remote sensing

Article

Radar Active Jamming Recognition under Open World Setting

Yupei Zhang [1], Zhijin Zhao [2,*] and Yi Bu [3]

[1] School of Electronic and Information, Hangzhou Dianzi University, Hangzhou 310018, China; 2143040036@hdu.edu.cn
[2] School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China
[3] School of Electrical and Computer Engineering, Royal Melbourne Institute of Technology (RMIT University), Melbourne 3000, Australia; prototyperimac@gmail.com
* Correspondence: zhaozj03@hdu.edu.cn

Abstract: To address the issue that conventional methods cannot recognize unknown patterns of radar jamming, this study adopts the idea of zero-shot learning (ZSL) and proposes an open world recognition method, RCAE-OWR, based on residual convolutional autoencoders, which can implement the classification of known and unknown patterns. In the supervised training phase, a residual convolutional autoencoder network structure is first constructed to extract the semantic information from a training set consisting solely of known jamming patterns. By incorporating center loss and reconstruction loss into the softmax loss function, a joint loss function is constructed to minimize the intra-class distance and maximize the inter-class distance in the jamming features. Moving to the unsupervised classification phase, a test set containing both known and unknown patterns is fed into the trained encoder, and a distance-based recognition method is utilized to classify the jamming signals. The results demonstrate that the proposed model not only achieves sufficient learning and representation of known jamming patterns but also effectively identifies and classifies unknown jamming signals. When the jamming-to-noise ratio (JNR) exceeds 10 dB, the recognition rate for seven known jamming patterns and two unknown jamming patterns is more than 92%.

Keywords: radar signals; jamming recognition; deep learning; zero-shot learning; convolutional neural network

Citation: Zhang, Y.; Zhao, Z.; Bu, Y. Radar Active Jamming Recognition under Open World Setting. *Remote Sens.* **2023**, *15*, 4107. https://doi.org/10.3390/rs15164107

Academic Editors: Qian Du, Yanni Dong and Xiaochen Yang

Received: 30 June 2023
Revised: 17 August 2023
Accepted: 18 August 2023
Published: 21 August 2023

1. Introduction

Radar faces increasingly complex electronic countermeasures, with various new types of radar jamming patterns continuously emerging as challenges [1,2]. The accurate recognition of radar jamming is a precondition and key to implementing anti-jamming measures, and automatic recognition of radar jamming patterns can effectively improve the target detection and tracking performance of radar. Therefore, jamming pattern recognition has always been a research hotspot in anti-jamming technology [3,4]. Suppression jamming by emitting high-power noise signals is not only effective for linear frequency modulated (LFM) radar systems but also for other modulated radar systems, making it the most widely used [5]. Therefore, this paper focuses on the recognition of suppression jamming. Its main patterns include amplitude modulation jamming (AMJ), frequency modulation jamming (FMJ), comb spectrum jamming (CJ), phase modulation jamming (PMJ), swept jamming (SJ), etc. Conventional jamming pattern detection and classification methods are generally based on feature engineering. Firstly, multi-dimensional features of signal in time domain, frequency domain, and transform domain are extracted, including features such as moment kurtosis, moment skewness, envelope fluctuation, noise factor [6–9], singular spectrum features [10], bispectrum features [11] and other signal features. Then, with the help of machine learning-based classifiers, such as support vector machine (SVM) [12], decision trees [13], and back propagation (BP) neural networks [12], classifications are accomplished. However, feature engineering-based classification methods are time-consuming and require expert experience, especially when the transform domain features are large, the classification performance has limited room for improvement, and the recognition rate is low in strong noise environment.

With the development of deep learning (DL) technology, the feature extraction capability of neural networks has been improving, and classification methods based on DL are emerging. In [14], A novel hybrid framework of optimized deep learning models combined with multi-sensor fusion is developed for condition diagnosis of concrete arch beam, and the results demonstrate that the method can achieve the classification of structural damage with limited sensors and high levels of uncertainties.

Convolutional neural networks (CNNs), due to their network architecture which incorporates weight sharing and small local receptive fields, have significantly reduced the number of node connections compared to conventional neural networks. This simplification of network connections has made CNNs widely applied in deep learning models [15–17]. CNNs can train their parameters using jamming signals, eliminating the need for manual feature extraction and the design of decision trees for classification criteria. As a result, CNNs have been extensively used in the research of classifying and recognizing radar jamming signals [18].

In [19], a jamming recognition algorithm based on improved LeNet·CNN network was designed which extracted one-dimensional radar received signals and adjusted the network structure parameters to achieve optimal performance for the recognition of jamming signals. Ref. [20] obtained the time-frequency spectrogram of jamming signals by short-time Fourier transform, combined with the improved VGGNet·16 network model for feature learning and training, and the simulation verified that the algorithm is still effective for the identification of six kinds of mixed jamming. Ref. [21] adopted an adaptive cropping algorithm to crop most of the redundant information of the time-frequency image and kept the complete information of the jamming in the CNN for training, and finally achieved the recognition of nine kinds of jamming signals with high accuracy and fast iteration. In [22], a 1D CNN-based radar jamming signal classification model was proposed to achieve the classification of 12 typical jamming signals by putting the real and imaginary parts of jamming signals into the parallel network for training. In [23], a CNN was constructed using the real and imaginary parts of the signal as inputs. With sufficient training samples, this method demonstrated excellent recognition capabilities. The mentioned papers primarily address the issue of recognizing jamming when there are sufficient labeled samples. However, [6,24,25] considered the case of insufficient labeled samples. Ref. [6] proposed a method based on a time-frequency self-attentive mechanism. The recognition rate for most of the patterns of jamming reaches 90% when the samples with labels account for 3%. In [24], Tian et al. inputed features obtained through empirical mode decomposition and wavelet decomposition into the network. Simulations conducted on a dataset consisting of only 8400 samples showed that the recognition rates for four types of jamming were all above 90% when the JNR exceeded 6 dB. In [25], a large number of unlabeled samples were first used to train an jamming recognition network to extract valuable features. Then, a small number of labeled samples were used to improve the classification accuracy.

Although the application of deep learning technology in radar jamming recognition is rapidly developing, the current methods still suffer from the closed-set assumption, i.e., the existing methods assume that the jamming patterns are included in the training set. However, in the actual battlefield environment, the enemy may invent new jamming patterns, making it challenging to collect data for all patterns in the training set, so the actual radar jamming environment is an open set scenario, i.e., the test environment is likely to have jamming patterns that do not exist in the training jamming library. In the actual open set jamming scenario, when a jamming pattern that does not exist in the training jamming library appears in the test environment, the existing radar jamming identification methods will incorrectly identify this unknown jamming as one of the known jamming patterns. In [26], Zhou et al. first investigated the open set recognition problem for radar jamming; however, the method can only detect or reject the unknown patterns, but cannot effectively identify the unknown patterns. How to further classify these unknown pattern signals remains a challenging task, and it falls under the research domain of open world recognition (OWR).

