Research Article
An Active Learning Method Based on Variational Autoencoder and DBSCAN Clustering

Fang Chen, Tao Zhang, and Ruilin Liu
School of Information and Mathematics, Yangtze University, Jingzhou 434023, China

Correspondence should be addressed to Tao Zhang; tzhang@yangtzeu.edu.cn
Received 11 March 2021; Revised 5 July 2021; Accepted 20 July 2021; Published 2 August 2021

Active learning is aimed to sample the most informative data from the unlabeled pool, and diverse clustering methods have been applied to it. However, the distance-based clustering methods usually cannot perform well in high dimensions and even begin to fail. In this paper, we propose a new active learning method combined with variational autoencoder (VAE) and density-based spatial clustering of applications with noise (DBSCAN). It overcomes the difficulty of distance representation in high dimensions and prevents the distance concentration phenomenon from occurring in the computational learning literature with respect to high-dimensional p-norms. Finally, we compare our method with four common active learning methods and two other clustering algorithms combined with VAE on three datasets. The results demonstrate that our approach achieves competitive performance, and it is a new batch mode active learning algorithm designed for neural networks with a relatively small query batch size.

1. Introduction
In practical issues, the amount of labeled data is relatively small and the vast majority is unlabeled. It is often unrealistic to consume a lot of human resources and expensive cost to annotate unlabeled data for the annotation budget is limited. Active learning [1] is born out of this problem, which is one field trying to address the difficulties in data labeling. It assumes that the collection of data is relatively easy, but the labeling process is costly. It tackles the problem of which samples we should label to result in the highest improvement in test accuracy under a fixed labeling budget. Existing active learning algorithms are mainly divided into two categories, query-synthesizing [2–4] and query-acquiring [5]. Query-synthesizing approaches use generative models to generate informative samples whereas query-acquiring algorithms use different sampling strategies to determine how to select the most informative samples.

In this paper, we mainly pay attention to query-acquiring methods. One of the query-acquiring algorithms is uncertainty-based method [6, 7]. Settles [8] mentioned that the classifier gives every unlabeled data a probability score which represents the uncertainty belonging to its class, and then we choose the data with the highest query batch size.
Therefore, we adopt VAE in our approach to solve the difficulty of distance representation especially for digital images [18]. VAE plays a great role in our method, which perfectly represents high-dimensional data into the low-dimensional representation in latent space, preventing the occurrence of the phenomenon of p-norm distance concentration of high-dimensional data. And in common clustering approaches [19–22], DBSCAN [23] is good at identifying the noise and discovering arbitrarily shaped clusters without knowing the number of cluster classes to be formed in advance. Therefore, we propose a new active learning strategy combined with VAE and DBSCAN clustering for considering their advantages. What is more, excellent active learning models are usually based on large query batch sizes. But they may ignore the performance of a relatively small query batch size based on small sample query batch sizes. We motivate our method with a simple idea. First, the experimental results based on a relatively small query batch size are provided in Section 4. In the end, we draw a conclusion in Section 5.

2. Related Notion

2.1. Problem Definition. We define the problem of active learning formally as follows. Given a labeled pool \((X_L, Y_L)\) and a much larger unlabeled pool \(X_U\), we are aimed to sample the most informative unlabeled data \(x_U\) from \(X_U\) by iteratively querying a fixed sampling budget in order to train the most label-efficient model in an active learning task. During this progress, \(n\) number of unlabeled data will be sampled by an acquisition function and be annotated by the oracle. This is an iterative process which is done until a certain stopping criterion is reached, such as the desired amount of samples or test accuracy.

2.2. Related Methods. With respect to VAE, the parameters are trained by two loss functions if \(N\) training samples are given. One is the reconstruction loss \(L_1\), which forces the reconstructed sample \(\hat{x}_i\) to match the original input sample \(x_i\). Here, we use the cross entropy for measurement:

\[
L_1 = -\sum_{i=1}^{N} \left[ x_i \cdot \log \hat{x}_i + (1 - x_i) \cdot \log \left(1 - \hat{x}_i\right) \right], \quad \text{where } i = 1, \ldots, N. \tag{1}
\]

