the pre-learning training:
for $input_l = 1, 2 \cdots N$ **do**
$z_{li} = W_{li} \cdot input_l + b_i$;
$o_{li} = f\left(z_{li}\right)$;
$z_{lo} = W_{lo} \cdot o_{li} + b_o$;
$o_{lo} = f\left(z_{lo}\right)$;
end for
compute the back-propagation process of the pre-learning training:
if $J_l > threshold$ **then**
$\sigma^{(lo)} = -\left(\mathrm{x} - o_{lo}\right) \cdot f'\left(z^{l_o}\right)$;
$\sigma^{(li)} = \left(W^{(l_o)}\right)^T \cdot \sigma^{(l_o)} \cdot f\left(z_{l_i}\right)$;
update the weights and biases:
$\Delta b = \Delta b + \Delta\sigma$;
$\Delta W = \Delta W + o \cdot \Delta\sigma$;
$b = b + \frac{1}{N}\Delta b$;
$W = W + \frac{1}{N}\Delta W$;
end if
end for
for $input_l = 1, 2 \cdots N$ **do**
compute the feed forward process of the dropout training:
for $layer l = 2, 3 \cdots L$ **do**
$z_{li} = W_{li} \cdot input_l + b_i$;
$o_{li} = f\left(z_{li}\right)$;
end for
compute the back-propagation process of the dropout training:
if $J_l > threshold$ **then**
for $layer l = 2, 3 \cdots L$ **do**
if $l = lastlayer$ **then**
$\sigma^{(lo)} = -\left(y - o_{lo}\right) \cdot f'\left(z^{lo}\right)$;
else
$\sigma^{(li)} = \left(W^{(l_o)}\right)^T \cdot \sigma^{(li+1)} \cdot f\left(z_{l_i}\right)$;
end if
end for
$\sigma^{(li)} = W \cdot \sigma^{(lo)} \cdot f'\left(z_{li}\right)$;
update the weights and biases:
$\Delta b = \Delta b + \Delta\sigma$;
$\Delta W = \Delta W + o \cdot \Delta\sigma$;
$b = b + \frac{1}{N}\Delta b$;
$W = W + \frac{1}{N}\Delta W$;
end if
end for

The details of the deep CFS algorithm is outlined in Algorithm 2.

Algorithm 2: The deep CFS algorithm

Require: deep text features $\{X^{(1)}, X^{(2)}, \cdots, X^{(N)}\}$, cut-off distance s_c
Ensure: Clustering results
for $i = 1, 2 \cdots N$ **do**
for $j = 1, 2 \cdots N\,(j \neq i)$ **do**
compute the distance s_{ij}:
$s_{ij} = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \cdots + \left(x_{in} - x_{jn}\right)^2}$;
end for
for $j = 1, 2 \cdots N\,(j \neq i)$ **do**
compute the local density r_i:
$r_i = \sum\limits_k x\left(s_{ij} - s_c\right)$;
$x\left(s_{ij} - s_c\right) = \begin{cases} 1 & if s_{ij} - s_c \leq 0 \\ 0 & otherwise \end{cases}$;
end for
for $j = 1, 2 \cdots N\,(j \neq i)$ **do**
compute the clustering distance δ_i:
if $r_i = r_{\max}$ **then**
$r_i = \max(s_{ij})$;
end if
$\delta_i = \min\limits_{j:r_j > r_i}\left(s_{ij}\right)$;
end for
the clustering centers $\{c_i\}$ are points with lager r_i and δ_i;
for $j = 1, 2 \cdots N\,(without\,\{c_i\})$ **do**
compute its centers $\{c_i\}$;
end for
end for

III. Experiments

A. Experiments on Fudan corpus

In this subsection, we assess the deep CFS algorithm on the Fudan corpus which is a typical Chinese text dataset [3]. This corpus contains 21 classes, such as art, education, and computer. In the experiment, we use 2815 texts grouped into 10 classes to evaluate the performance of the proposed model. The results are demonstrated in Fig. 2.

