IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 14, 2020, accepted April 20, 2020, date of publication April 27, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990629

Convolutional Neural Network-Based Radar Jamming Signal Classification With Sufficient and Limited Samples

GUANGQING SHAO, YUSHI CHEN, (Member, IEEE), AND YINSHENG WEI, (Senior Member, IEEE)
School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150000, China

Corresponding author: Yushi Chen (chenyushi@hit.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61971164.

ABSTRACT Jamming is a big threat to radar system survival and anti-jamming is a part of the solution. The classification of radar jamming signal is the first step toward to anti-jamming. Recently, as an important part of deep learning, convolutional neural network (CNN) based methods have shown their capability in discriminant feature extraction and accurate classification. In this study, in order to harness the powerfulness of deep learning, CNN based methods are proposed to classify radar jamming signal acting on pulse compression radar. Specifically, a 1D-CNN is designed for radar jamming signal classification under the condition of sufficient training samples. Furthermore, due to the fact that the collection of sufficient training samples is time-consuming and expensive, a CNN-based siamese network is proposed for radar jamming signal classification to deal with the issue of limited training samples. The experimental results with sufficient and limited training samples show that the CNN-based classification methods obtain good classification performance in terms of classification accuracy and show a huge potential for radar jamming signal classification.

INDEX TERMS Radar jamming signal, convolutional neural network (CNN), sufficient and limited training samples, siamese network.

I. INTRODUCTION

Radar is playing an important role since it is widely-used in civilian and military areas. Recently, electronic warfare has become one of the most important parts of modern warfare and radar is crucial to the victory of a war. In order to disturb the enemy radar system and influence its target detection, identification and tracking ability, various radar jamming techniques have been designed.

According to the different jamming mechanism, the jamming signals can be divided into suppression jamming and deception jamming [1]. Suppression jamming covers the target signal by transmitting high-power jamming signals. Among them, noise jamming is the most widely used [2], which can be divided into aiming jamming, blocking jamming, and sweeping jamming according to the ratio between the spectrum width of jamming and the passband of the receiver. Deception jamming actively transmits radio waves of a certain phase and frequency to the enemy radar, which is used to imitate the echo of the target, so that the enemy radar gets the wrong target information [3]. Typical deception jamming includes interrupted sampling repeater jamming (ISRJ) [4], distance deception jamming [5], dense false target jamming [6], etc.

At present, the increasingly complex electromagnetic environment and jamming technology seriously affect the survival and effectiveness of radar system. Therefore, radar anti-jamming technology has been developed. A complete anti-jamming process includes radar jamming signal classification, anti-jamming strategy selection and anti-jamming performance evaluation. As a first step of anti-jamming, the accurate classification of radar jamming signal is a core part of anti-jamming system.

In recently years, the methods of radar jamming signal classification mainly include likelihood-based methods and feature-based methods. The likelihood-based methods calculate the likelihood function of the jamming signal and compare it with a certain threshold to determine the type of

The associate editor coordinating the review of this manuscript and approving it for publication was Choon Ki Ahn.

jamming. For example, Greco *et al.* [7] presented adaptive coherent estimator and generalized likelihood ratio test to solve the problem of detecting deception jamming based on digital radio frequency memory. Zhao *et al.* [8] proposed a generalized likelihood ratio test target discriminator based on the classical linear model to distinguish between target and deception jamming. However, the likelihood-based methods need prior information and expert experience. Therefore, the scope of application is limited.

The feature-based methods include feature extraction and the design of classifier. Feature extraction is the key of radar jamming signal classification. Different radar jamming signals can be transformed through the time domain, frequency domain and time-frequency domain so that the characteristics between the signals are clearly distinguished [9]–[11]. In addition, the amplitude, phase and frequency of different jamming signals are also different. Therefore, it is possible to extract the statistical features of jamming signals in different domains to distinguish different radar jamming signals. For example, Li *et al.* [12] proposed feature extraction methods based on amplitude fluctuations, high order cumulants, and bispectrum to detect deception jamming. Ma *et al.* studied statistical algorithms to extract deception jamming features [13]. Liu *et al.* [14] studied the polarization scattering characteristics of chaff jamming and classified them by support vector machines (SVM). However, the feature-based methods mainly rely on artificial feature extraction, and the process of artificial feature extraction has high computational complexity and requires a lot of manpower.

In recent years, deep learning-based methods have been proposed and achieved outstanding performance in image, text and speech processing [15]. As a representative of deep learning, CNNs have great advantages in extracting discriminant and invariable features of inputs [16]. The powerful feature extraction ability of CNN is inspired by neuroscience [17], which is reasonable in theory. Furthermore, CNN has been successfully applied in the field of radar jamming signal classification. Such as, Yun *et al.* proposed a new method of barrage jamming detection and classification for SAR based on CNN [18]. Wang *et al.* [19] designed CNN to classify active jamming. However, there are few types of jamming signals that can be distinguish by radar jamming signal classification methods based on CNN, and with the development of electronic technology, more and more jamming patterns are presented. So, it is of great significance to design a reliable CNN model which can distinguish various radar jamming signals.

Furthermore, it should not be ignored that CNN-based methods usually need lots of training samples. If the training samples are limited, the phenomenon of overfitting will appear in the CNN model. In fact, due to the complex electromagnetic environment, the acquisition of jamming samples is often very difficult and the labeling of jamming samples is also a relatively tedious thing. Therefore, it is of great practical significance to design a model that can realize the accurate classification of radar jamming signals under the condition of limited training samples. At present, siamese network is widely used to solve the problem of insufficient training samples. For example, Wang and Wang [20] proposed an improved siamese network to solve the problem of leaf classification in the case of insufficient samples. With limited training samples, Sun *et al.* [21] realized the efficient identification of voltage sag sources by designing siamese networks. Therefore, in this paper, in order to address the problem of limited jamming training samples, an improved siamese network combined with CNN is proposed too.

The main contributions of this study are listed as followed:

1) A new classification model is proposed for radar jamming signal, which is based on 1D-CNN. The proposed model obtains good classification performance in terms of overall accuracy under the condition of sufficient training samples.
2) To deal with the practical problem of insufficient samples of radar jamming signal samples, an improved Siamese-CNN (S-CNN) is proposed for radar jamming signal classification.
3) The experimental results of 12 typical radar jamming signals showed that the proposed methods can effectively classify radar jamming signals with sufficient and limited samples.

The rest of this paper is organized as follows. Section II and Section III introduces 1D-CNN and S-CNN for radar signal jamming classification, respectively. Section IV mainly introduces the experimental details, experimental results, and comprehensively analyses. Section V gives the conclusion of the whole work.

II. CNN-BASED RADAR JAMMING CLASSIFICATION

This section introduces the CNN-based radar jamming signals classification with relatively sufficient training samples. In view of the time domain characteristics of radar jamming signals, 1-D CNN is adopted to extract the hierarchical features of jamming signals. Through the feature extraction of the convolutional layers and the pooling layers, discriminant and invariant features are finally obtained, which are critical for radar jamming classification.