In addition, we are aimed to make \(\hat{x}_i\) as close as possible to \(x_i\) itself. That is to say, it can retrieve as much of the original information as possible through decoding. The other one is regularization loss \(L_2\), which helps to learn latent space with good structure and reduces overfitting on the training data. The formula is shown as follows:

\[
L_2 = \frac{1}{N} \sum_{i=1}^{N} \left[ 1 + \log (\sigma_i^2) - \mu_i^2 - \exp\left(\log (\sigma_i^2)\right) \right], \quad \text{where } i = 1, \ldots, N. \tag{2}
\]

In our approach, clusters in DBSCAN are defined as the largest set of points connected by density. A region with sufficient density can be divided into a cluster, and points that are not clustered are called noise. The basic process of DBSCAN is as shown in Figure 1.

3. Proposed Model

3.1. The Combination of VAE and DBSCAN Clustering. In this paper, we suggested a new active learning method based on VAE and DBSCAN clustering on a relatively small query batch size. We motivate our method with a simple idea. First, VAE learns a valid low-dimensional latent feature space for the underlying distribution using a Gaussian prior from the labeled and unlabeled pool, whose space is a mixture of the latent features. Then we adopt density-reachable clustering DBSCAN to remove noise in the initial clusters, and we sample the most valuable unlabeled data in high density also
in different clusters from the latent space. The framework of our model is shown in Figure 2.

In detail, the VAE model in our experiment is combined by a convolutional neural network and a deconvolutional neural network. The convolutional one named encoder consists of four convolutional layers, one flatten layer, and three fully connected layers in order. And the deconvolutional one named decoder is relatively simple, consisting of a fully connected layer and two convolutional layers. We update VAE by descending stochastic gradients. After iteration and updating, we obtain the trained parameters \( \theta_{\text{VAE}} \) also the trained model VAE. In the end, VAE learned a two-dimensional latent feature space efficiently.

The low-dimensional feature vectors learned by VAE serve as the input data for the following clustering methods. In DBSCAN, we start sampling unlabeled points in the clustered classes after an initial unlabeled core object is computed. Here, we note that we remove the noise at the first step. Suppose the needed amount of unlabeled data is \( C \), the total number of all unlabeled density-reachable points in all clusters by DBSCAN is required to be as close to \( C \) as possible. The purpose is to ensure that different types of high-density unlabeled data can be retrieved as much as possible. After that, \( C \) number of corresponding original unlabeled data will be sampled by an acquisition function and to be annotated by Oracle. Then put them into the original labeled pool and use the task learner to test the mean accuracy of our proposed model.

### 3.2. Algorithms

For clarity, we describe our method in Algorithms 1 and 2.

### 4. Experiments

#### 4.1. Dataset and Task Module

We evaluated our method on three typical image datasets MNIST [24, 25], Fashion-MNIST, and CIFAR-10. The first one MNIST contains 60,000/10,000 (train/test) handwritten digit images (0–9) with the resolution of 28 × 28. And Fashion-MNIST is an alternative version of MNIST which has the same size as MNIST with frontal images of different items from 10 categories such as shirts, pants, sandals, and bags. They are also the single channel images with the same size of 28 × 28.

CIFAR-10 contains 10 categories of images such as airplanes, automobiles, birds, and cats. Different from the two datasets above, images in CIFAR-10 are 3-channel color RGB images with the size of 32×32. Not only the noise is very large, but also the proportion and characteristics of the object are not the same, which brings great difficulties for recognition.

We used the classic LeNet architecture [26] as our task module for MNIST and Fashion-MNIST, while for CIFAR-10 task we used VGG-16 (Simonyan and Zisserman, 2014) as our architecture. And we took a simple convolutional network just similar to LeNet architecture as the second task module when comparing the different clustering active learning algorithms. This module consists of three convolutional layers, two maximum pooling layers, a flatten layer, and two fully connected layers in order.