Currently, OWR techniques have been applied in target recognition of synthetic aperture radar (SAR) images. In [27], a hierarchical embedding and incremental evolutionary network (HEIEN) was designed for when there are fewer unknown target training sets in open scenarios, which requires only a small number of unknown target samples for effective model training. A more stable feature space was built in [28], which has better interpretability. In testing, experiments on a dataset containing seven known targets and one unknown target show that the method improves the reliability of recognizing unknown targets. In [29], Song et al. used physical EM simulation images of targets at different azimuths as training data in order to learn the features of unknown targets. An accuracy of 91.93% can be achieved in a recognition task with a dataset containing nine known targets and one unknown target.

Nevertheless, OWR is just starting in radar jamming pattern recognition. Zero-shot learning (ZSL) [30,31] is an effective approach to address the challenge of open world recognition. The most typical implementation of ZSL is based on feature mapping. The goal of this approach is to learn the mapping relationship between the original signal space and the semantic feature space using the known jamming patterns. This learned relationship is then generalized to the unknown pattern dataset, enabling recognition and classification of unknown patterns using the semantic features. ZSL can be classified into two types: traditional zero-shot learning (TZSL) and generalized zero-shot learning (GZSL) [32]. TZSL assumes that the training patterns are known, while the testing patterns are unknown, and there is mutual exclusion between the training and testing patterns. GZSL assumes that the training patterns include both known and unknown patterns in the testing phase. GZSL has a more relaxed experimental setting, which better reflects real-world scenarios. Therefore, in this paper, we consider the GZSL scenario.

To address the challenge of existing methods being unable to classify unknown jamming patterns specifically, this paper adopts the idea of ZSL and conducts research on jamming patterns recognition in an open world scenario. We propose a residual convolutional autoencoder-based radar jamming open world recognition algorithm, abbreviated as RCAE-OWR.

In summary, the main contributions of this paper are as follows:

- In order to address the limitations of existing radar jamming pattern recognition methods, which are mostly closed-set recognition or simply rejecting unknown patterns, we propose a zero-shot learning approach based on residual conventional autoencoders. This method does not require prior information about the patterns of jamming and can classify both known and unknown patterns using distance-based recognition methods.
- A hybrid loss function consisting of cross-entropy loss, center loss and reconstruction loss is introduced to recognize different patterns of jamming signals. Where the cross-entropy loss makes the features obtained from the mapping network divisible, and to some extent widens the distance among different patterns, the center loss makes it easier to delineate the boundaries of the various patterns, and the reconstruction loss ensures that the most essential characterization of the pattern features is learned from the known patterns dataset.
- Through extensive experimental simulations, we evaluate the open world recognition performance of the proposed algorithm and investigate the influence of JNR and the number of unknown patterns on the algorithm's performance. The simulation results demonstrate that the proposed algorithm achieves effective recognition of both known and unknown patterns, especially in high-JNR environments.

2. Jamming Signal Modeling

In radar jamming and anti-jamming training, the signals received by the radar receiver include radar echo signals, jamming signals and noise signals, which can be expressed as follows:

$$S(t) = s_o(t) + J(t) + v(t), \tag{1}$$

where $S(t)$ denotes the total signal received by the radar, $s_o(t)$ represents the echo signal, $J(t)$ means the jamming signal, and $v(t)$ is the noise signal.

2.1. Echo Signal

Typical modulation types of radar echo signals include continuous wave (CW), linear frequency modulation (LFM), phase shift keying (PSK), frequency shift keying (FSK), etc. LFM, with its frequency linearly changing over time, is the most widely used. Therefore, in this paper, we focus on studying the typical echo signal of LFM, which can be expressed as follows [33]:

$$s_o(t) = \sum_{i=1}^{N} \mathrm{rect}\left(\frac{t - iT_r}{T_p}\right) \mathrm{e}^{\left(j2\pi\left(\left(f_o - \frac{B}{2}\right)(t - iT_r) + \frac{K_{LF}}{2}(t - iT_r)^2\right)\right)} \tag{2}$$

where i denotes the pulse sequence; rect $\left(\frac{t-iT_r}{T_p}\right)$ represents the i-th rectangular pulse with a width of T_p; T_r is the pulse repetition period; T_p is the pulse width; f_o is the radar center frequency; B means the bandwidth; and $K_{LF} = \frac{B}{T_p}$ is the LFM coefficient.

2.2. Jamming Signal

In this paper, we select nine typical radar active jamming patterns as the research objects, including: radio noise jamming (RNJ) [26], amplitude modulation jamming (AMJ) [34], frequency modulation jamming (FMJ) [34], comb spectrum jamming (CJ) [13], phase modulation jamming (PMJ) [34], linear sweep frequency jamming (LSFJ) [34], non-linear sweep frequency jamming (NLSFJ) [35], hopping frequency jamming (HFJ) [35] and periodic Gaussian pulse jamming (PGJ) [36].

2.2.1. Radio Noise Jamming

RNJ is a narrowband Gaussian process generated by filtering and then amplifying the noise signal. Its mathematical model is expressed as:

$$J(t) = U_r(t)e^{j(\omega t + \varphi)} \tag{3}$$

where $U_r(t)$ obeys a Gaussian distribution, ω is the carrier frequency, and φ is the initial phase, following a uniform distribution on $[0, 2\pi]$.

2.2.2. Amplitude Modulation Jamming

The model of AMJ is represented as:

$$J(t) = [U_0 + K_{AM}U_n(t)]e^{j(\omega t + \varphi)} \tag{4}$$

where $U_n(t)$ is zero-mean Gaussian white noise, U_0 is the carrier voltage, and K_{AM} is the amplitude modulation index.

2.2.3. Frequency Modulation Jamming

FMJ is a type of barrage jamming, and its model is expressed as:

$$J(t) = U_0 \cdot e^{j\left(\omega t + 2\pi K_{FM} \int_0^t u(t')dt' + \varphi\right)}. \tag{5}$$

where K_{FM} is the frequency modulation index and $u(t)$ is a zero-mean stationary random process.

2.2.4. Comb Spectrum Jamming

CJ consists of multiple narrowband noise frequency modulation signals, and expresses as:

$$J(t) = \sum_{i=1}^{m} U_i e^{j\left[2\pi f_i t + 2\pi K_{\mathrm{FM}} \int_0^t u(t)dt + \varphi\right]}. \tag{6}$$

where U_i is the amplitude, f_i represents the frequency where the comb teeth appear, and m is the number of frequencies.

2.2.5. Phase Modulation Jamming

The model of PMJ is:

$$J(t) = U_0 \cdot e^{j(\omega t + K_{PMU} u(t) + \varphi)} \tag{7}$$

where K_{PM} is the phase modulation index.

2.2.6. Linear Sweep Frequency Jamming

LSFJ varies linearly with time in a frequency band, and expresses as:

$$J(t) = U_0 e^{j\left(2\pi f_0 t + 2\pi K_{FM} t^2 + \varphi\right)} \tag{8}$$

where f_0 is the initial frequency.