A. CNN-BASED FEATURE EXTRACTION

CNN mainly includes three basic parts: convolutional layers, nonlinear transformation, and pooling layers [22]. It is worth mentioning that deep CNNs can extract the input features hierarchically. With the help of local connections and shared weights, the features extracted by CNN are often invariant and robust.

A convolution layer and nonlinear transformation are defined in eq. (1) and eq. (2), respectively.

$$x_j^k = f\left(\sum_{i=1}^{N} x_i^{k-1} * w_{ij}^k + b_j^k\right) \tag{1}$$

$$f(x) = \max(0, x) \tag{2}$$

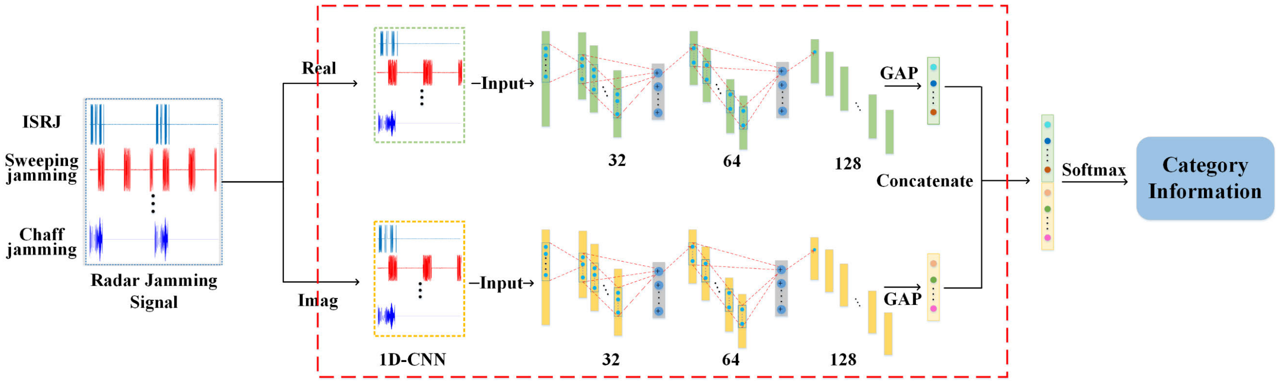


FIGURE 1. The framework of 1D-CNN model for radar jamming signal classification. The proposed framework consists of two 1D-CNN, which extract the deep features of the real and imaginary data of the input jamming samples, respectively.

Vector x_i^{k-1} is the ith feature vector of the previous (k-1)th layer, x_j^k is the jth feature vector of the current kth layer, and N is the number of input feature vectors. w_{ij}^k and b_j^k represent the weight and bias of the neuron, respectively. $*$ is the convolution operation. and $f(x)$ is a rectified linear unit (ReLU), which is used to increase the nonlinear expression ability of the network. Moreover, it can extract sparse features faster.

B. THE DESIGN OF THE 1D-CNN MODEL

The designed 1D-CNN model for the classification of radar jamming signals is shown in Fig. 1. In order to fully demonstrate the feature extraction ability of CNN, two 1D-CNNs are designed to extract the features of real and imaginary part of radar jamming data. Through convolution and pooling, the deep features of the real and imaginary parts of the radar jamming data are extracted. Finally, we concatenate the aforementioned features and send them to the softmax classifier to obtain the jamming category information.

Due to the high initial dimension of radar jamming data, the network is prone to overfitting. In order to effectively alleviate this phenomenon, dropout [22] and global average pooling (GAP) [23] are adopted in this work. Dropout makes the activation value of a certain probability p, which can make the model more generalized. In order to better match the jamming signal category with the feature map of the last convolution layer, GAP is used for replacing the traditional fully connected layers in CNN. Furthermore, GAP sums out the global spatial information, thus it is more robust to spatial translation of the jamming signal.

Meanwhile, In order to accelerate the training process, batch normalization (BN) is adopted in the 1D-CNN model. BN can maintain the same distribution of inputs at each layer of the neural network during the training process and accelerate the convergence of the network [24].

For the activation value of each neuron in the hidden layer, the BN mechanism can be formulated as follows:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \tag{3}$$

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{4}$$

$E[x^{(k)}]$ is mini-batch mean, $Var[x^{(k)}]$ is mini-batch variance. After this transformation, the activation $\hat{x}$ of a certain neuron forms a normal distribution with a mean of 0 and a variance of 1. In order to enhance the network expression, eq. (4) is carried out for the transformed activation. $\gamma^{(k)}$ and $\beta^{(k)}$ represent learnable parameters (scale and shift) [24].

III. S-CNN-BASED RADAR JAMMING CLASSIFICATION

This section introduces the S-CNN-based radar jamming signals classification with relatively limited training samples. CNNs have a powerful feature extraction capability when training samples are sufficient. However, the lack of adequate training samples is a common problem in radar jamming classification. If the training samples are insufficient, CNN often overtrains, which reduces the classification accuracy of the test samples. Due to the similarity between intraclass samples and the differences between interclass samples, S-CNN seizes this point and realizes the classification of different jamming signals with limited samples by learning the similarity between the two inputs.

A. S-CNN-BASED FEATURE EXTRACTION

The S-CNN is used to measure the similarity between the two inputs. S-CNN has two sub-networks with the same structure and weights. During training, two sub-networks extract features from two inputs, while connected neurons measure the distance between two feature vectors. The traditional classification model requires a lot of samples with labels. To some extent, S-CNN realizes the reuse of training samples by paired training. Therefore, in the case of limited training samples, S-CNN can be considered for classification. S-CNN measures the similarity of inputs through distance space, such as Manhattan distance (L1 distance) and Euclidean distance (L2 distance), and compares the new samples by learning similarity to determine the category.

S-CNN maps inputs to feature vectors by using CNN, and uses the distance between the vectors to represent the differences between the inputs. Since paired samples are used