From the Fig.2, we can conclude two observations. The first observation is that the deep CFS algorithm can achieve higher clustering accuracy than that of the CFS algorithm in the most cases. Specially, the best clustering accuracy of the deep CFS algorithm for the fudan corpus is 83.8 % is much larger than that of the CFS algorithm. The reason is that the deep text feature extractor of the deep CFS algorithm can produce the effective feature. The clustering results produced by the deep CFS algorithm is more stable that those of the compared algorithm, since the deep text features used in the proposed algorithm is more robust that of the direct vector of the text. Those two observations illustrate the effectiveness of the deep CFS algorithm.

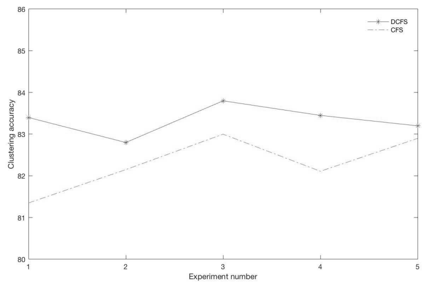


Fig. 2. The results on fudan corpus.

B. *Experiments on real text corpus*

In this subsection, we evaluate the proposed algorithm on the real text corpus which is collected from Huayu company. This corpus is composed of 4000 objects which can be classified into 4 classes, such as transportation, fight, and swindle. The clustering results are shown in the Fig. 3.

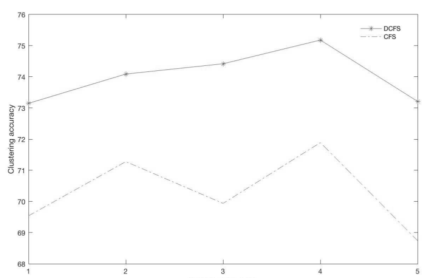


Fig. 3. The results on real corpus.

From the Fig. 3., we can obtain the similar observations. In other words, the clustering accuracy of the deep CFS algorithm is higher than that of the CFS algorithm. More specifically, the worst clustering accuracy produced by the proposed deep algorithm at the first experiment is higher than the best clustering accuracy produced by the CFS algorithm at the fourth experiment. Another observations is that the results of the real text dataset are lower than those of the representative text dataset, because there are more noises in the real dataset than the representative text dataset. However, the deep CFS algorithm still performs better than the CFS algorithm on the real text dataset, indicating the effectiveness of the proposed model.

IV. CONCLUSION

In this paper, an deep CFS model is designed to cluster the big text data. One notable advantage of the proposed model is to effectively implement the clustering of the big text data by combining the new deep robust text extractor and the CFS clustering algorithm. The new deep robust text extractor is used to learn the effective representation of the big text by the unsupervised pre-training learning and the supervised dropout learning. The CFS clustering algorithm discovers the non-spherical clusters with the automatical find of the cluster centers, grouping the text objects into different classes based on the similarity. The experimental results clearly demonstrate that the proposed model can perform better than the CFS algorithm in terms of the clustering accuracy, proving its potential for the clustering of the big text data.

Although the deep CFS model can cluster the big text of noises, it is still not efficient enough for the clustering of the big text without the density peak. To improve the performance of the deep CFS algorithm, new strategies for the find of clustering centers will be used to automatically decide the clustering centers.

V. ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program of China under Grant No. 2016YFC0800802.

REFERENCES

[1] R. Nikhil, N. Tikoo, S. Kurle, H. S. Pisupati, and G. Prasad, "A Survey on Text Mining and Sentiment Analysis for Unstructured Web Data," *International Journal of Emerging Technologies and Innovative Research*, vol.2, no 4, pp. 1292-1296, 2015.

[2] A. Skabar and K. Abdalgader, "Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no 1, pp. 1041-4347, 2013.

[3] B. Lv and W. You, "Typical Case Recommendation of Court Texts Based on Topic Model," *Microelectronics* & *Computer*, vol. 35, no 2, pp. 128-132, 2018.