2.2.7. Non-Linear Sweep Frequency Jamming

NLSFJ is similar to LSFJ, except that the instantaneous frequency magnitude of the jamming signal varies continuously with the square of time, and expresses as:

$$J(t) = U_0 e^{j\left(2\pi f_0 t + 2\pi K_{FM} t^3 + \varphi\right)} \tag{9}$$

2.2.8. Hopping Frequency Jamming

HFJ is a wideband non-stationary signal in which the frequency changes over time. The model of HFJ can be written as:

$$J(t) = \sum_{n=1}^{N} A_n e^{j(2\pi f_n t + \varphi_n)} \cdot p(t - nT_H). \tag{10}$$

where $\{A_n\}$ represents the amplitude sequence, $\{f_n\}$ is the pseudo-random frequency sequence, φ_n is the random phase sequence, T_H is the hop duration, and $p(t)$ is the base pulse signal with a pulse width of T_H.

2.2.9. Periodic Gaussian Pulse Jamming

PGP is a widely used active suppression jamming. The PGP is expressed as:

$$J(t) = \begin{cases} \Phi(t), 0 < t < \tau \\ 0, \tau < t < T \end{cases} \tag{11}$$

where $\Phi(t)$ represents a Gaussian function with a mean of 0 and a variance of 1, τ denotes the pulse duration, and T is the pulse period of the jamming.

The time-domain waveforms of the nine patterns are shown in Figure 1. As can be seen from the figure, there is little difference in the time domain between the various patterns of suppressed jamming signals, which are more difficult to distinguish manually and are easily disturbed by noise.

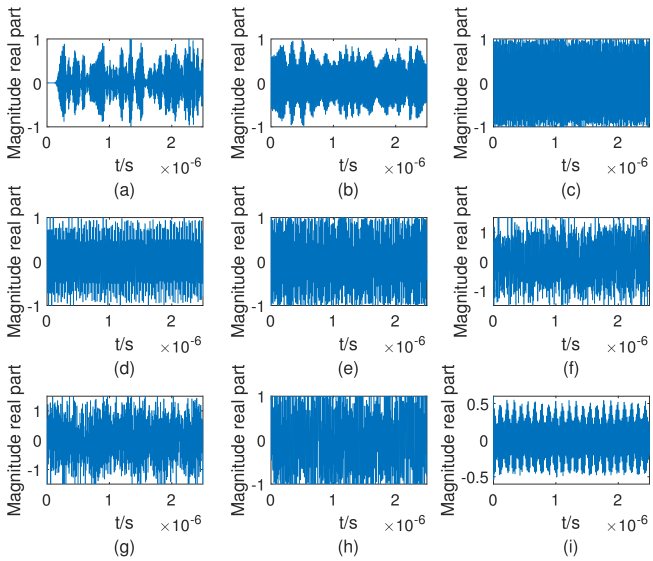

Figure 1. Time-domain waveforms of nine patterns of radar jamming signals. From (**a**) to (**i**): (**a**) RNJ, (**b**) AMJ, (**c**) FMJ, (**d**) CJ, (**e**) PMJ, (**f**) LSFJ, (**g**) NLSFJ, (**h**) HFJ, (**i**) PGJ.

3. Proposed Method

Assuming that the dataset of jamming signals received by the radar is $\mathcal{T}$, it consists of N_c+ N_u kinds of patterns of jamming. The subset $\mathcal{T}_c$ is composed of samples of N_c kinds of known patterns, while the subset $\mathcal{T}_u$ is composed of samples of N_u kinds of unknown patterns. The two subsets are complementary, meaning $\mathcal{T}_c \cup \mathcal{T}_u = \mathcal{T}$ and $\mathcal{T}_c \cap \mathcal{T}_u = \varnothing$. For the known patterns set $\mathcal{T}_c = \left\{ \left(\mathbf{x}_i^c, y_i^c, \mathbf{z}_{y_i^c} \right) \mid \mathbf{x}_i^c \in \mathcal{X}_c^{n_s \times N_d}, y_i^c \in \mathcal{Y}_c = \{1, 2, \cdots, N_c\}, \mathbf{z}_{y_i^c} \in \mathbf{Z}_c^{N_c \times N_k} \right\}$ containing n_c samples of N_c kinds of patterns, where $\mathbf{x}_i^c$ represents a N_d-dimensional vector of the i-th sample, y_i^c is the label of the sample, and $\mathbf{z}_{y_i}$ denotes the N_k-dimensional semantic vector describing the features of the sample's corresponding pattern to be obtained by supervised learning. Similarly, for the unknown patterns dataset consisting of n_u samples of N_c kinds of patterns, $\mathcal{T}_u = \left\{ \mathbf{x}_i^u, y_i^u, \mathbf{z}_{y_i^u}^u \right\}$, where $\mathbf{x}_i^u \in \mathcal{X}_u^{n_u \times N_d}$ is the N_d-dimensional feature vector of the i-th sample, $y_i^u \in \mathcal{Y}_u = \{N_c + 1, N_c + 2, \ldots, N_c + N_u\}$ represents the label of the sample, and $\mathbf{z}_{y_i^u} \in \mathbf{Z}_u^{N_u \times N_k}$ denotes the N_k-dimensional semantic information of the sample's corresponding pattern.

For the generalized zero-shot learning classification task considered in this paper, the supervised training phase only allows the utilization of dataset of known patterns $\mathcal{T}_c$. However, the objective is to ensure that the model trained in the supervised training phase can accurately classify $\mathcal{X}_c \cup \mathcal{X}_u$ into the $N_c + N_u$-dimensional space $\mathcal{Y}_c \cup \mathcal{Y}_u$ during the unsupervised classification phase.

Figure 2 illustrates the network framework of the RCAE-OWR algorithm, which consists of a supervised training phase and an unsupervised classification phase. In the supervised training phase, the residual convolutional autoencoder (RCAE) network is used to extract semantic features of known jamming patterns. Meanwhile, the network is trained using center loss, cross-entropy loss, and reconstruction loss. In [22,37], time-domain signals are directly used as inputs to the network to extract the deep features of different signals, and the simulation results prove that the direct time-domain signal-based recognition methods obtain good recognition performance in terms of accuracy and speed, demonstrating a huge potential for radar signal processing. Motivated by [22,37], in this paper, we also directly feed the IQ data of the jamming signals into the model. Then, after the supervised training, the unsupervised classification phase was entered. At this phase, the parameters of encoder of the RCAE are kept fixed, and both known and unknown jamming samples are input to the encoder to obtain their semantic features. Then, a distance-based discriminative method is employed to achieve open world recognition of radar jamming signals.