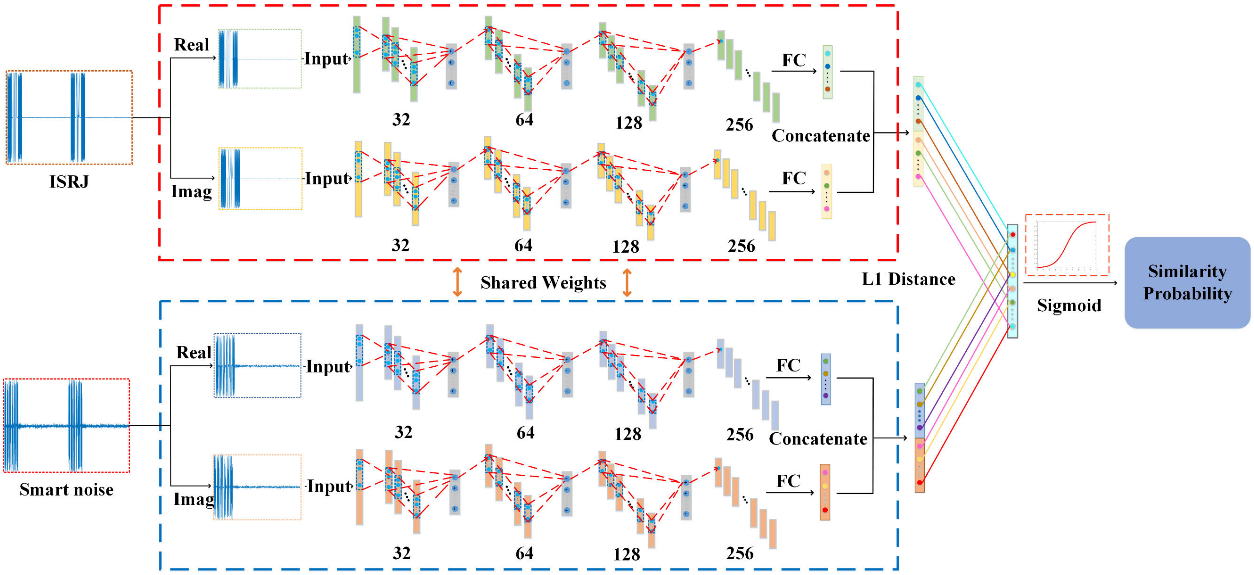


FIGURE 2. The framework of S-CNN model for radar jamming signal classification. The four 1D-CNNs with parameter sharing are used to extract features and the L1 distance of feature vectors is calculated.

to train S-CNN, it should be noted that the number of pairs of intraclass samples is consistent with the number of pairs of interclass samples. Only in this way can the S-CNN fully learn the similarity of intraclass samples and the difference of interclass samples. By training the S-CNN, the distance of the same class in the feature space is continuously reduced, and the distance of the different class is continuously increased.

S-CNN determine the category of test sample in the following way. Since the labels of the training samples are known, the test samples and training samples are paired into the trained S-CNN, and S-CNN will output the similarity probability between the test samples and the training samples. Therefore, the similarity probability between the test samples and various types of training samples can be obtained, and the category of the test samples is the category of the input training samples corresponding to the maximum similarity probability.

B. THE DESIGN OF THE S-CNN MODEL

The S-CNN model designed for the characteristics of radar jamming signals is shown in Fig. 2. First, in order to fully extract radar jamming signal features, separating the real part data and imaginary part data of paired input signals, and the four 1D-CNNs with parameter sharing are used to extract features, and concatenating the real part features and imaginary part features. Second, through the full connected layer (FC), paired inputs are finally represented by eigenvector. And then, L1 distance is calculated by using the eigenvector, and the obtained result is input into the sigmoid activation function to acquire the similarity probability P (The larger the P value, the more similar the two input signals were). Finally, S-CNN determine the category of the unknown jamming signal by the similarity probability P.

Table 1 shows the algorithm for the designed S-CNN model. In training model, L1 distance is used to calculate the distance between jamming features extracted by CNN. Let $p1, p2\}$ denote the input jamming signal pair, they go through the weight sharing CNN and get the feature representation $f(p1)$ and $f(p2)$. Finally, the L1 distance $D(p1, p2)$ is calculated, and $D(p1, p2)$ is expressed as follows:

$$D\left(p1, p2\right) = \sum_{i=1}^{n} |x_i - y_i| \tag{5}$$

n represents the dimensions of $f(p1)$ and $f(p2)$. x_i and y_i represent the element of $f(p1)$ and $f(p2)$, respectively.

The loss function of the proposed S-CNN is defined as:

$$\begin{aligned} L = &-y\left(p1, p2\right) log\sigma\left(D\left(p1, p2\right)\right) \\ &+ \left(1 - y\left(p1, p2\right)\right) log\sigma\left(1 - D\left(p1, p2\right)\right) \end{aligned} \tag{6}$$

where $y\left(p1, p2\right)$ denotes label information for training jamming signal pair. Suppose $y\left(p1, p2\right)$ [[*space*]] $= 1$ whenever $p1$ and $p2$ are from the same jamming signals class and $y\left(p1,p2\right) = 0$ otherwise. $\sigma\left(\cdot\right)$ is the sigmoidal activation function:

$$\sigma\left(x\right) = \frac{1}{1 + e^{-x}} \tag{7}$$

Due to the limited training samples for S-CNN model, the network is easy to overfit. In order to effectively avoid overfitting and accelerate the training process, L2 regularization and BN operations are adopted in the S-CNN model.

L2 regularization improves the generalization ability of the S-CNN model by punishing the weights of unimportant features [25]. By introducing L2 regularization, the loss function in this work is defined as follows:

$$\begin{aligned} L' = &-y\left(p1, p2\right) log\sigma\left(D\left(p1, p2\right)\right) \\ &+ \left(1 - y\left(p1, p2\right)\right) log\sigma\left(1 - D\left(p1, p2\right)\right) + \lambda \left\|w\right\|_2 \end{aligned} \tag{8}$$

TABLE 1. Algorithm for the designed S-CNN model.

Algorithm Iterative Training Method. N_k denotes the training samples of each class. M_k denotes the test samples of each class. k = {1, 2, …,12}. Q denotes the logarithm of all training samples used to train S-CNN. Random Training Set (Train) denotes randomly choosing Q′ pairs from Q in each iteration. y denotes the label of a pair of training samples p_i, i = {1, 2, …, Q′} and y_i=1 whenever p_i stands for homologous pair and y_i=0 otherwise. I denotes the total number of iterations.

```
Training: Train = $\{(p_1, y_1), (p_2, y_2), \ldots, (p_{Q'}, y_{Q'})\}$
for i in {1, 2, …, Q′} do
    $D_i = f(p_i)$      ⟹  Computing L1 distance
    $S_i = \sigma(D_i)$  ⟹  $\sigma(\cdot)$ is sigmoid function, $S_i$ denotes
end for                      similarity probability
for i in {1, 2, …, I} do
  $L_i \Leftarrow L_{i-1} + -y_i \times \log(D_i) - (1-y_i) \times \log(1-D_i)$
end for                      ⟹ Update Loss
Test: K is the number of jamming categories and K=12.
Support Set = $\{x_1, x_2, \ldots, x_n\}$, n = $\{1, 2, \ldots, N_1 + N_2 + \cdots + N_K\}$
Test Set = $\{t_1, t_2, \ldots, t_m\}$, m = $\{1, 2, \ldots, M_1 + M_2 + \cdots + M_K\}$
S = []
for i in $\{1, 2, \ldots, M_1 + M_2 + \cdots + M_K\}$ do
    $S_i$ = []
    for j in $\{1, 2, \ldots, N_1 + N_2 + \cdots + N_K\}$ do
        $D_j = f(\{x_j, t_i\})$
        $S_i$. append ( $\sigma(D_j)$ )
    S. append ($S_i$ )
end for
Determine the type of test samples: Test Type = []
for i in $\{1, 2, \ldots, M_1 + M_2 + \cdots + M_K\}$ do
    $S_i = [s_1, \ldots, s_{N_1}, \ldots, s_{N_1+N_2+\cdots+N_K}]$
    $T_1 = (s_1 + \cdots + s_{N_1})/N_1$, $T_2 = (s_{N_1+1} + \cdots + s_{N_1+N_2})/N_2$
    … $T_k = (s_{N_1+N_2+\cdots+N_{k-1}+1} + \cdots + s_{N_1+N_2+\cdots+N_k})/N_k$ …
    $T_K = (s_{N_1+N_2+\cdots+N_{K-1}+1} + \cdots + s_{N_1+N_2+\cdots+N_K})/N_K$
    for k in {1, 2, …, K} do
        type = 0
        if $T_k = \max\{T_1, T_2, \ldots, T_K\}$ do
          type = k
    end for
    Test type. append (type) ⟹ Save category for each test sample
end for
```

λ is the L2 regularization coefficient, which can reduce the complexity of the model and alleviate the over-fitting problem caused by limited samples. $||\cdot||_2$ represents L2 distance. w denote the weights of features.

