Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.034543
Article

Tech Science Press

Contrastive Clustering for Unsupervised Recognition of Interference Signals

Xiangwei Chen[1], Zhijin Zhao[1,2,*], Xueyi Ye[1], Shilian Zheng[2], Caiyi Lou[2] and Xiaoniu Yang[2]

[1]The School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, 310018, China
[2]The Science and Technology on Communication Information Security Control Laboratory, Jiaxing, 314001, China
*Corresponding Author: Zhijin Zhao. Email: zhaozj03@hdu.edu.cn

Received: 20 July 2022; Accepted: 21 October 2022

Abstract: Interference signals recognition plays an important role in anti-jamming communication. With the development of deep learning, many supervised interference signals recognition algorithms based on deep learning have emerged recently and show better performance than traditional recognition algorithms. However, there is no unsupervised interference signals recognition algorithm at present. In this paper, an unsupervised interference signals recognition method called double phases and double dimensions contrastive clustering (DDCC) is proposed. Specifically, in the first phase, four data augmentation strategies for interference signals are used in data-augmentation-based (DA-based) contrastive learning. In the second phase, the original dataset's k-nearest neighbor set (KNNset) is designed in double dimensions contrastive learning. In addition, a dynamic entropy parameter strategy is proposed. The simulation experiments of 9 types of interference signals show that random cropping is the best one of the four data augmentation strategies; the feature dimensional contrastive learning in the second phase can improve the clustering purity; the dynamic entropy parameter strategy can improve the stability of DDCC effectively. The unsupervised interference signals recognition results of DDCC and five other deep clustering algorithms show that the clustering performance of DDCC is superior to other algorithms. In particular, the clustering purity of our method is above 92%, SCAN's is 81%, and the other three methods' are below 71% when jamming-noise-ratio (JNR) is −5 dB. In addition, our method is close to the supervised learning algorithm.

Keywords: Interference signals recognition; unsupervised clustering; contrastive learning; deep learning; k-nearest neighbor

1 Introduction

With the rapid development of wireless communication technology and increased electronic equipment, the future communication system will face more kinds of interference. Adopting corresponding anti-interference methods according to the identified interference signals can improve communication quality. Therefore, interference signals recognition will be one of the critical technologies of intelligent anti-interference communication in the future.

Deep learning has achieved excellent performance in natural language processing [1,2] and computer vision [3–5]. Therefore, many recognition algorithms based on deep learning for interference signals were proposed to solve the problems of traditional interference signal recognition algorithms whose accuracy is low and significantly affected by artificial feature selection [6–12]. However, most interference signals recognition algorithms belong to supervised learning that requires labeled signals samples. Labeling each signal sample requires a lot of labor costs, which limits the applications of these algorithms. Therefore, unsupervised interference signals recognition should study urgently.

Clustering is one of the unsupervised learning methods which can classify data into different clusters without any labeled samples. Traditional clustering methods include partition-based methods, density-based methods, and hierarchy-based methods. In recent years, many unsupervised clustering methods of radio frequency signals have been proposed [13–19]. However, these methods used traditional clustering algorithms and manually selected features. Whether the feature selection is appropriate or not will directly affect its performance. Therefore, without prior knowledge of an interference signal, it is impossible to design artificial features in advance. In addition, the performance of traditional clustering algorithms is poor.

Recently, many deep clustering methods which solve the problems of traditional clustering algorithms have been proposed [20–25], such as semantic clustering by adopting nearest neighbors (SCAN) [24] and contrastive clustering (CC) [25]. Still, these algorithms were used to cluster images, and there is no research on deep clustering for interference signals. On the one hand, the performance of the deep clustering methods with contrastive learning is better than that without it in image clustering. On the other hand, since the interference signal contains noise, the algorithms which used contrastive learning in only one phase will lead to degradation of the noise-immunity caused by data augmentation. Therefore, a method, DDCC, is proposed in this paper. The main contributions can be summarized as follows.

- To achieve unsupervised interference signals recognition, we propose a method named double phases and double dimensions contrastive clustering. Unlike traditional clustering algorithms, DDCC uses a deep neural network to extract the feature of interference signals automatically.
- We design four data augmentation strategies for interference signals to carry out DA-based contrastive learning, which is used to pre-train the deep neural network in the first phase of DDCC. In the second phase of DDCC, the KNNset of original data through the pre-trained network is obtained, and the KNNset-based double dimensions contrastive learning is designed to cluster the samples according to their category features. In addition, the stability of DDCC is improved by using a dynamic entropy parameter strategy for regularization.
- We analyze the performance of the different parts of DDCC through extensive simulation experiments. DDCC, deep embedded clustering (DEC), DeepCluster, deep adaptive clustering (DAC), SCAN, and CC are applied to unsupervised interference signals recognition and compared in experiments. The results show that DDCC outperforms others on unsupervised interference signals recognition under various JNR.

The rest of this paper is organized as follows. Section 2 describes recent work in interference signal recognition based on deep learning and deep clustering. Section 3 discusses the proposed method in detail. Section 4 gives the simulation results and analyzes the performance of different methods in detail. Lastly, Section 5 concludes this paper.

2 Related Work

2.1 Interference Signal Recognition Based on Deep Learning

With the development of deep learning technology, there have been recognition algorithms based on deep learning. The singular value of the signal matrix was used as the input of the multi-layer perceptron

(MLP) in [6], and short-time Fourier transform data of jamming signals was used as the input of the convolutional neural network (CNN) in [7,8]. Jamming recognition network [9] based on power spectrum features was proposed to recognize ten kinds of suppression jamming signals. Siamese-CNN [10] and auxiliary classifier variational auto-encoding generative adversarial network [11] were proposed to solve the performance deterioration of the interference signals recognition method in the case of a small sample set. The Auto-Encoder network, which was built by stacking long short-term memory, separated the interference signal from the transmitted signal, and another recurrent neural network realized interference signals recognition [12]. But the above algorithms are supervised, which requires a lot of the labeled signals samples. Deep clustering is one of the unsupervised learning methods.