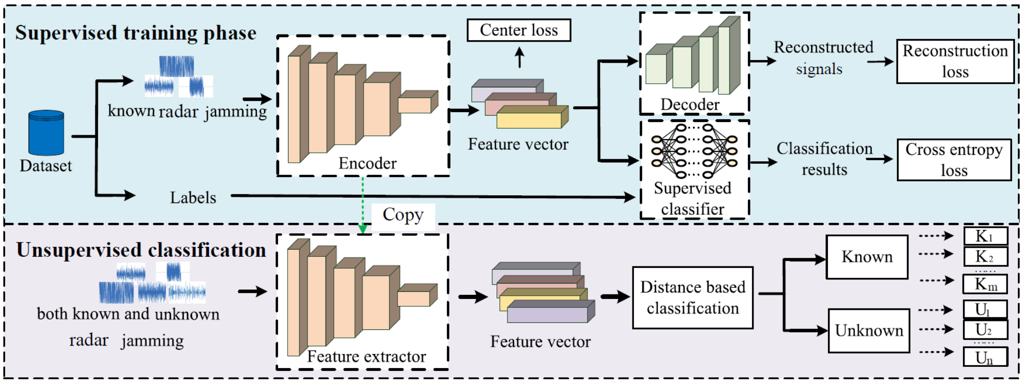

Figure 2. Radar jamming recognition framework of RCAE-OWR.

3.1. Supervised Training

The supervised training phase mainly consists of an autoencoder network (AE) and a supervised classification network classifier. The autoencoder is divided into two parts: the encoder and the decoder. In the supervised training phase, the focus is on constructing the mapping relationship between the time-domain signal and the semantic features.

3.1.1. Residual Convolutional Autoencoder

Due to the simplicity of the traditional AE structure, this study primarily considers the Residual Convolutional Autoencoder (RCAE). It replaces the fully connected layers in AE with convolutional layers and pooling layers, inheriting the characteristics of an autoencoder. This enables better feature learning and improves the efficiency of feature learning in AE. To prevent degradation in recognition performance, a residual network structure is employed. The input signals are IQ dual-channel data with a length of 512, resulting in a dimension of 2×512. Additionally, to maintain the vector dimensions after convolution, we have set the convolution kernel size to 3×3, padding = 1 and stride = 1. RCAE is based on the semantic autoencoder (SAE) architecture [31], and SAE enables mapping functions learned from known patterns to be better generalized to unknown patterns, which can effectively resist the domain shift problem [38]. In the encoding process, convolutional operations are used to extract features from input samples and obtain semantic vectors. Then, the decoder utilizes transpose convolution to reconstruct the semantic vectors and restore them to the original inputs.

The basic components of RCAE include the input layer, convolutional layers, semantic layer, deconvolutional layers, and output layer.

The RCAE designed in this paper is illustrated in Figure 3. It replaces the fully connected layers (FC) in the AE with convolutional layers and pooling layers, inheriting the characteristics of the AE for better feature learning. In addition, to prevent degradation of recognition performance, a residual structure is employed. The encoder uses convolutional operations to extract features from the input samples to obtain the semantic vector; the decoder utilizes transposed convolution to reconstruct the semantic vector and reduce it to the original signal.

The basic components of RCAE include: input layer, convolutional layer, semantic layer, deconvolutional layer and output layer.

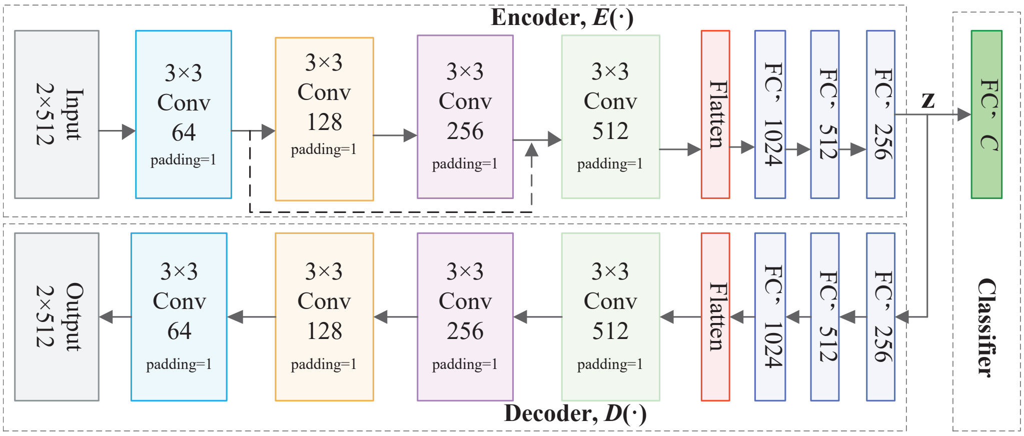

Figure 3. Residual Convolutional Autoencoder Network Structure.

In the encoding part, the input layer receives the input data $\mathbf{x}_i^c$ and passes it to the encoder. The encoder gradually extracts the semantic features of the input data through multiple convolutional layers and their residual structures, denoted as $\mathbf{z} = E(\mathbf{x}_i^c)$, where $E(\cdot)$ denotes the mapping function of the encoder.

The convolutional layers extract features from the input data $\mathbf{x}_i^c$ using convolutional operations. The data processing can be described as follows:

$$\mathbf{g}_i = f[\mathrm{conv}(\mathbf{x}, \mathbf{k}_i^{\mathrm{conv}}) + b_i^{\mathrm{conv}}]. \tag{12}$$

where $\mathbf{k}_i^{\mathrm{conv}}$ is the i-th convolutional kernel matrix, $\mathrm{conv}(\cdot)$ denotes the convolution operation, b_i^{conv} represents the i-th bias term, $f(\cdot)$ represents the activation function. The feature layer integrates the diverse features extracted by the convolutional layers and outputs the semantic vector $\mathbf{z}$.

In the decoding part, the semantic feature $\mathbf{z}$ is up-sampled through transpose convolutional operations, aiming to reconstruct the original signal $\tilde{\mathbf{x}}_i^c$ based on the semantic features. This process is denoted as $\tilde{\mathbf{x}}_i^c = D(\mathbf{z})$, where $D(\cdot)$ denotes the mapping function of the decoder. Finally, the reconstructed results are outputted through the output layer. The data processing can be described as follows:

$$\mathbf{q}_i = f\left[\mathrm{deconv}\left(\mathbf{z}, \mathbf{k}_i^{\mathrm{dec}}\right) + b_i^{\mathrm{dec}}\right] \tag{13}$$

where $\mathbf{k}_i^{\mathrm{dec}}$ represents the i-th convolutional kernel matrix, deconv $(\cdot)$ denotes the transpose convolution operation, b_i^{dec} represents the i-th bias term.

The training process of RCAE aims to minimize the reconstruction error, ensuring the effectiveness of feature extraction. It can be expressed as:

$$L_{re} = \frac{1}{2M} \sum_{i=1}^{M} \|\tilde{\mathbf{x}}_i^c - \mathbf{x}_i^c\|_2^2 \tag{14}$$

where M represents the number of samples in a batch, $\tilde{\mathbf{x}}_i^c$ is the signal reconstructed by $\mathbf{x}_i^c$ through the RCAE network.