#### 4.2. Baseline Algorithms

To demonstrate the effectiveness of our method, we took the following four common active learning methods and also the random sampling as the baseline algorithms. They are briefly described as follows:

(i) Random: the query batch is chosen uniformly at random
(ii) Uncertainty: uncertainty sampling with minimal top confidence
(iii) Core Set: in every round of query, sample the unlabeled data which are of the largest distance from the labeled set and then add to the labeled pool
(iv) EGL: estimated gradient length
(v) Bayesian: Bayesian uncertainty sampling with minimal top confidence

Furthermore, in order to comparing the performance of different clustering algorithms [27] in active learning task, we replaced DBSCAN by two classic and widely used clustering algorithms: K-means [28] and Mean shift [29]. And they are also compared with random sampling. These clustering methods in our experiment were based on a two-dimensional feature dataset learned by VAE which are extracted from the three datasets listed above. Their brief description is as follows:
K-means with VAE: after VAE, we sample the two-dimensional data around the cluster centers according to K-means clustering rule and a given radius.

Mean shift with VAE: after VAE, solve for a vector that moves the center of the circle in the direction with the highest density of the dataset, and the average position of the points inside the circle is found as the new center position in each iteration. We choose the points around these centers.

DBSCAN with VAE: our method, detailed above.

4.3. Implementation Details. In our VAE model, the learning rate was selected by default as \( \alpha = 1 \times 10^{-3} \). What is more, we chose Sigmoid as our optimizer. Based on a relatively small query batch size, it shows that when epochs reach about 400, the learning loss almost does not decrease and the distribution of latent feature space learned by VAE comes to be stable. In other words, we set the number of training epochs to 400 to prevent overfitting. Figure 3 shows the two-dimensional feature points learned by VAE in different epochs. After VAE is trained, a certain number of low-
dimensional data samples were extracted by its well-trained encoder. We checked some corresponding original images as shown in Figure 4 (taking Fashion-MNIST as an example).

In order to observe the performance of active learning algorithms on a relatively small query batch size, we tried to sample 100 low-dimension data points randomly from the latent space learned by encoder in VAE as our clustering dataset for DBSCAN, including 10 labeled data and 90 unlabeled data, and our query batch size is set to 5 which is relatively small. Finally, we obtained the needed amount of unlabeled density-reachable points in different clusters after removing the noise. It is particularly noted that we adjust the parameters to make the total number of unlabeled points in all density-reachable clusters to be as close as possible to the needed amount of data for annotations.

4.4. Results. In our experiment, we used the test accuracy as our metric to evaluate the performance. The results are averaged from 20 runs in order to ensure statistical validity. By the results in Table 1, we plot the test accuracy of experiments using four different active learning methods on MNIST in Figure 5. The results demonstrate that, in MNIST, our method, DBSCAN with VAE, performs better on small query batch size than the active learning methods listed above. Figure 5 also shows that some methods perform on par or worse than random sampling. This may be the reason that the sample size is relatively small and the randomness is strong, which reduces the gap between each sampling method and random sampling. It may also be because some methods are more suitable for large query batch sizes.

We also notice that, except for EGL performing well than random sampling on very small batch sizes below 20, most algorithms listed above including random sampling consistently outperform EGL. As for EGL, a possible explanation for the discrepancy between these results may be the architecture used for the different tasks because EQL uses the gradient relative to the model parameters as the score function [12]. Among all the methods above, Bayesian performs the poorest and it may be because Bayesian is more suitable for large training datasets.

Next, we did a comparison test on CIFAR-10. The result is in Table 2 and we plot it in Figure 6.

As can be seen from Figure 6 on CIFAR-10, Core Set often performs worse than random sampling, and it is also possibly because the random advantage is greater than Core Set when the sample size is small. And when the size is below 25, our approach shows little advantage than other methods, but our approach leads all the way obviously when the number of labeled samples is over 25.

Furthermore, we replaced DBSCAN by K-means and Mean shift in order to study the different effects caused by different clustering methods. We also combined them with VAE and did experiment on MNIST and Fashion-MNIST. Detailed experimental results are listed in Tables 3 and 4, respectively. And we plot their performance in Figures 7 and 8.

However, what we may notice that, in Figure 7, the result of K-means with VAE is better than our method when the number of labeled data reaches 30 on MNIST. DBSCAN and K-means perform similarly until size 35 and DBSCAN performs clearly better after size 40. The reason may be that, in this quantity case, a great part of samples sampled by K-means method are closer to the potential characteristics of the labeled data than the data sampled by DBSCAN. In other side, this situation is probably caused by the uneven distribution of the samples when mapping the high-dimensional image data into the two-dimensional latent space. After size 40, DBSCAN may be more sensitive to detect the outlier points in latent space and prevent them being clustered. So the performance of DBSCAN is better than that of K-means when size is greater than 40. Of course, this is a hypothetical explanation. Probably, it is difficult to know for sure why K-means starts performing worse after size 40. However, on CIFAR-10 in Figure 8, compared with random sampling and the other two clustering methods, DBSCAN with VAE behaves better and the performance of increasing accuracy is more stable. It is somewhat clear from the results that our proposed method performs well on the whole on MNIST and Fashion-MNIST.