2.2 Deep Clustering

Deep clustering methods can be divided into two groups. The first group of methods combined traditional clustering algorithms with deep neural networks, such as DEC [20], DeepCluster [21], etc. DEC pre-trained a stacking auto-encoder to obtain the features and used the Kullback-Leibler divergence to jointly train the network and the clustering center obtained by k-means. DeepCluster carried out network training process and k-means clustering alternately. However, their performance can be compromised by the errors accumulated during the traditional clustering. The second group of methods used representation learning and the idea of “label as representation” to solve the problem of the first group of methods, such as DAC [22], invariant information clustering (IIC) [23], SCAN [24], CC [25], etc. DAC used the binary classification with the pseudo label as the pretext task, while IIC used mutual information maximization to realize representation learning. But sometimes, their category features are insufficient to reflect the category relationship between samples. Recently, unsupervised contrastive learning (CL) has achieved state-of-the-art performance in representation learning [26–29]. SCAN and CC that applied contrastive learning to clustering have better clustering performance than DAC and IIC. Because of the high intra-class compactness and inter-class separability of sample features obtained by contrastive learning methods such as simple framework for contrastive learning of visual representations (SIMCLR) [29], SCAN obtained embedding space by using SIMCLR and used the SCAN-loss [24] to make adjacent samples have similar category features. CC performed the instance-level and clustering-level contrastive learning to improve the feature extraction ability of the network and to cluster following the idea of “label as representation”. But SCAN and CC algorithms used contrastive learning only in one phase so that the features extraction network cannot finetune. Moreover, SCAN ignored the inter-class separability of category features, and CC did not utilize the feature distribution explicitly. Therefore the performance of deep clustering for interference signals with noise will deteriorate. Therefore, double phases and double dimensions contrastive clustering for interference signals recognition is proposed to cope with these problems.

3 Double Phases and Double Dimensions Contrastive Clustering

3.1 Data-Augmentation-Based Contrastive Learning

In supervised learning, the feature extraction network training can be guided by sample labels. In contrast, unsupervised learning requires designing pretext-task for network training to enable the network to obtain feature extraction ability, such as rotation prediction [30] and puzzles [31] in image classification. As a promising paradigm of representation learning, contrastive learning is to make the similarity between positive pairs as high as possible and the similarity between negative pairs as low as possible. However, unsupervised learning has no available label to obtain information on positive and negative pairs. Although positive pairs can be obtained by data augmentation that IIC did, the lack of information on negative pairs makes it difficult for features to have inter-class separability.

It is a solution that data augmentation samples from the same sample were considered positive pairs, and that from the different samples were considered negative pairs. Then, the network parameters were updated using the respective contrastive loss functions in [26,27]. Momentum contrast [28] proposed a momentum training approach based on [26], which improves performance in the case of insufficient resources but requires a massive dataset for network training. SIMCLR [29] used a larger batch size, more complex data augmentation strategies, and projection head based on [27], improving the performance without restricting resources. Following SIMCLR, the DA-based contrastive learning is designed in the first phase of DDCC, and the process is shown in Fig. 1.

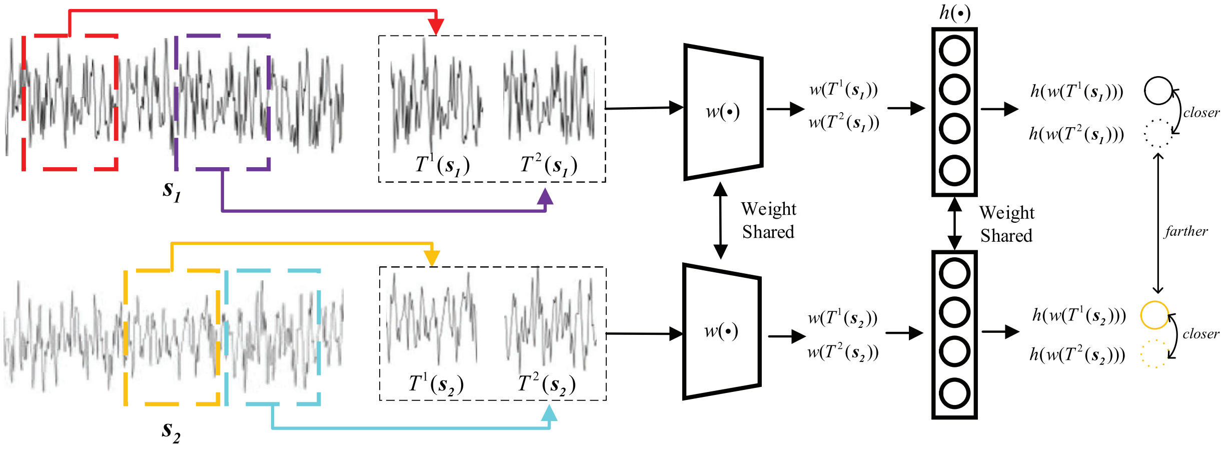


Figure 1: The process of DA-based contrastive learning

Given the unlabeled interference signals dataset $D = \{\boldsymbol{s}_i\}_{i=1}^{N}$, where N is the number of samples, $\boldsymbol{s}_i = (s_i[0], s_i[1], s_i[2], \ldots)$ is the sequence of an interference signal. The data augmentation dataset $D^{new} = \{T^1(\boldsymbol{s}_i), T^2(\boldsymbol{s}_i)\}_{i=1}^{N}$ is obtained by applying two data augmentation T^1 and T^2 to all the samples from D. One shared feature extraction network $w(\cdot)$ is used to extract the features of the sample from D^{new}, and a projection head $h(\cdot)$ embeds the features into the K_f dimensional embedding space to get $\boldsymbol{z}_n^m = h(w(T^m(\boldsymbol{s}_n)))$.

The cosine distance between sample features is used to measure the similarity between samples, as shown in Eq. (1),

$$sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} \tag{1}$$

The normalized temperature-scaled cross entropy (NT-Xent) [32] is used as the loss function of DA-based contrastive learning to maximize the similarity of the positive pair while minimizing the similarity of the negative pair, as shown in Eq. (2),

$$L_1 = \frac{1}{2N}\sum_{n=1}^{N}\left[l(\boldsymbol{z}_n^1, \boldsymbol{z}_n^2) + \boldsymbol{l}(\boldsymbol{z}_n^2, \boldsymbol{z}_n^1)\right] \tag{2}$$

where $l(\boldsymbol{x}, \boldsymbol{y}) = -\log \dfrac{e^{\tau sim(x,y)}}{\sum_{m=1}^{2}\sum_{k=1}^{N} e^{\tau sim(x,z_k^m)} - 1}$, τ is the temperature parameter and we let $\tau = 0.5$.

3.2 *KNNset-Based Contrastive Learning*

In most cases, the nearest neighbors in the embedding space obtained by SIMCLR belong to the same class [24]. Therefore, to enable the network to mine more meaningful semantic information and get better clustering performance, the second phase of DDCC constructs positive and negative pairs according to the KNNset, and performs double dimensions contrastive learning in the feature and the clustering dimension.

3.2.1 *Construction of Positive and Negative Pairs*

Original samples can provide more meaningful semantic information, so the second phase of DDCC uses KNNset to construct positive and negative pairs while not using data augmentation.