To encourage the feature vectors of the same patterns to be close to their corresponding pattern centers and far from centers of different patterns, a center loss is introduced. During model training, the center loss assigns a center for each jamming pattern. Assuming the input sample is $\mathbf{x}_i^c$ with label y_i^c, and the center for pattern y_i is denoted as $\bar{\mathbf{z}}_{y_i^c}^c$. The center loss can be defined as:

$$L_{cl} = \frac{1}{M} \sum_{i=1}^{M} \left\| E(\mathbf{x}_i^c) - \bar{\mathbf{z}}_{y_i^c}^c \right\|_2^2 \tag{15}$$

During the model iteration process, the selection of pattern center $\overline{\mathbf{z}}^c_{y_i^c}$ is an important issue. Theoretically, the optimal center for pattern y_i^c would be the mean of all the feature vectors in that pattern. However, calculating the mean for all samples in each iteration would impose additional computational cost and reduce the efficiency of the model. To address this, the pattern center are initialized randomly, and then updated separately for each batch. The update process is as follows:

$$\Delta\overline{\mathbf{z}}^c_{y_i^c} = \frac{\sum_{m=1}^{M} \delta(y_i^c = y_m^c) \cdot \left(\overline{\mathbf{z}}^c_{y_i^c} - E(\mathbf{x}_m^c)\right)}{0.1 + \sum_{m=1}^{M} \delta(y_i^c = y_m^c)} \tag{16}$$

$$\overline{\mathbf{Z}}^{t+1}_{y_i^c} = \overline{\mathbf{Z}}^{t}_{y_i^c} - \alpha \cdot \Delta\overline{\mathbf{Z}}^{c}_{y_i^c} \tag{17}$$

where $\delta(y_i = k) = 1$ when $y_i = k$; otherwise, it is 0. $\overline{\mathbf{z}}^t_{y_i^c}$ is the semantic center of y_i^c at the t-th epoch, α is the learning rate.

3.1.2. Supervised Classifier

The encoder, acting as a feature extractor, takes the encoded features and feeds them into a fully connected layer followed by a softmax classifier, which outputs the predicted label. The loss function for this process utilizes cross-entropy between the predicted label and the true label.

$$L_{ce} = -\frac{1}{M}\sum_{i=1}^{M} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) \tag{18}$$

where $\mathbf{y}_i$ represents the true label $\mathbf{x}_i^c$ in one-hot format, and $\hat{\mathbf{y}}_i$ represents the predicted probability vector.

The reconstruction loss L_{re} ensures that the reduced-dimensional semantic features are representative, the central loss L_{cl} promotes cohesion among feature vectors of the same jamming pattern, and the cross-entropy loss L_{ce} enhances the discriminative ability of feature vectors among different jamming patterns. To achieve both increased inter-class distance and reduced intra-class distance, a joint loss function is proposed:

$$L = L_{ce} + \lambda_{cl} L_{cl} + \lambda_{re} L_{re} \tag{19}$$

where λ_{cl} and λ_{re} are constants used to balance the weights of the three loss functions. The detailed gradient $\nabla_\theta L$ of L is shown in Appendix A.

The network parameters during the supervised training phase are updated as shown in Algorithm 1.

Algorithm 1 Pseudocode for supervised training of RCAE-OWR

Input: known jamming patterns dataset $\left\{(\mathbf{x}_1^c, y_1^c), \cdots, (\mathbf{x}_n^c, y_n^c)\right\}$ and hyperparameter.
Output: Optimal parameters θ.

```
while epoch = 1, ..., N do
    for each batch with size M do
        Feed in a batch {(x_1^c, y_1^c), ..., (x_M^c, y_M^c)}.
        Calculate ΔZ̄^c_{y_i^c} via Equation (16).
        Calculate Z̄^t_{y_i^c} via Equation (17).
        Calculate L_ce via Equation (18).
        Calculate L_cl via Equation (15).
        Calculate L_re via Equation (14).
        Calculate L via Equation (19).
        Update θ : θ ← θ − η∇_θ L.
    end for
end while
```

3.2. Unsupervised Classification

After learning the mapping relationship between the original time-domain signals and the semantic feature space in the supervised training phase, this learned relationship can be generalized to the unknown jamming patterns dataset. Finally, the unknown patterns can be recognized and classified using the semantic features. Inspired by [39], a distance-based classification rule is proposed.

3.2.1. Known Jamming Patterns Classification

Sequentially, the n_c known jamming samples from the training set are inputted into the trained RCAE-OWR model to extract semantic features. The semantic center vector $\bar{\mathbf{z}}_k^c$ corresponding to the k-th known jamming pattern is defined as:

$$\bar{\mathbf{z}}_k^c = \frac{\sum_{i=1}^{n_c} \delta(y_i = k)\mathbf{z}_{y_i^c}}{\sum_{i=1}^{n_c} \delta(y_i = k)}, k = 1, 2, \ldots, N_c \tag{20}$$

where $\mathbf{z}_{y_i^c}$ denotes the semantic feature extracted from the i-th input sample $\mathbf{x}_i^c$. When a test sample $\mathbf{t}$ is input, its semantic features $E(\mathbf{t})$ are extracted by the encoder, and its Mahalanobis distance to the center vectors of each known jamming pattern is calculated as:

$$d(E(\mathbf{t}), \bar{\mathbf{z}}_k^c) = \sqrt{\left(E(\mathbf{t}) - \bar{\mathbf{z}}_k^c\right)^T \mathbf{\Omega}_k^{-1} \left(E(\mathbf{t}) - \bar{\mathbf{z}}_k^c\right)} \tag{21}$$

where $\mathbf{\Omega}_k$ is the diagonal covariance matrix corresponding to $\bar{\mathbf{z}}_k$, and $\mathbf{\Omega}_k^{-1}$ is its inverse matrix.

Let $d_1 = \min_k d\left(E(\mathbf{t}), \bar{\mathbf{z}}_k^c\right)$. If $d_1 \le \Theta_c$, then $\mathbf{t}$ belongs to the known jamming patterns. Here, $\Theta_c = \lambda_c \times 3\sqrt{N_\Omega}$ is a given threshold determined by the 3σ criterion [34], where λ_c is a coefficient and N_Ω is the dimension of $\mathbf{\Omega}_k$. In this case, the label $y_\mathbf{t}$ of $\mathbf{t}$ can be determined as follows:

$$y_\mathbf{t} = \arg\min_k (d(E(\mathbf{t}), \bar{\mathbf{z}}_k^c)) \tag{22}$$

when $d_1 > \Theta_c$, $\mathbf{t}$ belongs to the unknown jamming patterns, and its label $y_\mathbf{t}$ in this case is described in Section 3.2.2.