IV. EXPERIMENT AND RESULTS

In this section, first, the radar jamming signals data set and the setting of comparative experiments are introduced. Then, the setting of 1D-CNN and S-CNN model are introduced in detail. Third, the experimental results of 1D-CNN under the condition of sufficient training samples are presented. Finally, we analyse and summarize the results of S-CNN under the condition of limited training samples.

A. DATA DESCRIPTION

In this study, we considered typical radar jamming signals currently acting on linear frequency modulation pulse (LFM) signal emitted by pulse compression radar [26]. The LFM signal was defined as follows:

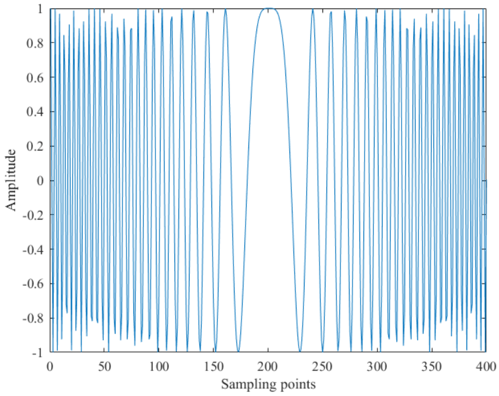


FIGURE 3. Radar target signal (real points) in this paper.

$$s(t) = \mathrm{rect}\left(\frac{t - T/2}{T}\right) \exp\left(j\pi \frac{B}{T} t^2\right) \tag{9}$$

where the pulse width T = 20μs, bandwidth B = 10MHz, and the sampling rate was 20MHz.

Then, typical radar jamming signals simulated by experts were used to evaluate the performance of the proposed method. There were 12 typical types of radar jamming signals, including suppression jamming such as aiming jamming, blocking jamming, sweeping jamming and ISRJ, distance deception jamming, dense false target jamming, smart noise jamming and typical passive jamming such as chaff jamming [14]. Finally, it also included the additive compound jamming such as ISRJ + chaff, dense false target + smart noise, and distance deception + sweeping. In the jamming data set, 500 samples were simulated for each kind of radar jamming signals, and the number of sampling points per sample was 2000 complex sampling points (2000 real points + 2000 imaginary points), and the real part and the imaginary part were separated and assembled into a row vector. Some jamming signals simulation parameters and time domain waveform were shown in Table 2 and Fig. 4, respectively. Among them, JNR represented the jamming noise ratio and others represented the corresponding radar jamming signals simulation parameters. All simulated jamming signals suppressed or deceived the target signal.

In order to satisfy the needs of the comparative experiments, experts extracted the features of the jamming signal. Extracted statistical features (SF) including skewness, kurtosis [27], normalized instantaneous amplitude frequency maximum, frequency smoothness, envelope fluctuation parameter [28], mean, and variance, which were a total of seven features.

B. EXPERIMENTAL PARAMETERS SETTINGS

1) COMPARATIVE EXPERIMENTS

In order to explore the classification effect of CNN and S-CNN on jamming signals, a series of comparative experiments were designed in this paper. We used feature data set

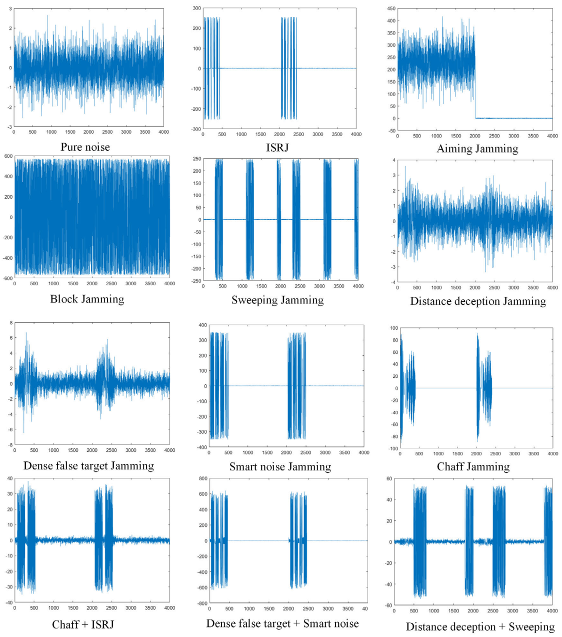


FIGURE 4. Radar jamming signals (real points + imaginary points) in this paper.

TABLE 2. Partial jamming signals core parameter range.

Type of jamming	Parameter	Typical parameter value range
ISRJ	JNR Sampling duration Pulse repetition times	30~60 dB 1~4μm 1~3
Aiming jamming	JNR Jamming bandwidth	30~60 dB 20~40 MHz
Blocking jamming	JNR Jamming bandwidth	30~60 dB 50~80 MHz
Swept jamming	JNR Sweep range Sweep cycle	30~60 dB 10MHz~10MHz 40e-6~80e-6
Distance deception jamming	False target delay range False target range	1~10μs 0.5~2
Smart noise jamming	JNR Sampling duration Pulse repetition times	30~60dB 1~4μm 1~3
Dense false target jamming	Number of false targets False target delay range False target range	3~5 1~10μs 0.5~2
Chaff jamming	Number of chaff Average Doppler frequency Doppler parity variance	1000~2000 20~50 10~20

to train support vector machine (SVM), decision tree classifiers, logistic regression and random forest (RF). In terms of parameter settings, the kernel function of SVM was radial basis function and we searched the best parameters in the way of exponentially growing sequences of$C$ and γ (The scope of C and γ is: $10^{-3} \sim 10^{3}$). We adopted the classification mode of logistic regression was one-vs-rest and we chose classification and regression tree algorithm to train decision tree. At the same time, the number of decision trees in RF was 200 and the number of features to consider when looking for the best split was set to 10. Furthermore, we compared the designed algorithm with the current 2D-CNN method applied to radar jamming classification (we referred to the network structure of [19]). Finally, we used overall accuracy (OA), and kappa coefficient (K) [29] to compare and estimate the capabilities of the proposed models. The OA was computed by eq. (10).

$$\mathrm{OA} = \frac{\text{number of correctly classified samples}}{\text{total number of test samples}} \times 100 \tag{10}$$

Suppose the number of test samples in each class was $a_1, a_2, \ldots, a_n$, and the number of samples for each type of prediction was $b_1, b_2, \ldots, b_n$. The kappa coefficient was computed by eq. (11).

$$\text{kappa} = \frac{a_1 \times b_1 + a_2 \times b_2 + \ldots + a_n \times b_n}{(\text{total number of test sample})^2} \tag{11}$$

2) 1D-CNN

In this experiment, we spilt the jamming data set into three subsets (i.e., training, validation and test samples). We randomly chose 50, 100, and 150 samples from each kind of jamming signal as training set. The validation samples were from the radar jamming data set outside the training set, and 50 samples were randomly selected from each kind of jamming signals. Then, the rest of the samples were used as the test set. The generated architecture of the 1D-CNNs for jamming data set was shown in Table 3. In the training process, the size of the mini-batch was set to 64, the dropout ratio was set to 0.5, and the number of training epochs was set to 300 for jamming data set. At the same time, the initial learning rate of all 1D-CNNs was set to 0.005, and the learning rate decreased with a step size of 75 epochs.