According to the embedding space obtained from contrastive learning in the first phase and cosine distance between samples, KNNset $\Delta^k = \{\eta_k(\boldsymbol{s}_i)\}_{i=1}^{N}$ is obtained from the dataset D, where $\eta_k(\boldsymbol{s}_i)$ denotes the set of k-nearest neighbors of sample $\boldsymbol{s}_i$.

In the training phase, given a batch of samples $\{\boldsymbol{s}_i\}_{i=1}^{M}$, two samples are randomly selected from $\eta_k(\boldsymbol{s})$ and two new batches $\{\boldsymbol{s}_i^a\}_{i=1}^{M}$ and $\{\boldsymbol{s}_i^b\}_{i=1}^{M}$ are obtained, where $\boldsymbol{s}_i^a$ and $\boldsymbol{s}_i^b$ are selected from $\eta_k(\boldsymbol{s}_i)$, and M is the size of a batch. $\boldsymbol{s}_i^a$ and $\boldsymbol{s}_i^b$ are considered as positive pairs, $\boldsymbol{s}_i^a$ and $\boldsymbol{s}_j^b$ are considered negative pair, where $i \neq j$.

3.2.2 *Double Dimensions Contrastive Learning*

As shown in Fig. 2, DDCC performs contrastive learning in the second phase at the feature and clustering dimensions, respectively. The feature dimensional contrastive learning enables the network to extract more meaningful semantic features. In contrast, the clustering dimensional contrastive learning clusters signals through considering category features as the soft assignment probabilities of clusters. Note that, the parameters of $w(\cdot)$ and $h_F(\cdot)$ in the KNNset-based contrastive learning are initialized by the parameters of $w(\cdot)$ and $h(\cdot)$ in the DA-based contrastive learning, and the parameters of $h_C(\cdot)$ are initialized randomly.

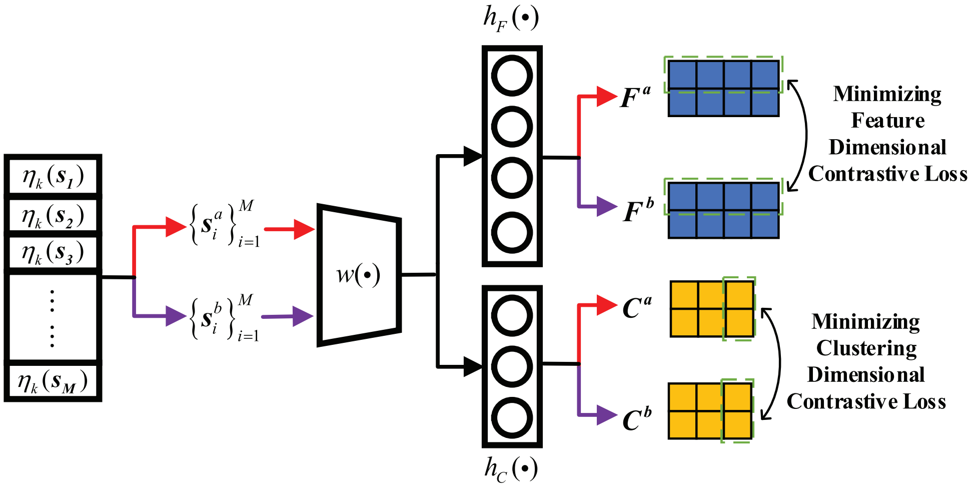


Figure 2: The process of KNNset-based contrastive learning

In the feature dimension, let the feature dimensional outputs of signals $\boldsymbol{s}_n^a$ and $\boldsymbol{s}_n^b$ be $\boldsymbol{f}_n^a = h_F(w(\boldsymbol{s}_n^a))$ and $\boldsymbol{f}_n^b = h_F(w(\boldsymbol{s}_n^b))$, respectively, and define the output matrices as $\boldsymbol{F}^a = [\boldsymbol{f}_1^a, \boldsymbol{f}_2^a, \cdots, \boldsymbol{f}_M^a]^{\mathrm{T}} \in \mathbb{R}^{M \times K_f}$ and $\boldsymbol{F}^b = [\boldsymbol{f}_1^b, \boldsymbol{f}_2^b, \cdots, \boldsymbol{f}_M^b]^{\mathrm{T}} \in \mathbb{R}^{M \times K_f}$, where K_f is feature dimension. And we also utilize the NT-Xent as the loss function of feature dimensional contrastive learning, as shown in Eq. (3),

$$L_F = \frac{1}{2M}\sum_{n=1}^{M}\left[l_F(\boldsymbol{f}_n^a, \boldsymbol{f}_n^b) + l_F(\boldsymbol{f}_n^b, \boldsymbol{f}_n^a)\right] \tag{3}$$

where $l_F(\boldsymbol{x}, \boldsymbol{y}) = -\log\frac{e^{\tau_F sim(x,y)}}{\sum_{n=1}^{M} e^{\tau_F sim(x,f_n^a)} + \sum_{n=1}^{M} e^{\tau_F sim(x,f_n^b)} - 1}$, τ_F is the temperature parameter and we let $\tau_F = 0.5$.

In the clustering dimension, a SoftMax layer is added to the end of $h_C(\cdot)$ to satisfy the probability's non-negative and sum-to-one properties. Therefore, $\boldsymbol{c}_n^a = h_C(w(\boldsymbol{s}_n^a))$ can be defined the soft assignment probabilities of each cluster of sample s_n^a. The inference process is formulated as:

$$s \in class\ \arg \max_{i=1}^{K_c} h_C(w(s)) \tag{4}$$

where K_c is the output dimension of the clustering dimensional projection head, i.e., the number of clusters.

$\boldsymbol{C}^a = [\boldsymbol{c}_1^a, \boldsymbol{c}_2^a, \cdots, \boldsymbol{c}_M^a]^{\mathrm{T}} \in \mathbb{R}^{M\times K_c}$ and $\boldsymbol{C}^b = [\boldsymbol{c}_1^b, \boldsymbol{c}_2^b, \cdots, \boldsymbol{c}_M^b]^{\mathrm{T}} \in \mathbb{R}^{M\times K_c}$ are the clustering dimensional output matrices. $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_i^b$ are the i^{th} column of $\boldsymbol{C}^a$ and the i^{th} column of $\boldsymbol{C}^b$, respectively. So $\boldsymbol{y}_i^a$ and $\boldsymbol{y}_i^b$ can be considered as two representations of the i^{th} cluster. Ideally, $\boldsymbol{y}_i^a$ is equal to $\boldsymbol{y}_i^b$, and $\boldsymbol{y}_i^a$ is orthogonal to $\boldsymbol{y}_j^b$ when $i \neq j$. Similar to the feature dimensional contrastive learning, the representations of the same cluster should be similar enough, while the representations of different clusters should be different as much as possible, so the clustering dimensional contrastive learning still uses NT-Xent as loss function:

$$L_C = \frac{1}{2K_c}\sum_{k=1}^{K_c}\left[l_C(\boldsymbol{y}_k^a, \boldsymbol{y}_k^b) + l_C(\boldsymbol{y}_k^b, \boldsymbol{y}_k^a)\right] \tag{5}$$

where $l_C(\boldsymbol{x}, \boldsymbol{y}) = -\log\frac{e^{\tau_C sim(\boldsymbol{x},\boldsymbol{y})}}{\sum_{k=1}^{K_c} e^{\tau_C sim(x,y_k^a)} + \sum_{k=1}^{K_c} e^{\tau_C sim(x,y_k^b)} - 1}$, τ_C is the temperature parameter and we let $\tau_C = 1$.