3.2.2. Unknown Jamming Patterns Classification

Let $\mathcal{T}_U$ represent the dataset of unknown jamming patterns. If $\mathcal{T}_U = \varnothing$, then $\mathbf{t}$ belongs to the first sample of a new unknown jamming pattern, and its label is $y_\mathbf{t} = \mathcal{U}_1$. The sample $\mathbf{t}$ is then recorded in the $\mathcal{T}_u$. If $\mathcal{T}_U \neq \varnothing$, the following rules are applied to determine whether $\mathbf{t}$ belongs to a previously recorded unknown jamming pattern or a new unknown jamming pattern. First, let $\bar{\mathbf{z}}_i^u$ represent the semantic center vector of pattern $\mathcal{U}_i$ in the $\mathcal{T}_U$:

$$\bar{\mathbf{z}}_i^u = \frac{\sum_{k \in \mathcal{T}_u} \delta(y_k = \mathcal{U}_i)\mathbf{z}_{y_k^u}}{\sum_{k \in \mathcal{T}_u} \delta(y_k = \mathcal{U}_i)}, i = 1, 2, \cdots, N \tag{23}$$

where N is the number of patterns that already exist in $\mathcal{T}_U$. $\mathbf{z}_{y_j^u}$ denotes the semantics extracted from the j-th unknown pattern sample $\mathbf{x}_j^u$. Let $d_2 = \min_{\mathbf{z}_i} d\left(E(\mathbf{t}), \bar{\mathbf{z}}_i^u\right)$. If $d_2 \le \Theta_u$, then $\mathbf{t}$ belongs to an existing unknown jamming pattern. Here, $\Theta_u = \lambda_u \max(d_c, d_u)$ is a given threshold, where λ_u is a coefficient, d_c and d_u are defined as follows:

$$d_c = \max_{\bar{\mathbf{z}}_k^c} d(E(\mathbf{t}), \bar{\mathbf{z}}_k^c) \tag{24}$$

$$d_u = \max_{\bar{\mathbf{z}}_i^u} d(E(\mathbf{t}), \bar{\mathbf{z}}_i^u) \tag{25}$$

In this case, the label $y_\mathbf{t}$ of $\mathbf{t}$ can be determined as follows:

$$y_\mathbf{t} = \arg\min_{u_i} (d(E(\mathbf{t}), \bar{\mathbf{z}}_i^u)) \tag{26}$$

Otherwise, $\mathbf{t}$ belongs to a new unknown jamming pattern, and its label is $y_{\mathbf{t}} = \mathcal{U}_{N+1}$. The test sample $\mathbf{t}$ is then recorded in the $\mathcal{T}_u$.

In summary, the recognition rules for the unsupervised classification phase are presented in Algorithm 2.

Algorithm 2 Pseudocode for unsupervised classification of RCAE-OWR

Input: test sample $\mathbf{t}_i$ and well-trained weight of Encoder, θ_c.
Output: Jamming pattern y_i.

for $i = 1, \ldots, N_{\text{test}}$ **do**
  Calculate semantic vector $\mathbf{z}_i$ of $\mathbf{t}_i$
  $d_1 = \min_k d\left(E(\mathbf{t}), \bar{\mathbf{z}}_k^c\right)$
  $d_2 = \min_i d\left(E(\mathbf{t}), \bar{\mathbf{z}}_i^u\right)$
  $d_c = \max\limits_{\bar{\mathbf{z}}_k^c} d\left(E(\mathbf{t}), \bar{\mathbf{z}}_k^c\right)$
  $d_u = \max\limits_{\bar{\mathbf{z}}_i^u} d\left(E(\mathbf{t}), \bar{\mathbf{z}}_i^u\right)$
  $\Theta_u = \lambda_u \max(d_c, d_u)$
  if $d_1 \leq \Theta_c$ **then**
    $y_i = \arg\min\limits_k \left(d\left(E(\mathbf{t}), \bar{\mathbf{z}}_k^c\right)\right)$
  else if $d_1 > \Theta_c$ and $\mathcal{T}_U = \varnothing$ **then**
    add $\mathbf{t}$ to $\mathcal{T}_u$
    $y_i = \mathcal{U}_1$
  else if $d_1 > \Theta_c, \mathcal{T}_U \neq \varnothing$ and $d_2 \leq \Theta_u$ **then**
    $y_i = \arg\min\limits_{\mathcal{U}_i} \left(d\left(E(\mathbf{t}), \bar{\mathbf{z}}_i^u\right)\right)$
  else
    $y_i = \mathcal{U}_{N+1}$
    add **t to** $\mathcal{T}_u$**.**
  end if
end for

4. Performance Evaluation

4.1. Simulation Parameter Settings

The simulations are performed on a PC with an Intel(R) Core(TM) i7-9700 CPU and a GeForce RTX2070s GPU. The deep learning framework PyTorch is used for training and testing the neural network. The RCAE-OWR network is initialized with random weights, and the learning rate is set to 0.001 . The batch size is set to 256, and the number of epochs is set to 250. The values of $\lambda_{cl}, \lambda_{re}, \lambda_c$, and λ_u are set to $0.02, 10, 0.5$, and 1.5, respectively. Furthermore, grid search is applied to ascertain the optimal hyperparameters. Specifically, we train the model for each possible combination among all the candidate parameters by loop traversal and pick the parameter combination that minimizes the validation set error as the optimal hyperparameters.

4.2. Original Dataset

The jamming signals are generated using MATLAB. The dataset includes the nine kinds of radar active jamming signals mentioned in Section 2.2, with detailed parameters shown in Table 1. The data description is shown in Table 2. Gaussian white noise is added during the simulation, and the JNR ranges from -4 dB to 18 dB with a step size of 2 dB. For each JNR, 1000 samples are generated for each type of radar active jamming signal. In the following experiments, N_c kinds of the jamming patterns will be treated as known jamming, while the remaining $9 - N_c$ kinds of patterns will be treated as unknown jamming.

Table 1. Nine kinds of Radar Active Jamming Parameter Settings.

Jamming Pattern	Parameter Description
RNJ	Carrier frequency: 60∼110 MHz.
AMJ	Carrier frequency: 60∼110 MHz, bandwidth: 5∼10 MHz, K_{AM}: 0.1∼0.9
FMJ	Carrier frequency: 60∼110 MHz, bandwidth: 5∼10 MHz, K_{FM}: 0.1∼0.9
CJ	Carrier frequency: 60∼110 MHz, bandwidth: 5∼10 MHz, number of bands: 2∼4
PMJ	Carrier frequency: 60∼110 MHz, bandwidth: 5∼10 MHz, K_{PM}: 0.1∼0.9
LSFJ	Starting frequency: 1∼10 MHz, ending frequency: 50∼100 MHz.
NLSFJ	Starting frequency: 1∼10 MHz, ending frequency: 50∼100 MHz.
HFJ	$\mathcal{N}$ = 20, $\{f_c\}$: [10, 100] MHz, T_H: 32∼64 µs
PGJ	Pulse period T: [2.5, 10] µs . Duty cycle: [1/8, 1/2]

Table 2. The original dataset.