3) S-CNN

In this experiment, the way to split the data set was consistent with the previous setting. However, the number of training samples for each class of radar jamming signals was set to three, four, and five. The architecture of the S-CNN for radar jamming data set was shown in Table 4. In the training process, the number of training iterations was 300 and the number of logarithms to train S-CNN in one iteration was 12. Meanwhile, the size of the mini-batch was set to 12 and L2 regularization weight was set to 2×10^{-4}. At the same time, the S-CNN model adopted adam algorithm [30], and its learning rate was set to 0.0001.

4) EXPERIMENTAL ENVIRONMENT

The platform for jamming data generation was MATLAB 2018. All the experiments were run on pycharm-community-2017.2.1 and a 2.30 GHz CPU with a GTX 960M GPU card. Furthermore, all the algorithms were implemented with Keras 2.1.0, Scikit-learn 0.18.0, Numpy 1.15.1, and Scipy 0.19.0.

TABLE 3. The architecture of the designed 1D-CNN model.

Layer	Convolution	Pooling	Activation	Dropout
Input	data dimension is 2000×1			
Conv1	5×1×32	2×1	ReLU	No
Conv2	7×32×64	2×1	ReLU	50%
Conv3	9×64×128	2×1	ReLU	50%
GAP	data dimension is 128×1			
Concatenate	data dimension is 256×1			

TABLE 4. The architecture of the designed S-CNN model.

Layer	Convolution	Units	Pooling	Activation	L2 Regularization	Dropout	BN
Input	data dimension is 2000×1						
Conv1	5×1×32	——	2×1	ReLU	Yes	No	Yes
Conv2	7×32×64	——	2×1	ReLU	Yes	No	No
Conv3	9×64×128	——	2×1	ReLU	Yes	No	No
Conv4	9×128×128	——	2×1	ReLU	Yes	Yes	No
FC1	——	256	——	ReLU	No	No	No
Concatenate	data dimension is 512×1						
Sigmoid	——	1	——	Sigmoid	Yes	No	No

TABLE 5. Classification results (values ± standard deviation) of various classification models under sufficient training samples.

Method Training samples of each class		SVM-RBF	Decision tree	SF-SVM-RBF	RF-200	SF-Decision tree	SF-RF-200	2D-CNN	**1D-CNN**
50	OA (%)	60.78 ± 1.78	62.77 ± 0.96	79.06 ± 0.44	80.69 ± 0.30	85.97 ± 0.91	87.80 ± 0.65	87.83 ± 1.84	**91.95 ± 2.19**
	K×100	57.21 ± 1.94	59.39 ± 1.04	77.15 ± 0.48	78.93 ± 0.33	84.69 ± 0.99	86.69 ± 0.71	86.72 ± 2.01	**91.21 ± 2.39**
100	OA (%)	68.04 ± 0.91	67.26 ± 0.76	79.77 ± 0.43	83.77 ± 0.36	87.81 ± 0.94	88.63 ± 0.45	89.88 ± 0.47	**96.97 ± 0.88**
	K×100	65.13 ± 1.00	64.28 ± 0.83	77.93 ± 0.47	82.29 ± 0.40	86.70 ± 1.02	87.59 ± 0.49	88.97 ± 0.52	**96.69 ± 0.96**
150	OA (%)	71.16 ± 0.70	71.14 ± 0.72	80.22 ± 0.66	85.61 ± 0.25	88.35 ± 0.16	89.64 ± 0.43	91.33 ± 1.29	**97.34 ± 0.37**
	K×100	68.54 ± 0.76	68.51 ± 0.78	78.42 ± 0.71	84.31 ± 0.27	87.29 ± 0.17	88.70 ± 0.47	90.55 ± 1.40	**97.10 ± 0.41**
	Pure noise	81.49 ± 4.11	72.97 ± 4.62	**100.00 ± 0.00**	**100.00 ±0.00**	**100.00 ±0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
	ISRJ	66.06 ± 4.00	47.31 ± 2.86	52.00 ± 20.61	65.83 ± 3.28	83.20 ± 5.44	86.74 ± 3.10	86.40 ± 8.45	**99.93 ± 0.13**
	Aiming jamming	**100.00 ± 0.00**	96.97 ± 1.85	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
	Blocking jamming	**100.00 ± 0.00**	84.57 ± 1.92	56.29 ± 4.35	**100.00 ± 0.00**	64.06 ± 4.78	64.17 ± 3.78	96.60 ± 6.80	**100.00 ± 0.00**
	Sweeping jamming	73.09 ± 3.28	98.34 ± 1.24	64.34 ± 4.19	**100.00 ± 0.00**	58.06 ± 5.55	57.83 ± 5.23	98.50 ± 1.82	**100.00 ± 0.00**
	Distance deception jamming	79.26 ± 2.92	47.89 ± 4.89	99.94 ± 0.11	85.77 ± 2.12	99.89 ± 0.23	**100.00 ± 0.00**	98.90 ± 0.80	92.87 ± 4.19
	Dense false target jamming	81.89 ± 2.65	58.29 ± 5.11	99.26 ± 0.39	91.60 ± 1.18	**99.77 ± 0.46**	99.43 ± 0.65	88.80 ± 9.52	83.67 ± 3.26
	Smart noise jamming	24.17 ± 1.60	38.29 ± 5.37	71.83 ± 22.15	42.69 ± 3.76	77.66 ± 1.45	83.37 ± 3.08	66.10 ± 13.84	**98.27 ± 2.81**
	Chaff jamming	99.43 ± 0.31	94.63 ± 1.48	99.66 ± 0.33	**100.00 ± 0.00**	98.97 ± 1.12	99.37 ± 0.33	**100.00 ± 0.00**	**100.00 ± 0.00**
	Chaff + ISRJ	22.97 ± 1.67	33.14 ± 5.17	64.97 ± 2.88	41.49 ± 2.37	89.02 ± 2.48	93.09 ± 0.91	63.30 ± 12.66	**96.07 ± 4.37**
	Dense false target + smart noise	52.69 ± 3.02	84.74 ± 2.62	54.51 ± 3.35	**100.00 ± 0.00**	89.60 ±1.65	91.66 ± 1.28	98.40 ± 0.97	97.27 ± 0.65
	Distance deception + sweeping	72.63 ± 2.13	96.80 ± 1.01	99.88 ± 0.23	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	99.00 ± 1.30	**100.00 ± 0.00**

C. THE CLASSIFICATION RESULTS OF THE 1D-CNN MODEL

Table 5 showed the classification results of different classification models under different training samples. In all models, the 1D-CNN model we designed achieved the optimal OA and K under sufficient training samples. For example, when the number of training samples of each class was 50, 100, and 150, the OA of the designed 1D-CNN model was 91.95 ± 2.19%, 96.97± 0.88%, and 97.34 ± 0.37%, which was 4.12%, 7.09%, and 6.01% higher than the optimal comparative experimental results, respectively. At the same time, the average accuracy of the designed 1D-CNN for nine kinds of jamming signals reached the optimal value under the

TABLE 6. Confusion matrix for the proposed 1D-CNN at JNR of 30-60dB.