To avoid the trivial solution caused by assigning nearly all samples to one cluster, we add the entropy term as shown in Eq. (6) for regularization:

$$L_H = -\sum_{n=1}^{K_c}\left[P(\boldsymbol{y}_n^a)\log P(\boldsymbol{y}_n^a) + P(\boldsymbol{y}_n^b)\log P(\boldsymbol{y}_n^b)\right] \tag{6}$$

where $P(\boldsymbol{y}_n^a) = \left\|\boldsymbol{y}_n^a\right\|_1 / \|\boldsymbol{C}^a\|_1$, $P(\boldsymbol{y}_n^b) = \left\|\boldsymbol{y}_n^b\right\|_1 / \left\|\boldsymbol{C}^b\right\|_1$.

Finally, to realize the end-to-end training, that is, to update the parameters of the feature extraction network $w(\cdot)$, those of the feature dimensional projection head $h_F(\cdot)$ and those of the clustering dimensional projection head $h_C(\cdot)$ at the same time, the hybrid loss function of KNNset-based contrastive learning is defined as:

$$L_2 = L_F + L_C + \lambda L_H \tag{7}$$

where λ is the entropy parameter. A small λ leads to a trivial solution; a large λ can avoid trivial solution but ignores the clustering task of the clustering dimension. Since different datasets may require different λ, we propose a dynamic entropy parameter strategy as shown in Eq. (8):

$$\lambda = \lambda_{\max}\left(0.1 \times \left\lceil \frac{10 \times e}{e_{\max}} \right\rceil\right) \tag{8}$$

where $\lambda_{\max}$ and $e_{\max}$ are the maximum entropy parameter and the maximum number of iterations, respectively. e is the current epoch, $\lceil\cdot\rceil$ denote ceiling operation.

In summary, the entire process of DDCC is shown in Algorithm 1.

Algorithm 1: DDCC

```
Input: Dataset $D$, feature extraction network $w(\cdot)$, projection head $h(\cdot)$, feature dimensional projection head $h_F(\cdot)$, cluster dimensional projection head $h_C(\cdot)$, update strategy of entropy term parameter, batch size $M$, maximum iterations $e_{\max}$ and other training hyperparameters.
First Contrastive Learning:
For $e = 1, 2, \ldots, e_{\max}$ do
    For $t = 1, 2, \ldots, |D|/M$ do
        Sample mini-batch $\{s_k\}_{k=1}^{M}$ from $D$;
        Perform two augmentations to $\{s_k\}_{k=1}^{M}$ to get $\{T^1(s_k), T^2(s_k)\}_{k=1}^{M}$;
        Compute NT-Xent loss $L_1$ through Eq. (2) and update $w(\cdot)$ and $h(\cdot)$ to minimize $L_1$;
    End for
End for
Second Contrastive Learning:
Get KNNset $\Delta^k = \{\eta_k(s_i)\}_{i=1}^{N}$ using $w(\cdot)$ and $h(\cdot)$;
For $e = 1, 2, \ldots, e_{\max}$ do
    For $t = 1, 2, \ldots, |D|/M$ do
        Sample mini-batch $\{s_k\}_{k=1}^{M}$ from $D$;
        Get $\{s_k^a\}_{k=1}^{M}$ and $\{s_k^b\}_{k=1}^{M}$ according to $\Delta^k$ and $\{s_k\}_{k=1}^{M}$;
        Compute feature contrastive loss $L_F$, cluster contrastive loss $L_C$ and the entropy term $L_H$ through Eqs. (3), (5) and (6);
        Compute hybrid loss $L_2$ through Eq. (7) and update $w(\cdot)$, $h_F(\cdot)$ and $h_C(\cdot)$ to minimize $L_2$;
        Update $\lambda$ according to Eq. (8);
    End for
End for
Output: $K_c$ clusters of samples from $D$.
```

4 Performance Analysis

4.1 The Setting of the Experiment

4.1.1 Datasets

Nine kinds of interference signals, which include single-tone jamming, multi-tone jamming, periodic Gaussian pulse jamming, frequency hopping jamming, linear sweeping frequency jamming, sinusoidal sweeping frequency jamming, binary phase shift keying (BPSK) modulation jamming, noise frequency modulation jamming and quadrature phase shift keying (QPSK) modulation jamming, are generated by matlabR2020a. The details of the parameters of each interference signal are shown in Table 1. The sampling frequency is 10 MHz, and the number of sampling points is 1024. The added noise is additive white Gaussian noise, JNR is −5, 0, 5 and 10 dB, respectively. The number of samples for each category under a JNR is 1000, so there are four datasets with different JNR, and each dataset contains 9000 samples of interference signals.

Table 1: Interference signal and their parameters setting

Interference types	Parameter setting
Single-tone	The center frequency f_c is between [100, 500] kHz, and the phase φ is between [0, 2π].
Multi-tone	The number N of audio is [2,10], and f_c, φ are the same as single-tone jamming.
Periodic Gaussian pulse	The pulse period T is 2.5∼10 μs, and the duty cycle is 1/8∼1/2.
Frequency hopping	N = 20, $\{f_c\}$ is between [100, 500] kHz, the frequency hopping period T_h is between 3.2∼6.4 μs, and the phase φ is between [0, 2π].
Linear sweeping frequency	The starting frequency f_{c1} is [50, 100] kHz, and the ending frequency f_{c2} is [300, 1000] kHz.
Sinusoidal sweeping frequency	The frequency varies sinusoidally, and other parameters are the same as linear sweeping frequency jamming.
BPSK modulation	The information symbol is a 32-bits 0, 1 random sequence, the symbol period is 3.2 μs, and the modulation signal is sinusoidal.
Noise frequency	The frequency modulation coefficient is between [0.125, 0.933], and the carrier signal parameters are the same as single-tone jamming.
QPSK modulation	The information symbol is a 32-bit 0, 1 random sequence, and the symbol period is 3.2 μs, the I-channel modulation signal is sinusoidal, Q-channel modulation signal is cosine.