Total Samples	Samples of Each Pattern	Samples Each JNR	Feature Dimension	Patterns
108,000	12,000	1000	2 × 512	9
patterns				
RNJ, AMJ, FMJ, CJ, PMJ, LSFJ, NLSFJ, HFJ, PGJ				
number of JNR values		JNR values		
12		−4, −2, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18		

4.3. Training Process

Firstly, N_c patterns are selected from the original dataset as known patterns, and the remaining $9 - N_c$ patterns are considered unknown patterns. The samples from known patterns are used not only for training the network but also for testing. However, the samples from unknown patterns are exclusively used for testing purposes. To reduce the impact of noise on the training results, during the training phase, only the known pattern samples with a JNR higher than 12dB are used to train the RCAE-OWR network. For each JNR, 700 samples are randomly selected from the known pattern samples to form the training set, and another 100 samples are randomly chosen to construct the validation set. In the testing phase, a wider range of JNR values is used, namely, JNR = −4–18 dB. For each of the nine jamming patterns, 200 samples are randomly selected at each JNR value to form the test set. It is important to note that the training set, validation set and test set for the known patterns samples are mutually exclusive and do not overlap with each other.

Figures 4–6 show the curves of cross-entropy loss L_{ce}, central loss L_{cl} and reconstruction loss L_{re} of the proposed RACE-OWR algorithm with the number of iterations. It can be seen from the figures that the three losses are gradually decreasing and converging after the epoch exceeds 50, which illustrates the effectiveness and the fast convergence and low complexity of the algorithm. In the convergence stage, the reconstruction loss approaches approximately 0.274, which indicates that the features extracted by the network designed in this paper can effectively represent the original signal and further provide a guarantee for unsupervised classification. The cross-entropy loss converges to 0.82, which contributes to the separation of different patterns, while the central loss converges to 0.12, which contributes to the aggregation of samples from the same pattern.

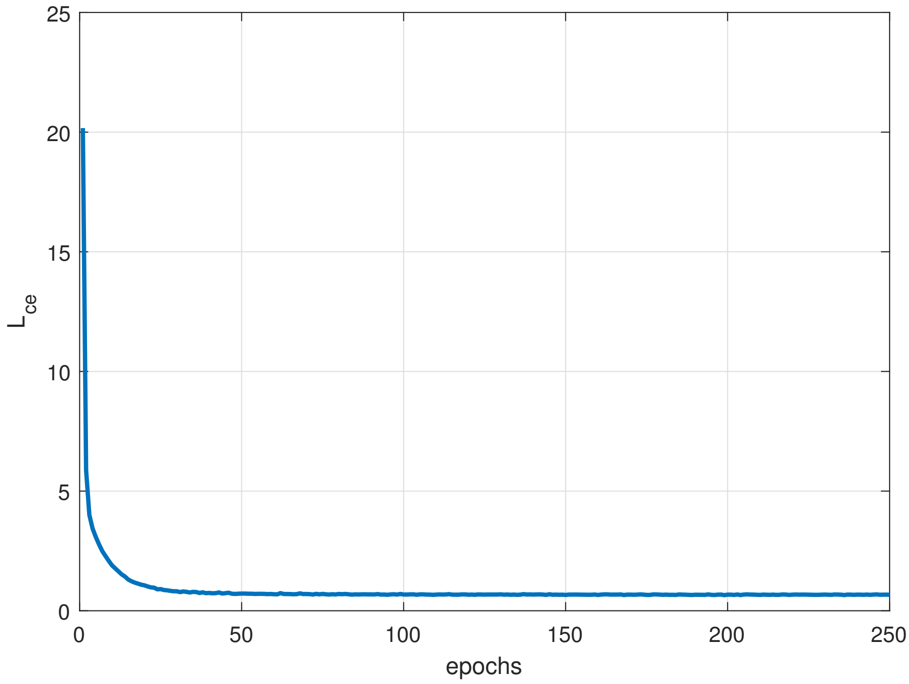

Figure 4. Cross-entropy loss.

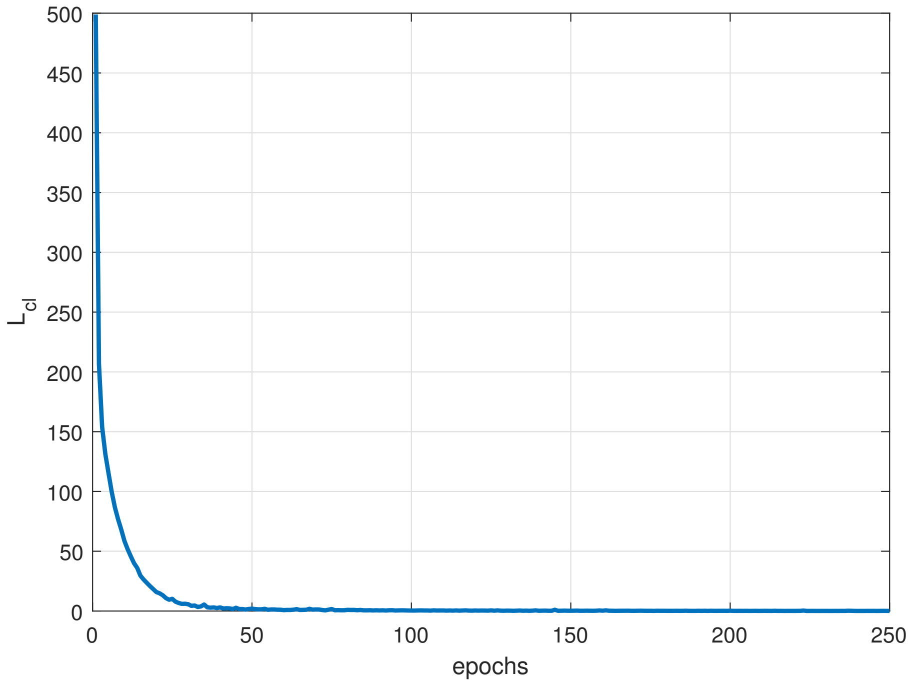

Figure 5. Center loss.

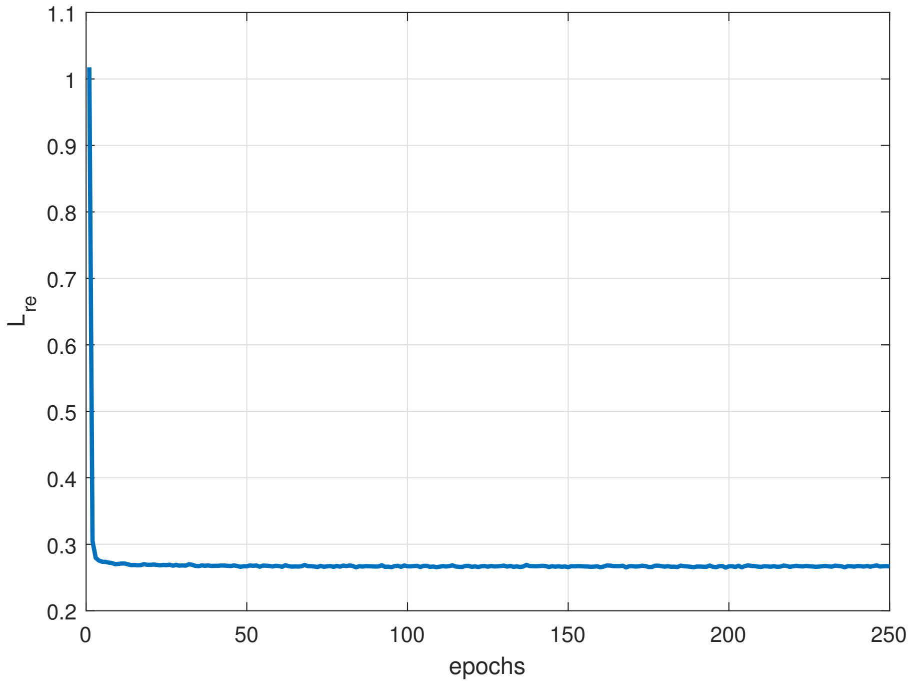

Figure 6. Reconstruction loss.