	Pure noise	ISRJ	Aiming jamming	Blocking jamming	Sweeping jamming	Distance deception jamming	Dense false target jamming	Smart noise	Chaff jamming	Chaff + ISRJ	Dense false target + smart noise	Distance deception + sweeping
Pure noise	300	0	0	0	0	0	0	0	0	0	0	0
ISRJ	0	300	0	0	0	0	0	0	0	0	0	0
Aiming jamming	0	0	300	0	0	0	0	0	0	0	0	0
Blocking jamming	0	0	0	300	0	0	0	0	0	0	0	0
Sweeping jamming	0	0	0	0	300	0	0	0	0	0	0	0
Distance deception jamming	7	0	0	0	0	279	14	0	0	0	0	0
Dense false target jamming	0	0	0	0	0	41	259	0	0	0	0	0
Smart noise	0	0	0	0	0	0	0	290	0	10	0	0
Chaff jamming	0	0	0	0	0	0	0	0	300	0	0	0
Chaff + ISRJ	0	0	0	0	0	0	0	24	0	276	0	0
Dense false target + smart noise	0	6	0	0	0	0	0	0	0	0	294	0
Distance deception + sweeping	0	0	0	0	0	0	0	0	0	0	0	300

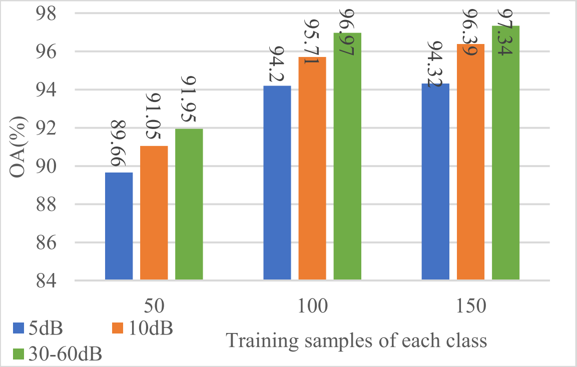


FIGURE 5. Classification results of the designed 1D-CNN under different JNR.

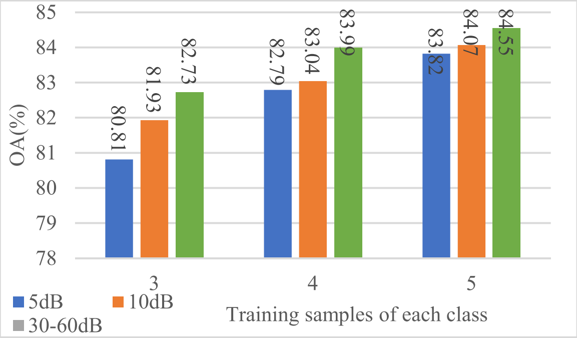


FIGURE 6. Classification results of the designed S-CNN under different JNR.

condition of 150 training samples for each type of jamming signals. For example, when the number of training samples of each class was 150, most classification models had poor classification of the smart noise jamming, but the OA of 1D-CNN was 98.27% ± 2.81%.

Since the range of JNR in the simulated radar jamming data set was 30-60dB, the jamming signals were not affected much by the noise. However, in the complex electromagnetic environment, the jamming signals were mixed with a lot of noise, which affected the performance of the classification model. Therefore, in order to explore the anti-noise ability of the designed 1D-CNN model, the JNR of radar jamming data was changed, and the experimental results were shown in Fig. 5. Under the condition of different training samples, the JNR was reduced to 5dB and 10dB (white Gaussian noise was mainly introduced), and the classification accuracy of the 1D-CNN model did not decrease significantly. For example, when the number of training samples of each class was 150, the OA of the designed 1D-CNN with the JNR of 5dB and 10dB was lower 3.02% and 0.95% than the designed 1D-CNN model without changing the JNR, respectively. This showed that the designed 1D-CNN model in this paper had better anti-noise ability.

Table 6 showed the confusion matrix for the proposed 1D-CNN at JNR of 30-60dB. It can be seen from the table that the proposed 1D-CNN algorithm had good classification performance for radar jamming signals when the training samples were sufficient. According to the confusion matrix, classification errors mainly occurred on jamming signals with similar jamming mechanism and simulation parameters, such as, distance deception jamming and dense false target jamming. This was mainly because the simulation range of false target parameters of the two types of radar jamming signals was consistent (the number of false targets was different), which led to the correlation and confusion between the two types of jamming signals.

D. THE CLASSIFICATION RESULTS OF THE S-CNN MODEL

Table 7 showed the classification results of different classification models with limited training samples. In all classification models, the designed S-CNN had achieved the best OA

TABLE 7. Classification results (values ± standard deviation) of various classification models under limited training samples.