4.1.2 Network Architecture

We use 1-dimensional densely connected convolutional network [33] as a feature extraction network $w(\cdot)$, and the network used in the first and second phase of DDCC is shown in Fig. 3a. The convolution layer parameter meanings are the convolution kernel's size, type, number, and stride. For example, {7 conv1d, 32, /2} means the convolution layer uses one-dimensional convolution with 32 convolution kernels whose size is 7×1 and stride is 2. For simplicity, we do not draw the batch-normalization layer and rectified linear unit layer before each convolution layer in Fig. 3a. The pooling layer parameter meanings are the pooling kernel's type, size, and stride. For example, {1dAvgPool, 4, /2} means the pooling layer uses one-dimensional average pooling with a pooling kernel whose size is 4×1, and the stride is 2. The fully-connected (FC) layer parameter is the number of the output dimension. The DenseBlock module is shown in Fig. 3b, bn and gr are its parameters of the bottleneck structure and the growth rate, respectively. We set $bn = 4$, $gr = 48$. The structure of the transition layer is shown in Fig. 3c, Dim_{in} is the input dimension of the transition layer, and cr is the compression rate of the transition layer which is set to 0.5.

4.1.3 Training Parameter Setting

During training, the whole network is trained for 120 epochs using the adaptive moment estimate algorithm with 0.0003 learning rate as the optimizer and batch gradient descent with a batch size of 512 as the optimization algorithm. The $\lambda_{\max}$ of dynamic entropy parameter strategy is 0.5.

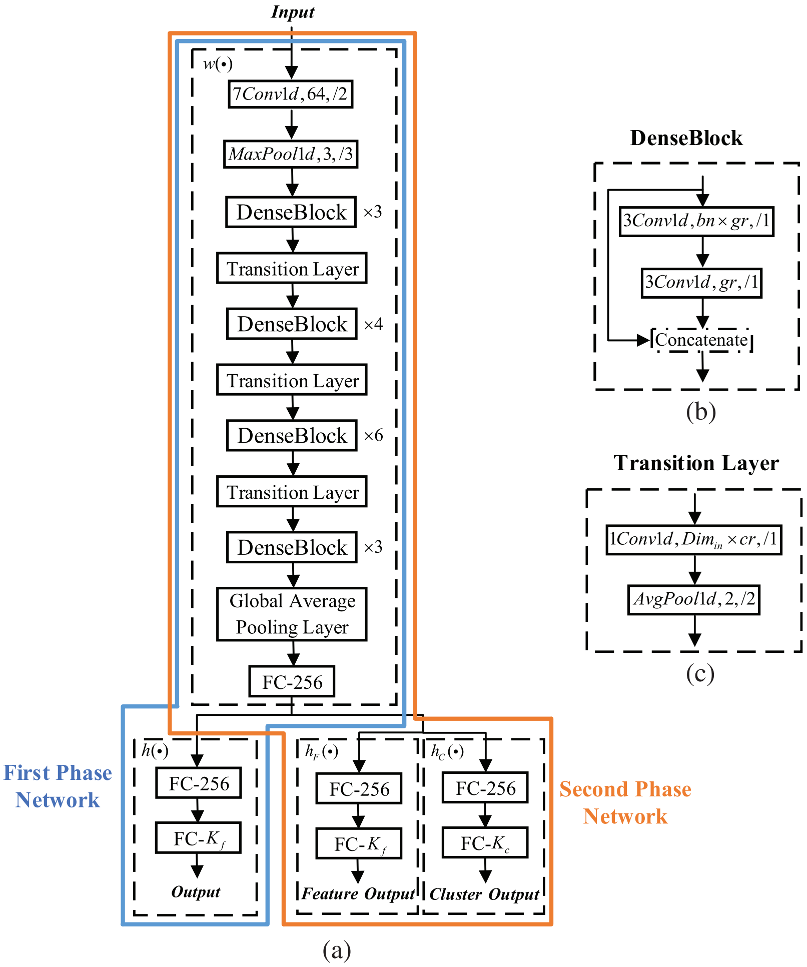


Figure 3: The architectures (a) The network of DDCC; (b) The DenseBlock; (c) The Transition Layer

4.2 Performance Analysis of Data Augmentation Strategies

Four data augmentation strategies, including random cropping, multipath synthesis, down-sampling, and noise injection, to implement DA-based contrastive learning of interference signals, are used and analyzed in this section.

The performance of the data augmentation strategy of the interference signal is measured by the label matching degree (LMD), as shown in Eq. (9):

$$LMD = \frac{1}{N}\sum_{n=1}^{N}\frac{\Lambda(\eta_k(e_n);\ e_n)}{k} \tag{9}$$

where $\Lambda(\eta_k(e_n);\ e_n)$ denotes the number of samples in $\eta_k(e_n)$ with labels consistent with e_n.

The length of random cropping is 256, the number of paths of multipath synthesis is 3, the down-sampling rate is 4, and additive white Gaussian noise with half the power of the original signal is used in noise injection. When k = 20, $K_f = 8$ and JNR = −5, 0, 5, and 10 dB, the LMD of different data augmentation strategies are shown in Fig. 4. It can be seen that the LMD of all four strategies increase with the increase of JNR. The LMD of the random cropping strategy is optimal at different JNR. Therefore, random cropping is used as the data augmentation in the following experiments.

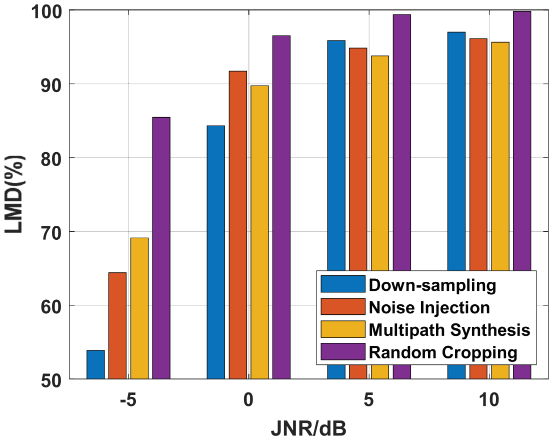


Figure 4: The LMD comparison of contrastive learning using different data augmentation strategies

When JNR is 10 dB, and the random cropping strategy is used with $K_f = 3$, the 3D scatter plot of the features of the interference signals obtained from the first phase of DDCC is shown in Fig. 5. Note that, because of the normalization, all the samples are projected onto the unit circle. It can be seen that the different types of interference signals have different intra-class compactness and high inter-class separation. So the pre-training method, SIMCLR, can also be applied to the interference signal.