[4] J.Gao, J.Li, and Y.Li, "Approximate event detection over multi-modal sensing data," *Journal of Combinatorial Optimization*, vol. 32, pp. 1002-1016, 2016.

[5] J. Gao, J. Li, Z. Cai, and H. Gao, "Composite event coverage in wireless sensor networks with heterogeneous sensors," in *Proceedings of 2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 217-225.

[6] Y. Li, A. Algarni, M. Albathan, Y. Shen, and M. A. Bijaksana, "Relevance Feature Discovery for Text Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no 6, pp. 1656-1669, 2015.

[7] C. K. Emani, N. Cullot, and C. Nicolle, "Understandable Big Data: A survey," *Computer Science Review*, vol. 17, pp. 70-81, 2015.

[8] Q. Zhang, C. Zhu, L. T. Yang, Z, Chen, L, Zhao, and P. Li, "An Incremental CFS Algorithm for Clustering Large Data in Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1193-1201, 2017.

[9] P. Li, Z. Chen, L. T. Yang, L. Zhao, and Q. Zhang, "A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing," *Neurocomputing*, vol. 256, pp. 82-89, 2017.

[10] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678, 2005.

[11] H. P. Kriegel, P. Kroger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1:, 2009.

[12] P. Li, Z. Chen, Y. Hu, Y. Leng, and Q. Li, "A Weighted Fuzzy c-Means Clustering Algorithm for Incomplete Big Sensor Data," in *procceding of China Conference on Wireless Sensor Networks*, 2017, pp. 55-63.

[13] F. Bu, Z. Chen, P. Li, T. Tang, and Y. Zhang, "A High-Order CFS Algorithm for Clustering Big Data," *Mobile Information Systems*, vol. 2016, pp. 4356127:1-8, 2016.

[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881-892, 2002.

[15] H. Park and C. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336-3341, 2009.

[16] E. Bocchieri and B. K. W. Mak, ”Subspace distribution clustering hidden Markov model,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 264-275, 2001.

[17] M. Ester, H. Kriegel, J. Sander, and X. Xu, ”A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD96)*, 1996, pp. 226231.

[18] K. Mumtaz1 and K. Duraiswamy, ”A Novel Density based improved k-means Clustering Algorithm Dbkmeans,” *International Journal on Computer Science and Engineering*, vol. 2, no. 2, pp. 213-218, 2010.

[19] A. Rodriguez and A. Laio, ”Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014.

[20] W. Zhang and J. Li, ”Extended Fast Search Clustering Algorithm: Widely Density-Clusters, No Density Peaks,” in *Proceedings of the 3rd International Conference on Database and Data Mining*, 2015, pp. 1-18.

[21] J. Gao, L. Zhao, Z. Chen, P. Li, H. Xu, and Y. Hu, ”Icfs: An improved fast search and find of density peaks clustering algorithm,” in *Proceedings of the 14th International Conference on Pervasive Intelligence and Computing*, 2016, pp. 537-543.

[22] Y. LeCun, Y. Bengio, and G. Hiton, ”Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[23] Q. Zhang, L. T. Yang, Z. Chen, and Peng Li, ”A survey on deep learning for big data,” *Information Fusion*, vol. 42, pp. 146-257, 2018.

[24] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, ”Deep convolutional computation model for feature learning on big data in Internet of Things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790-798, 2018.

[25] Q. Zhang, L. T. Yang, Z. Chen, ”Deep computation model for unsupervised feature learning on big data,” *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 161-171, 2016.

[26] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, ”Deep learning for visual understanding: A review,” *Neurocomputing*, vol. 187, pp. 27-48, 2016.

[27] G. Yarin and Z. Ghahramani, ”Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 1050-1059.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[29] X. Bouthillier, K. Konda, P. Vincent and R. Memisevic, ”Dropout as Data Augmentation,” 2016, arXiv:1506.08700v4.

[30] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, ”An Adaptive Dropout Deep Computation Model for Industrial IoT Big Data Learning with Crowdsourcing to Cloud Computing,” *IEEE Transactions on Industrial Informatics*, 2018, DOI: 10.1109/TII.2018.2791424.