In order to show our experimental results more significantly, the improvement percentage of different clustering methods over random sampling is listed in Tables 5 and 6, and we plot their performance in Figures 9 and 10.
Figure 4: Some original images corresponding to the data points in latent space.

Table 1: Mean accuracy (%) of different methods on MNIST from 20 runs.

| Number of labeled examples | 15  | 20  | 25  | 30  | 35  | 40  | 45  | 50  |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| DBSCAN with VAE           | 49.76 | 58.51 | 59.67 | 60.90 | 64.38 | 69.02 | 69.68 | 71.28 |
| Random                    | 30.19 | 43.05 | 52.34 | 51.68 | 57.30 | 64.03 | 60.75 | 64.96 |
| Uncertainty               | 37.74 | 48.29 | 44.15 | 59.35 | 61.41 | 67.69 | 61.70 | 68.59 |
| Core Set                  | 49.03 | 48.03 | 54.51 | 54.99 | 56.34 | 58.89 | 53.97 | 60.13 |
| EGL                       | 44.73 | 46.80 | 41.27 | 52.37 | 49.89 | 51.49 | 57.05 | 59.73 |
| Bayesian                  | 32.33 | 31.50 | 35.30 | 37.59 | 39.52 | 45.13 | 54.41 | 56.21 |

Figure 5: Mean accuracy (%) of different active learning methods on MNIST.
### Table 2: Mean accuracy (%) of different methods on CIFAR-10 from 20 runs.

| Number of labeled examples | DBSCAN with VAE | Random | Uncertainty | Core Set | EGL | Bayesian |
|-----------------------------|----------------|--------|-------------|----------|-----|----------|
| 15                          | 11.19          | 11.10  | 10.22       | 9.70     | 11.21| 8.61     |
| 20                          | 11.84          | 11.50  | 11.87       | 10.43    | 9.41 | 10.34    |
| 25                          | 13.48          | 10.17  | 12.87       | 11.17    | 13.71| 12.66    |
| 30                          | 17.25          | 11.60  | 12.92       | 10.32    | 15.09| 13.01    |
| 35                          | 18.85          | 12.63  | 13.91       | 10.82    | 16.09| 13.15    |
| 40                          | 18.81          | 12.40  | 13.95       | 13.28    | 15.01| 14.91    |
| 45                          | 20.75          | 16.64  | 15.56       | 12.36    | 19.02| 17.61    |
| 50                          | 20.98          | 18.27  | 20.24       | 13.65    | 18.31| 18.12    |

### Figure 6: Mean accuracy (%) of different active learning methods on CIFAR-10.

![Mean accuracy on CIFAR-10](image)

### Table 3: Mean test accuracy (%) of different clustering methods on MNIST.

| Number of labeled examples | Random (%)  | K-means with VAE (%) | Mean shift with VAE (%) | DBSCAN with VAE (%) |
|---------------------------|-------------|----------------------|-------------------------|---------------------|
| 15                        | 44.6        | 48.36                | 48.34                   | 49.76               |
| 20                        | 47.2        | 57.77                | 49.97                   | 58.51               |
| 25                        | 54.56       | 58.77                | 56.99                   | 59.67               |
| 30                        | 55.42       | 61.68                | 57.38                   | 60.90               |
| 35                        | 60.27       | 64.09                | 63.34                   | 64.38               |
| 40                        | 65.17       | 65.50                | 64.95                   | 69.02               |
| 45                        | 67.53       | 66.57                | 68.83                   | 69.68               |
| 50                        | 68.04       | 69.24                | 69.18                   | 71.28               |