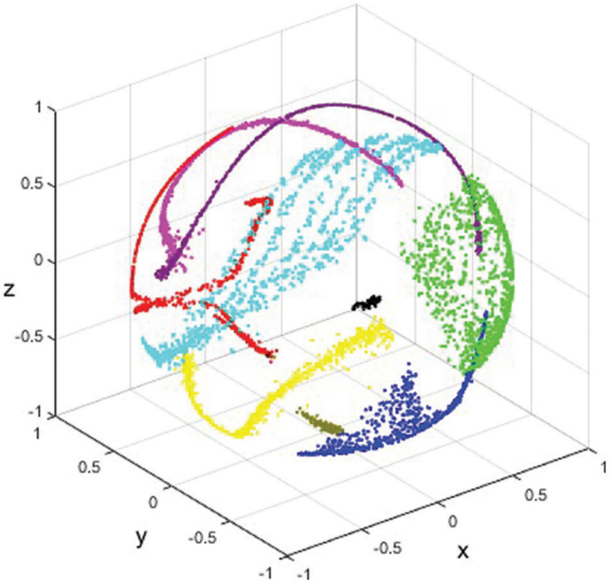


Figure 5: The 3D scatter of signal features in the embedding space

4.3 Effect of the KNNset Selection and Feature Dimensional Contrastive Learning

The data augmented and original samples are fed into the pre-training network to obtain the corresponding KNNset. When k = 20, the LMD of two KNNsets with different JNR is shown in Table 2. It can be seen that the LMD of the KNNset obtained according to data augmentation samples decreases significantly when the JNR decreases. In contrast, the LMD of the KNNset obtained according to original samples remains above 95% when JNR is −5 dB. It indicates that although random cropping can obtain excellent LMD of KNNset, it may also lead to losing important information. Since the interference signal contains noise, the data augmentation also affects the noise immunity performance of the algorithm. Therefore, the KNNset is constructed according to original samples and data augmentation is not performed on the KNNset in the second phase.

Table 2: The LMD of different KNNsets under various JNR

KNNset\JNR	−5 dB	0 dB	5 dB	10 dB
Augment data	85.45	96.52	99.35	99.81
Original data	**95.58**	**98.97**	**99.94**	**99.98**

DDCC without using feature dimensional contrastive learning is abbreviated as DCC, and DDCC using data-augmented KNNset is abbreviated as DDCC(DA). The clustering performance of these algorithms is measured using purity, which is defined as Eq. (10):

$$Purity = \frac{1}{N}\sum_{i=1}^{K_c} \max_j \left| p_i \cap q_j \right| \tag{10}$$

where N is the number of samples, p_i is the set of samples predicted to belong to the i^{th} cluster, q_j is the set of samples that belong to the j^{th} class, and $\left| p_i \cap q_j \right|$ denotes the total number of samples in the intersection of p_i and q_j.

The clustering purity of DDCC, DCC, and DDCC(DA) at different JNR is shown in Table 3. We can see that: (1) the purity of DDCC is the highest; the purity of DDCC is higher than that of DCC, indicating that feature dimensional contrastive learning can improve the clustering performance; the purity of DDCC is higher than that of DDCC(DA), indicating that data augmentation to KNNset impairs the noise immunity of DDCC. (2) The purity of DCC is lower than that of DDCC(DA) under 5 dB and 10 dB JNR, and the purity of DCC is better than that of DDCC(DA) under −5 dB and 0 dB JNR, indicating that larger noise power is, more severe degradation of noise immunity performance caused by data augmentation is.

Table 3: The purity of DDCC, DCC and DDCC(DA) under different JNR

Method\JNR	−5 dB	0 dB	5 dB	10 dB
DDCC	**93.06**	**96.07**	**98.78**	**99.98**
DDCC(DA)	84.76	90.72	95.16	99.78
DCC	87.29	93.56	94.16	97.94

4.4 Effect of Dynamic Entropy Parameter Strategy

Since datasets under various JNR are different and different pre-training networks obtained from the first phase of DDCC lead to different KNNsets, this section analyzes the effect of the dynamic entropy parameter strategy of DDCC.

When the entropy parameter is fixed, the purity curves of DDCC with different pre-training networks under different entropy parameters and JNR are shown in Fig. 6. It can be seen that the optimal parameters are related to the JNR and the pretrained network. When JNR = −5 dB, the optimal entropy parameters of different pretrained networks are 0.15, 0.35 and 0.45, respectively. When JNR = 0 dB, the optimal entropy parameters are 0.1, 0.2 and 0.5, respectively. When JNR = 5 dB, the optimal entropy parameters are 0.15, 0.2 and 0.25, respectively. When JNR = 10 dB, the optimal entropy parameters is 0.05 and 0.15, respectively.

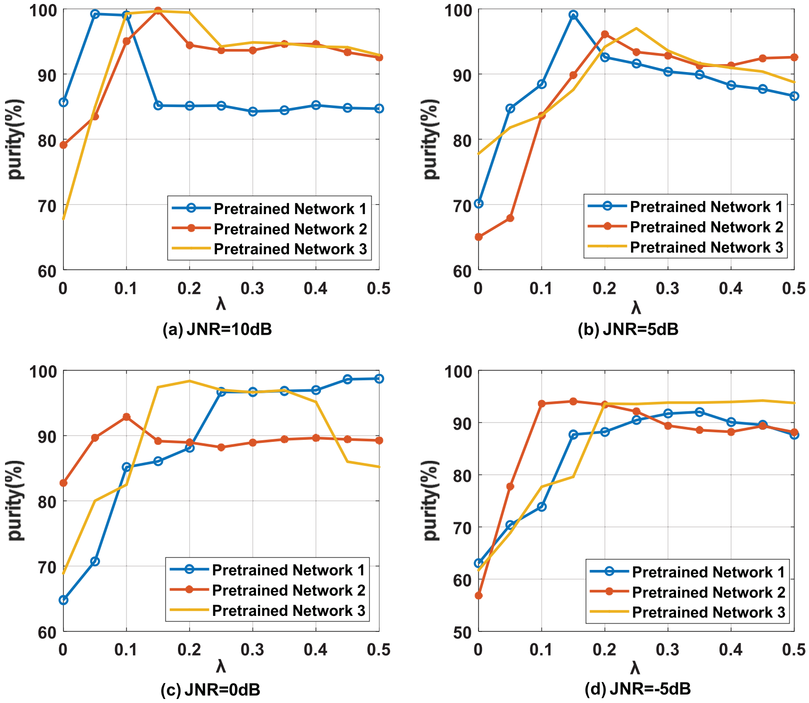
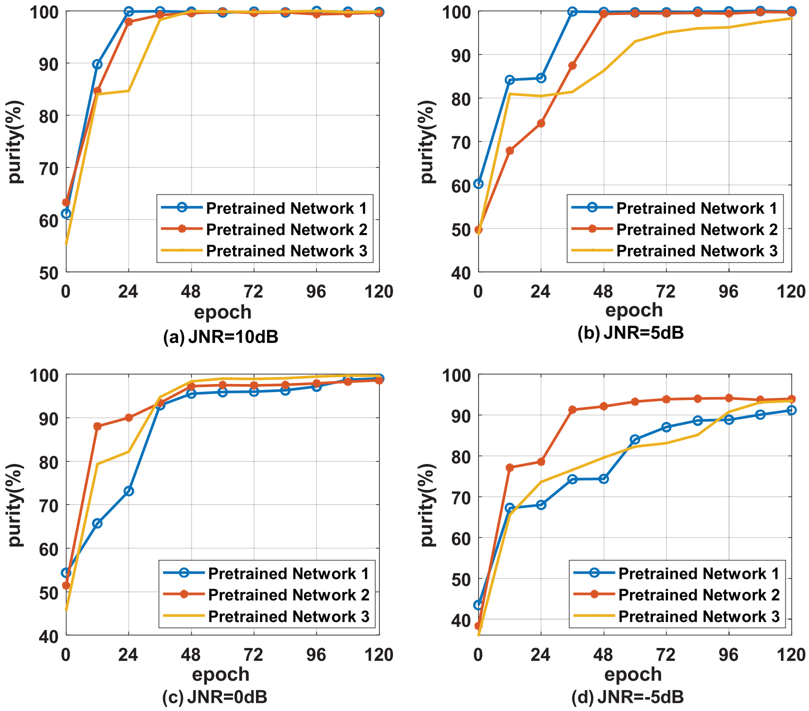