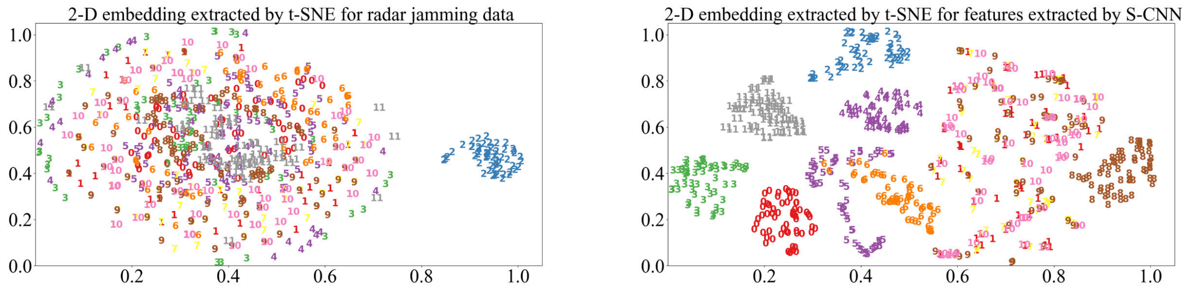
Method Training samples of each class		SVM-RBF	2D-CNN	RF-200	1D-CNN	SF-SVM-RBF	SF-Decision tree	SF-RF-200	**S-CNN**
3	OA (%)	30.47 ± 1.91	45.63 ± 3.39	48.91 ± 2.10	65.65 ± 3.72	66.49 ± 2.12	76.84 ± 2.56	78.57 ± 3.16	**82.73 ± 3.67**
	K×100	24.14 ± 2.08	40.69 ± 3.70	44.27 ± 2.29	62.52 ± 4.06	63.44 ± 2.31	74.74 ± 2.79	76.62 ± 3.45	**81.16 ± 4.00**
4	OA (%)	33.71 ± 2.59	47.16 ± 5.10	53.74 ± 1.12	66.60 ± 2.45	67.53 ± 0.73	75.83 ± 3.12	79.86 ± 1.14	**83.99 ± 2.49**
	K×100	27.68 ± 2.83	42.36 ± 5.56	49.54 ± 1.22	63.56 ± 2.67	64.58 ± 0.80	73.63 ± 3.40	78.02 ± 1.24	**82.54 ± 2.71**
5	OA (%)	38.89 ± 2.39	49.47 ± 3.33	54.10 ± 2.08	68.01 ± 5.79	70.62 ± 2.56	76.40 ± 3.69	81.01 ± 0.95	**84.55 ± 2.08**
	K×100	33.33 ± 2.60	44.88 ± 3.63	49.93 ± 2.27	65.10 ± 6.32	67.95 ± 2.80	74.25 ± 4.03	79.28 ± 1.04	**83.15 ± 2.27**
	Pure noise	4.16 ± 4.07	0.00 ± 0.00	71.27 ± 7.35	80.00 ± 40.00	**100.00 ± 0.00**	95.35 ± 9.29	**100.00 ± 0.00**	98.90 ± 0.61
	ISRJ	27.72 ± 8.37	9.22 ± 12.91	15.07 ± 2.89	46.69 ± 7.70	61.17 ± 13.92	59.43 ± 12.59	51.52 ± 12.15	**74.37 ± 8.50**
	Aiming jamming	**100.00 ± 0.00**	53.59 ± 30.46	99.60 ± 0.81	99.63 ± 0.73	99.07 ± 1.34	95.35 ± 5.69	**100.00 ± 0.00**	**100.00 ± 0.00**
	Blocking jamming	**100.00 ± 0.00**	52.20 ± 43.11	70.83 ± 15.72	92.16 ± 10.26	56.12 ± 13.10	55.64 ± 20.83	44.12 ± 12.38	**100.00 ± 0.00**
	Sweeping jamming	43.84 ± 28.22	42.94 ± 19.94	95.84 ± 6.62	44.82 ± 14.51	48.16 ± 17.04	42.79 ± 22.63	58.58 ± 19.00	**100.00 ± 0.00**
	Distance deception jamming	2.91 ± 2.01	38.65 ± 47.39	27.03 ± 3.76	33.88 ± 27.69	98.34 ± 2.64	99.03 ± 0.87	**99.68 ± 0.27**	94.94 ± 2.15
	Dense false target jamming	23.68 ± 7.45	68.78 ± 40.68	27.39 ± 8.66	**99.02 ± 1.66**	90.75 ± 7.78	88.40 ± 7.54	96.36 ± 1.89	93.18 ± 3.45
	Smart noise jamming	13.54 ± 6.34	39.80 ± 15.38	15.39 ± 5.48	55.51 ± 11.95	35.76 ± 13.50	36.57 ± 25.32	**61.33 ± 6.83**	54.78 ± 14.93
	Chaff jamming	87.56 ± 5.38	99.76 ± 0.49	**100.00 ± 0.00**	95.71 ± 1.40	**100.00 ± 0.00**	92.24 ± 3.62	94.99 ± 5.88	95.27 ± 3.81
	Chaff + ISRJ	11.27 ± 5.19	44.00 ± 8.22	9.86 ± 3.98	46.20 ± 8.77	48.04 ± 11.14	**77.62 ± 17.09**	74.87 ± 6.14	37.27 ± 12.14
	Dense false target + smart noise	21.05 ± 5.85	69.22 ± 22.93	20.65 ± 4.60	61.18 ± 5.94	52.61 ± 5.65	78.91 ± 13.96	**90.67 ± 2.75**	65.96 ± 6.45
	Distance deception + sweeping	30.91 ± 16.85	75.47 ± 7.36	96.32 ± 5.25	61.31 ± 15.20	57.45 ± 22.62	95.43 ± 5.52	**100.00 ± 0.00**	**100.00 ± 0.00**

TABLE 8. Confusion matrix for the proposed S-CNN at JNR of 30-60dB.

	Pure noise	ISRJ	Aiming jamming	Blocking jamming	Sweeping jamming	Distance deception jamming	Dense false target jamming	Smart noise	Chaff jamming	Chaff + ISRJ	Dense false target + smart noise	Distance deception + sweeping
Pure noise	487	0	0	0	0	3	0	0	0	0	0	0
ISRJ	0	470	0	0	0	0	0	0	0	9	11	0
Aiming jamming	0	0	490	0	0	0	0	0	0	0	0	0
Blocking jamming	0	0	0	490	0	0	0	0	0	0	0	0
Sweeping jamming	0	0	0	0	490	0	0	0	0	0	0	0
Distance deception jamming	2	0	0	0	0	450	38	0	0	0	0	0
Dense false target jamming	0	0	0	0	0	13	476	0	1	0	0	0
Smart noise	0	0	0	0	0	0	0	279	0	211	0	0
Chaff jamming	0	0	0	0	0	0	0	0	483	0	7	0
Chaff + ISRJ	0	0	0	0	0	0	0	255	0	235	0	0
Dense false target + smart noise	0	211	0	0	0	0	0	102	0	0	177	0
Distance deception + sweeping	0	0	0	0	0	0	0	0	0	0	0	490

and K. For example, when the number of training samples of each class was 3, 4, and 5, the OA of the designed S-CNN model was 82.73 ± 3.67%, 83.99± 2.49%, and 84.55 ± 2.08%, which was 4.16%, 4.13%, and 3.54% higher than the optimal comparative experimental results, respectively. At the same time, in all classification models, the OA of the designed S-CNN for five kinds of jamming signal reached the optimal value under the condition of 5 training samples for each type of jamming signals. Such as, most classification models had poor classification accuracy for sweeping jamming signals, but the OA of S-CNN was 100% ± 0.00%.

Meanwhile, the anti-noise ability of the designed S-CNN model was analyzed. The change in JNR was consistent with previous experiments and the experimental results were shown in Fig. 6. Under the condition of different training samples, the JNR was reduced to 5dB and 10dB, and the classification accuracy of the S-CNN model did not decrease significantly. For example, when the number of training samples of each class was 5, the OA of the designed S-CNN with the JNR of 5dB and 10dB was lower 0.73% and 0.48% than the designed S-CNN model without changing the JNR, respectively. This showed that the



(a)

(b)

FIGURE 7. **Jamming data set: 2-D embedding extracted by t-SNE for test data (a) and test data features (b) extracted by t-SNE.**

designed S-CNN model in this study had better anti-noise ability.

Table 8 showed the confusion matrix for the proposed S-CNN at JNR of 30-60dB. It can be seen from the table that the proposed S-CNN algorithm had good classification performance for radar jamming signals. According to the confusion matrix, classification errors mainly occurred on jamming signals with similar jamming mechanism. Since smart noise was generated by adding noise frequency modulation on the basis of ISRJ, ISRJ, smart noise, chaff + ISRJ, and dense false target + smart noise were easy to be confused in time domain. Referring to the confusion matrix for the proposed 1D-CNN, this phenomenon was more obvious when the training samples were limited.