### Table 4: Mean test accuracy (%) of different clustering methods on Fashion-MNIST.

| Number of labeled examples | Random (%)  | K-means with VAE (%) | Mean shift with VAE (%) | DBSCAN with VAE (%) |
|---------------------------|-------------|----------------------|-------------------------|---------------------|
| 15                        | 9.49        | 10.56                | 9.52                    | 11.19               |
| 20                        | 9.80        | 10.74                | 10.32                   | 11.84               |
| 25                        | 10.49       | 12.27                | 10.67                   | 13.48               |
| 30                        | 11.61       | 15.10                | 13.58                   | 17.25               |
| 35                        | 13.58       | 17.48                | 13.27                   | 18.85               |
| 40                        | 14.62       | 17.12                | 15.98                   | 18.81               |
| 45                        | 14.43       | 16.22                | 18.37                   | 20.75               |
| 50                        | 15.26       | 18.35                | 18.46                   | 20.98               |
MNIST, our method DBSCAN with VAE, achieves the best results when the size of labeled samples reaches 20, increasing the accuracy by 11.30% over random sampling. Generally speaking, our approach performs even better in the other labeled data quantities. However, on Fashion-MNIST, our approach performs clearly better than the other two clustering methods at every labeled size in our experiment. And it achieves the best improvement of 6.32% when the labeled samples size reaches 45.

In the end, we computed the average accuracy improvement of each clustering method combined with VAE to see their overall improvement over random sampling, and we plot them in Figure 11. On MNIST, K-means and Mean shift achieve the improvement of 3.26% and 1.86%, respectively. And our method achieves the best performance of 4.48%. And on Fashion-MNIST, the three clustering active learning methods achieve the improvement over random sampling by 3.23%, 1.36%, and 4.23% in order. In a word, our proposed algorithm, DBSCAN with VAE, finally shows its superiority than K-means and Mean shift when combined with VAE on a relatively small query batch size on MNIST and Fashion-MNIST.

### Table 5: Accuracy improvement percentage over random sampling on MNIST.

| Number of labeled examples | K-means (%) | Mean shift (%) | DBSCAN with VAE (%) |
|---------------------------|-------------|----------------|---------------------|
| 15                        | 3.72        | 3.70           | 5.12                |
| 20                        | 10.56       | 2.76           | 11.30               |
| 25                        | 4.21        | 2.43           | 5.11                |
| 30                        | 6.42        | 1.96           | 5.48                |
| 35                        | 3.82        | 3.07           | 4.11                |
| 40                        | 0.33        | −0.22          | 3.85                |
| 45                        | −0.96       | 1.30           | 2.15                |
| 50                        | 1.20        | 1.77           | 3.24                |

### Table 6: Accuracy improvement percentage over random sampling on Fashion-MNIST.

| Number of labeled examples | K-means (%) | Mean shift (%) | DBSCAN with VAE (%) |
|---------------------------|-------------|----------------|---------------------|
| 15                        | 1.07        | 0.03           | 1.70                |
| 20                        | 0.94        | 0.52           | 2.04                |
| 25                        | 1.78        | 0.18           | 2.99                |
| 30                        | 3.49        | 1.97           | 6.44                |
| 35                        | 3.90        | −0.31          | 5.27                |
| 40                        | 2.50        | 1.36           | 4.19                |
| 45                        | 1.79        | 3.94           | 6.32                |
| 50                        | 3.09        | 3.20           | 5.72                |
Figure 8: Mean test accuracy (%) of different clustering methods on Fashion-MNIST.

Figure 9: Accuracy improvement percentage over random sampling on MNIST.
5. Conclusion and Discussion

In this paper, we proposed a new active learning method based on VAE and DBSCAN designed for neural networks with a relatively small query batch size. It overcomes the difficulty of distance representation in high dimensions and prevents the distance concentration phenomenon from occurring in the computational learning literature with respect to high-dimensional p-norms. Based on the results on MNIST, Fashion-MNIST, and CIFAR-10, we empirically show that our method achieves competitive performance compared with four common active learning methods listed above, and it is also superior to the other two clustering active learning methods for image classification when the query batch size is relatively small. For active learning models are usually based on a relatively large query batch sizes, our approach of small query batch sizes can be regarded as a supplement to the previous research studies. In addition, our method is simple to implement and can conceptually be extended to other domains, and we see it as a welcome addition to the arsenal of methods in use today. Furthermore, the inadequacy of our method is that manual parameter adjustment is required to make the total number of unlabeled points in all density-reachable clusters to be as close as possible to the needed amount of data for annotations in DBSCAN. And we will still thoroughly study the related work presented in this paper in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (61673006).