Figure 6: The purity of different pre-trained networks under different JNR using fixed parameter

When using the dynamic entropy parameter strategy, the purity curves of DDCC with different pre-trained networks under different training epoch and JNR are shown in Fig. 7. It can be seen that with the training, the increase of entropy parameter does not cause a significant decrease of purity, and the optimal purity of DDCC with dynamic entropy parameter is higher than that with fixed entropy parameter. In summary, the dynamic entropy parameter strategy proposed in this paper can effectively improve the stability of DDCC.

4.5 Clustering Performance of Different Methods for Interference Signals

This section applies DDCC, DEC, DeepCluster, DAC, SCAN, and CC to interference signals unsupervised clustering under various JNR. The encoder of DEC, the feature extraction network of DeepCluster, and DAC are the same as the feature extraction network of DDCC. The decoder of DEC reconstructs the original signal through a two-layer FC layer. The network structure of SCAN is the same as the pre-training network of DDCC. The clustering heads of DeepCluster, DAC and SCAN have the same structure as the clustering dimensional projection head of DDCC. The network structure of CC is identical to that of the second phase of DDCC. SCAN, CC, and DDCC all use random cropping as data augmentation. In addition, the performance of the supervised learning method using the cross-entropy loss function is also given. The training set of the unsupervised methods is the test set, and the supervised learning method needs to use an additional test set to evaluate the performance. The purity curves of the seven algorithms under different JNR are shown in Fig. 8. The purity of all methods is the average of the results of 10 random experiments.



(a) JNR=10dB

(b) JNR=5dB

(c) JNR=0dB

(d) JNR=-5dB

Figure 7: The purity of different pre-trained networks under different JNR using dynamic parameter

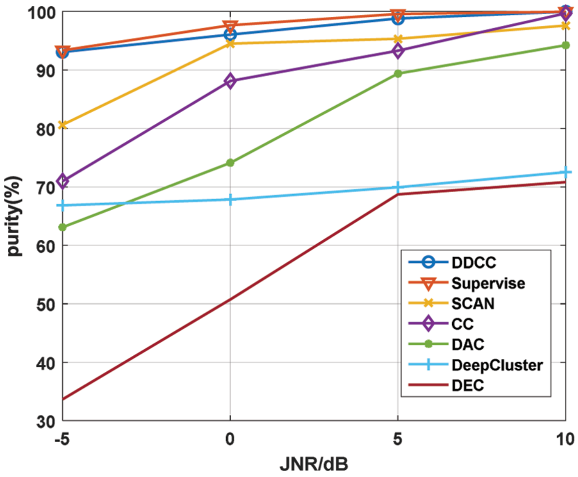


Figure 8: The purity comparison of seven algorithms under different JNR

As can be seen from Fig. 8, the purity of all the algorithms increases with the increase of JNR, and DDCC is the best among the six unsupervised algorithms under different JNR, especially in the low JNR scenario. Under JNR = −5 dB, the purity of DDCC is 12% higher than that of SCAN, which is the suboptimal method. And the purity of DDCC is close to the supervised algorithm when JNR is in the range of −5 to 10 dB. SCAN used SCAN-loss to maintain the intra-class compactness of category features in the second phase, but the loss function ignored their inter-class separability. Unlike SCAN, DDCC uses double dimensional contrastive learning based on KNNset in the second phase, which improves intra-class compactness and inter-class separability of category features and achieves better clustering performance. In addition, the use of the original KNNset makes DDCC have stronger noise immunity. CC used data augmentation samples for double level contrastive learning, which did not explicitly utilize the feature distribution in embedding space obtained by instance-level contrastive learning, and DA-based contrastive learning was not sufficient for obtaining good noise immunity. Unlike CC, DDCC uses KNNset-based contrastive learning, which improves the network feature extraction ability and avoids the problems of information loss and noise immunity degradation caused by data augmentation, as mentioned in Section 4.3. The purity of CC is close to 100% under 10 dB JNR, which is higher than SCAN, but its performance decreases significantly with the decline of JNR.

It also can be seen from Fig. 8 that the purity of the unsupervised algorithms which used contrastive learning is higher than that did not use contrastive learning. The purity of DEC is the worst. DeepCluster performs slightly better than DEC. Still, its clustering purity remains unsatisfactory, while DAC has improved the purity in the JNR range of 0 to 10 dB compared to DeepCluster.

5 Conclusions and Discussion

An unsupervised recognition algorithm of interference signals, double phases and double dimensions contrastive clustering was proposed in this paper. In the first phase, DA-based contrastive learnin was used to pre-train the features extraction network. KNNset-based contrastive learning in the second phase was proposed that used the positive and negative pairs from the KNNset obtained using the distribution of the original signals in the feature space of the pre-trained network. In the second phase, the feature dimensional contrastive learning improved the feature extraction ability of the network by learning the information between the original signals, and the clustering dimensional contrastive learning achieved clustering. Extensive simulation experiments have verified the effectiveness of DDCC. In summary, DDCC has the following characteristics.

DDCC can unsupervised extract more meaningful semantic features and more accurate category features. Simulation results showed the unsupervised recognition performance of DDCC for nine interference signals is better than the other unsupervised clustering algorithms and close to the supervised learning algorithm in the JNR range of −5 to 10 dB.