To better understand the classification power of the designed S-CNN model, we randomly selected 60 labeled samples per class from jamming data set and used t-SNE [32] algorithm to reduce the dimensionality of inputs and features extracted by S-CNN to two. The results were visualized in Fig. 7, where different colors represented different classes in jamming data set. It was obvious from the Fig. 7 (a) that the original inputs of different jamming classes were confused with each other. Then, these labeled samples were input into the trained S-CNN to extract the features of full connection layer. The feature visualization of the radar jamming signals extracted by the designed S-CNN was shown in Fig.7 (b). It can clearly see that the features extracted by S-CNN significantly reduced the distance between intraclass samples and further increased the separability between interclass samples.

V. CONCLUSION

In this paper, we proposed a new radar jamming signal classification model based on 1D-CNN and made full use of the hierarchical feature extraction ability of 1D-CNN. Through experiments, the method based on 1D-CNN showed strong ability in feature extraction and accurate classification when the training samples were sufficient. At the same time, in order to solve the problem of limited training samples, a radar jamming signal classification model based on S-CNN was proposed. The proposed S-CNN fully showed the ability to measure similarities between jamming signals. Compared with other classification models, both designed 1D-CNN model and S-CNN model achieved the optimal classification performance for a variety of typical radar jamming types. Meanwhile, the proposed radar jamming signal classification models had good anti-noise ability. The experimental results indicated the effectiveness of deep learning model, and it had great potential in radar jamming signal classification.

REFERENCES

[1] L. Neng-Jing and Z. Yi-Ting, "A survey of radar ECM and ECCM," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 3, pp. 1110–1120, Jul. 1995.

[2] D. Orlando, "A novel noise jamming detection algorithm for radar applications," *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 206–210, Feb. 2017.

[3] A. Farina, "Electronic counter-countermeasures," in *Radar Handbook*, 3rd ed. M. I. Skolnik, Ed. New York, NY, USA: McGraw-Hill, 2008.

[4] Q. Wu, F. Zhao, X. Ai, X. Liu, and S. Xiao, "Two-dimensional blanket jamming against ISAR using nonperiodic ISRJ," *IEEE Sensors J.*, vol. 19, no. 11, pp. 4031–4038, Jun. 2019.

[5] R. Shi and J. Xu, "Multipath effect analysis and pre-distortion processing for jamming on wideband ground radar through antenna sidelobe," *IET J. Eng.*, vol. 2019, no. 19, pp. 5672–5676, Oct. 2019.

[6] C. Wen, J. Peng, Y. Zhou, and J. Wu, "Enhanced three-dimensional joint domain localized STAP for airborne FDA-MIMO radar under dense false-target jamming scenario," *IEEE Sensors J.*, vol. 18, no. 10, pp. 4154–4166, May 2018.

[7] M. Greco, F. Gini, and A. Farina, "Radar detection and classification of jamming signals belonging to a cone class," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1984–1993, May 2008.

[8] S. Zhao, Y. Zhou, L. Zhang, Y. Guo, and S. Tang, "Discrimination between radar targets and deception jamming in distributed multiple-radar architectures," *IET Radar, Sonar Navigat.*, vol. 11, no. 7, pp. 1124–1131, Jul. 2017.

[9] L. Xu, D. Feng, Y. Liu, X. Pan, and X. Wang, "A three-stage active cancellation method against synthetic aperture radar," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6173–6178, Nov. 2015.

[10] Y. Liu, D. Feng, X. Pan, D. Dai, and W. Wang, "A frequency-domain three-stage algorithm for active deception jamming against synthetic aperture radar," *IET Radar, Sonar Navigat.*, vol. 8, no. 6, pp. 639–646, Jul. 2014.

[11] L. Du, L. Li, B. Wang, and J. Xiao, "Micro-Doppler feature extraction based on time-frequency spectrogram for ground moving targets classification with low-resolution radar," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3756–3763, May 2016.

[12] L. Jian-xun, S. Qi, and Y. Hai, "Signal feature analysis and experimental verification of radar deception jamming," in *Proc. IEEE CIE Int. Conf. Radar*, Oct. 2011, pp. 230–233.

[13] X. Ma, J. Qin, and J. Li, "Pattern recognition-based method for radar anti-deceptive jamming," *Syst. Eng. Electron.*, vol. 16, no. 4, pp. 802–805, Dec. 2005.

[14] Y. Liu, S. Xing, Y. Li, D. Hou, and X. Wang, "Jamming recognition method based on the polarisation scattering characteristics of chaff clouds," *IET Radar, Sonar Navigat.*, vol. 11, no. 11, pp. 1689–1699, Nov. 2017.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[16] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1106–1114.

[18] Y. Junfei, L. Jingwen, S. Bing, and J. Yuming, "Barrage jamming detection and classification based on convolutional neural network for synthetic aperture radar," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 3281–3284.

[19] Y. Wang, B. Sun, and N. Wang, "Recognition of radar active-jamming through convolutional neural networks," *IET J. Eng.*, vol. 2019, no. 21, pp. 7695–7697, Nov. 2019.

[20] B. Wang and D. Wang, "Plant leaves classification: A few-shot learning method based on siamese network," *IEEE Access*, vol. 7, pp. 151754–151763, Oct. 2019.

[21] H. Sun, H. Yi, G. Yang, F. Zhuo, and A. Hu, "Voltage sag source identification based on few-shot learning," *IEEE Access*, vol. 7, pp. 164398–164406, Nov. 2019.

[22] G. Hinton, N. Srivastava, A. Krizhevsky, R. R. Salakhutdinov, and I. Sutskever, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, Jul. 2012.

[23] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, pp. 448–456, 2015.

[25] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by Nuclear/l2,1-norm regularization for image feature extraction," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1607–1622, Apr. 2017.

[26] J. Dai, X. Hao, P. Li, Z. Li, and X. Yan, "Antijamming design and analysis of a novel pulse compression radar signal based on radar identity and chaotic encryption," *IEEE Access*, vol. 8, pp. 5873–5884, Jan. 2020.

[27] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.

[28] A. N. Gavrilov and I. M. Parnum, "Fluctuations of seafloor backscatter data from multibeam sonar systems," *IEEE J. Ocean. Eng.*, vol. 35, no. 2, pp. 209–219, Apr. 2010.

[29] M. N. Sumaiya and R. Shantha Selva Kumari, "Logarithmic mean-based thresholding for SAR image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 11, pp. 1726–1728, Nov. 2016.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[31] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

GUANGQING SHAO is currently pursuing the master's degree with the Department of Information Engineering, School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China. His research interests include radar signal classification, machine learning, and deep learning.

YUSHI CHEN (Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2008.

He is currently an Associate Professor with the School of Electronics and Information Engineering, Harbin Institute of Technology. His research interests include remote sensing, data processing, and machine learning.

YINSHENG WEI (Senior Member, IEEE) was born in Heilongjiang, China, in 1974. He received the M.S. and Ph.D. degrees in communication and information systems from the Harbin Institute of Technology, Harbin, China, in 1998 and 2002, respectively.

He joined the Department of Electronics Engineering, Harbin Institute of Technology, as a Lecturer, and became a Professor, in 2011. His main research interests include radar signal processing, and radar system analysis and simulation.