References

[1] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” Journal of Artificial Intelligence Research, vol. 4, pp. 129–145, 1996.
[2] D. Mahapatra, B. Bozorgtabar, J.-P. Thiran, and M. Reyes, “Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 580–588, Springer, Granada, Spain, September 2018.
[3] C. Mayer and T. Radu, “Adversarial sampling for active learning,” 2018, https://arxiv.org/abs/1808.06671v2.
[4] M. Prem and J. Raymond, “Diverse ensembles for active learning,” in Proceedings of the twenty-first international conference on Machine learning, p. 74, July 2004.
[5] S. Sinha, S. Ebrahimi, and T. Darrell, “Variational adversarial active learning,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 5972–5981, Seoul, Korea, November 2019.
[6] M. Gorriz, A. Carlier, E. Faure, and X. Giro-i Nieto, “Cost-effective active learning for melanoma segmentation,” 2017, https://arxiv.org/abs/1711.09168v2.
[7] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, “Cost-effective active learning for deep image classification,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 12, pp. 2591–2600, 2017.
[8] B. Settles, “Active learning,” Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, no. 1, pp. 1–114, 2012.
[9] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 3–12, Springer-Verlag New York, Inc., Dublin, Ireland, July 1994.
[10] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” 2017, https://arxiv.org/abs/1703.02910v1.

[11] S. Ebrahimi, M. Elhoseiny, Trevor Darrell, and M. Rohrbach, “Uncertainty-guided continual learning with bayesian neural networks,” 2019, https://arxiv.org/abs/1906.02425v2.

[12] D. Gissin and S. Shalev-Shwartz, “Discriminative active learning,” 2018, https://arxiv.org/abs/1907.06347v1.

[13] O. Sener and S. Savarese, “Active learning for convolutional neural networks: a core-set approach,” in International Conference on Learning Representations, Vancouver, BC, Canada, April 2018.

[14] S. D. Jain and K. Grauman, “Active image segmentation propagation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2864–2873, San Juan, PR, USA, June 2016.

[15] Damien François, “High-dimensional data analysis,” in From Optimal Metric to Feature Selection, pp. 54-55, VDM Verlag Saarbrucken, Saarbrucken, Germany, 2008.

[16] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” https://arxiv.org/abs/1312.6114v10.

[17] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: a review and new perspectives,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[18] I. Kossyk and Z.-C. Márton, “Discriminative regularization of the latent manifold of variational auto-encoders,” Journal of Visual Communication and Image Representation, vol. 61, pp. 121–129, 2019.

[19] H. Du, S. Zhao, D. Zhang et al., “Novel clustering-based approach for local outlier detection,” in Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 802–811, IEEE, San Francisco, CA, USA, April 2016.

[20] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, “Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2389–2406, 2018.

[21] J. Wu, S. Guo, J. Li, and D. Zeng, “Big data meet green challenges: greening big data,” IEEE Systems Journal, vol. 10, no. 3, pp. 873–887, 2016.

[22] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, “Big data meet cyber-physical systems: a panoramic survey,” IEEE Access, vol. 6, pp. 73603–73636, 2018.

[23] F. Severino and Galán, ”Comparative evaluation of region query strategies for DBSCAN clustering.” Information Sciences, vol. 502, pp. 76–90, 2019.

[24] L. Li Deng, “The MNIST database of handwritten digit images for machine learning research [best of the web],” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, 2012.

[25] LeCun, Y. The MniSt Database of Handwritten Digits. 1998.

[26] H. Xi, H. Liu, and Yu Zhang, “Recognition and optimization algorithm of MNIST dataset based on LeNet5 network structure,” in Proceedings of the 2018 International Conference on Transportation & Logistics,Information & Communication,Smart City(TLICSC 2018), pp. 333–339, Atlantis Press, Chengdu, China, October 2018.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999.

[28] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, “Constrained k-means clustering with background knowledge,” in Proceedings of the International Conference on Machine Learning, pp. 577–584, San Francisco, CA, US, June 2001.

[29] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603–619, 2002.