Random cropping is the best strategy among the four data augmentation strategies in the first phase of DDCC. The double dimensions contrastive learning in the second phase, which uses the KNNset and hybrid loss function, can fine-tune the feature extraction network and cluster more accurately and steadily. Simulation results validated the high clustering purity of DDCC under low JNR and the stability of DDCC. But DDCC needs to know the cluster number in advance like the other deep clustering method.

In future work, open-world recognition of interference signals will be achieved by combining open-set recognition and incremental learning.

Funding Statement: This research was supported by the National Natural Science Foundation of China under Grant No. U19B2016., and Zhejiang Provincial Key Lab of Data Storage and Transmission Technology, Hangzhou Dianzi University.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Anwar Hussen Wadud, M. F. Mridha, J. Shin, K. Nur and A. Kumar Saha, "Deep-bert: Transfer learning for classifying multilingual offensive texts on social media," *Computer Systems Science and Engineering*, vol. 44, no. 2, pp. 1775–1791, 2023.

[2] A. R. W. Sait and M. K. Ishak, "Deep learning with natural language processing enabled sentimental analysis on sarcasm classification," *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2553–2567, 2023.

[3] A. S. Almasoud, "Intelligent deep learning enabled wild forest fire detection system," *Computer Systems Science and Engineering*, vol. 44, no. 2, pp. 1485–1498, 2023.

[4] R. Punithavathi, A. D. C. Rani, K. R. Sughashini, C. Kurangi, M. Nirmala *et al.,* "Computer vision and deep learning-enabled weed detection model for precision agriculture," *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2759–2774, 2023.

[5] K. Bayoudh, R. Knani, F. Hamdaoui and A. Mtibaa, "A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets," *Visual Computer*, vol. 38, no. 8, pp. 2939–2970, 2022.

[6] M. Feng and Z. Wang, "Interference recognition based on singular value decomposition and neural network," *Journal of Electronics & Information Technology*, vol. 42, no. 11, pp. 2573–2578, 2020.

[7] Y. Wang, B. Sun and N. Wang, "Recognition of radar active-jamming through convolutional neural networks," *The Journal of Engineering*, vol. 2019, no. 21, pp. 7695–7697, 2019.

[8] Q. Liu and W. Zhang, "Deep learning and recognition of radar jamming based on CNN," in *Proc. ISCID*, Hangzhou, China, pp. 208–212, 2019.

[9] Q. Qu, S. Wei, S. Liu, J. Liang and J. Shi, "JRNet: Jamming recognition networks for radar compound suppression jamming signals," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15035–15045, 2020.

[10] G. Shao, Y. Chen and Y. Wei, "Convolutional neural network-based radar jamming signal classification with sufficient and limited samples," *IEEE Access*, vol. 8, pp. 80588–80598, 2020.

[11] Y. Tang, Z. Zhao, X. Ye, S. Zheng and L. Wang, "Jamming recognition based on AC-VAEGAN," in *Proc. ICSP*, Beijing, China, pp. 312–315, 2020.

[12] Q. Wu, Z. Sun and X. Zhou, "Interference detection and recognition based on signal reconstruction using recurrent neural network," in *Proc. GC Wrokshop*, Hawaii Big Island, USA, pp. 1–6, 2019.

[13] T. C. Clancy, A. Khawar and T. R. Newman, "Robust signal classification using unsupervised learning," *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1289–1299, 2011.

[14] G. Zhang, X. Yan, S. Wang and Q. Wang, "A novel automatic modulation classification for M-QAM signals using adaptive fuzzy clustering model," in *Proc. ICCWAMTIP*, Chengdu, China, pp. 45–48, 2018.

[15] F. Yang, L. Yang, D. Wang, P. Qi and H. Wang, "Method of modulation recognition based on combination algorithm of K-means clustering and grading training SVM," *China Communications*, vol. 15, no. 12, pp. 55–63, 2018.

[16] J. Norolahi, M. Mehrnia and P. Azmi, "Blind modulation classification via combined machine learning and signal feature extraction," in *Proc. ISMODE*, Jakarta, Indonesia, pp. 266–271, 2022.

[17] G. Jajoo, Y. K. Yadav and S. Yadav, "Blind signal digital modulation classification through k-medoids clustering," in *Proc. ANTS*, Indore, India, pp. 1–5, 2018.

[18] J. Mouton, M. Ferreira and A. Helberg, "A comparison of clustering algorithms for automatic modulation classification," *Expert Systems with Applications*, vol. 151, pp. 113317, 2020.

[19] L. Liu and S. Xu, "Unsupervised radar signal recognition based on multi-block multi-view low-rank sparse subspace clustering," *IET Radar Sonar and Navigation*, vol. 16, no. 3, pp. 542–551, 2018.

[20] J. Xie, R. Girshick and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. ICML*, New York City, USA, pp. 478–487, 2016.

[21] M. Caron, P. Bojanowski, A. Joulin and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. ECCV*, Munich, Germany, pp. 139–156, 2018.

[22] J. Chang, L. Wang, G. Meng, S. Xiang and C. Pan, "Deep adaptive image clustering," in *Proc. ICCV*, Venice, Italy, pp. 5880–5888, 2017.

[23] X. Ji, J. Henriques and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. ICCV*, Seoul, Korea, pp. 9864–9873, 2019.

[24] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans and L. V. Gool, "SCAN: Learning to classify images without labels," in *Proc. ECCV*, Glasgow, UK, pp. 268–285, 2020.

[25] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou *et al.,* “Contrastive clustering,” in *Proc. AAAI*, Vancouver, Canada, pp. 8547–8555, 2021.

[26] Z. Wu, Y. Xiong, S. X. Yu and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *Proc. CVPR*, Salt Lake City, USA, pp. 3733–3742, 2018.

[27] M. Ye, X. Zhang, P. C. Yuen and S. Chang, “Unsupervised embedding learning via invariant and spreading instance feature,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 6203–6212, 2019.

[28] K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. CVPR*, Seattle, USA, pp. 9726–9735, 2020.

[29] T. Chen, S. Kornblith, M. Norouzi and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. ICML*, Seattle, USA, pp. 1597–1607, 2020.

[30] S. Gidaris, P. Singh and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” 2018. [Online]. Available: https://arxiv.org/abs/1803.07728.

[31] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su *et al.,* “Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning,” in *Proc. CVPR*, Long Beach, USA, pp. 1910–1919, 2019.

[32] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *Proc. NeurIPS*, Barcelona, Spain, pp. 1857–1865, 2016.

[33] G. Huang, Z. Liu, L. Maaten and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. CVPR*, Honolulu, USA, pp. 2261–2269, 2017.