4.4. Impact of Different Combinations of Unknown Patterns on Classification Performance

In order to investigate the influence of different combinations of unknown patterns on the algorithm's classification performance, we conducted experiments by randomly selecting two jamming patterns from the dataset as unknown patterns. We define two metrics for evaluation: true known rate (TKR) and true unknown rate (TUR). TKR is calculated as TK/K, and TUR is calculated as TU/U, where TK and TU represent the number of correctly identified samples from known and unknown patterns, respectively, while K and U represent the total number of samples from known and unknown patterns. For ease of comparison, we ensure that the test set's JNR is consistent with the training set, ranging from 12 to 18 dB. The experimental results are summarized in Table 3. From the table, it can be observed that at higher JNR values, the algorithm effectively distinguishes between known and unknown patterns, validating the efficacy of our proposed approach. However, under the three combinations tested, neither TKR nor TUR could reach 1. This indicates that some samples from the known patterns are misclassified as unknown, and vice versa.

Table 3. Performance with different unknown pattern combinations.

Indicator	Unknown Patterns		
	RNJ, HFJ	**AMJ, PGJ**	**RNJ, NLSFJ**
Average accuracy	0.957	0.965	0.978
TKR	0.966	0.982	0.994
TUR	0.999	0.999	0.995

4.5. Impact of λ_c on Algorithm Performance

The value of Θ_c, which determines the proportion of jamming signal samples classified as known or unknown patterns, is influenced by the value of λ_c. In Section 3.2.1, Θ_c is defined as $\Theta_c = \lambda_c \times 3\sqrt{N_\Omega}$, where N_Ω is a constant representing the dimension of the matrix. Therefore, this section investigates the impact of λ_c on the algorithm's performance. Since TKR and TUR cannot both increase with the increase in λ_c, a weighted true rate (WTR) [39] is defined to balance these two metrics, i.e., $WTR = \xi \times TKR + (1 - \xi) \times TUR$, where ξ is a balancing factor set to 0.5 , indicating equal importance given to TKR and

TUR. Figure 7 shows the curves of TKR, TUR, and WTR as λ_c varies. It can be observed that as λ_c increases, TKR increases while TUR decreases. This is because as λ_c increases, the value of Θ_c increases, so more samples are discriminated as known jamming patterns and fewer samples are discriminated as unknown jamming patterns. The value of WTR initially increases and then decreases with the increase in λ_c, reaching its maximum at a value of approximately 0.5∼0.7.

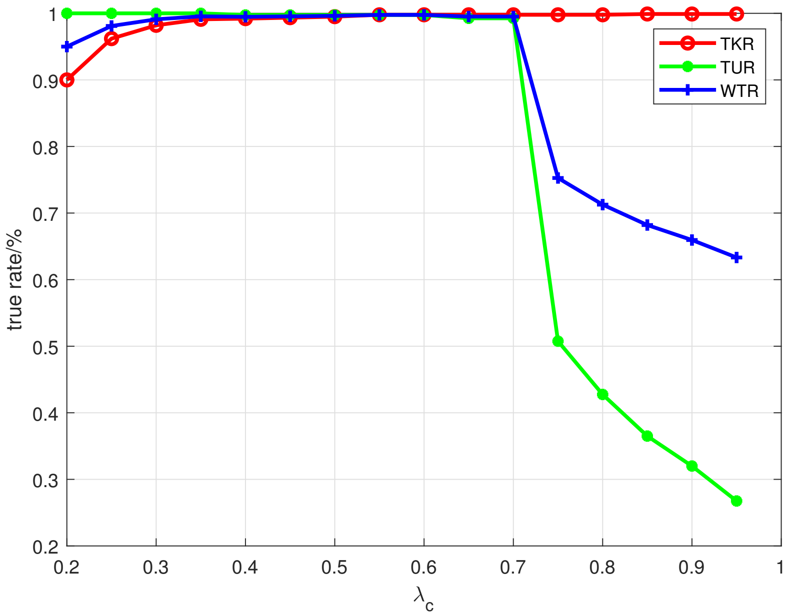

Figure 7. The impact of λ_c on recognition rate.

4.6. Ablation Study

In this research, we conduct an ablation study to comprehend the individual contributions of component loss functions in the context of the RCAE-OWR learning model. By systematically eliminating specific component loss functions, we investigate their impact on the overall performance. The experimental findings, as summarized in Table 4, reveal crucial insights into the significance of each loss function. Notably, the removal of L_{ce} (Cross-Entropy Loss) leads to the most substantial performance degradation, resulting in an accuracy of 95.1%. These results underscore the paramount role played by the cross-entropy loss in the learning process, surpassing the influence of the other two component loss functions, namely, L_{re} and L_{ct}. Furthermore, we observe that the incorporation of Lre enhances the accuracy by 3.3%. This improvement can be attributed to the ability of L_{re} to preserve essential semantic features, thereby facilitating the differentiation between known and unknown patterns. Additionally, the utilization of the center loss function, L_{ct} yields a minor enhancement in our learning model's performance.

Table 4. Ablation study on RCAE-OWR. L_{ce}: cross entropy loss. L_{cl}: center loss. L_{re}: reconstruction loss.

Variant	True Rate
RCAE-OWR(L_{ce},L_{cl} and L_{re})	0.995
without L_{ce}	0.951
without L_{cl}	0.982
without L_{re}	0.962

4.7. Open set Recognition Performance

When N_c is set to 7 as well as PGJ and NLSFJ as unknown patterns, the accuracy of the RCAE-OWR algorithm for the seven known jamming patterns and the detection rate for the two unknown jamming patterns are shown in Figure 8. It can be observed that as the JNR increases, the recognition rates for various patterns of jamming also increase.This is because with a higher JNR, the signal becomes cleaner, and the extracted features become more discriminative. When the JNR is below -2 dB, the various jamming signals are submerged in the noise, and the extracted semantic features are not sufficient to separate them. As a result, samples of known jamming patterns are often recognized as samples of unknown jamming patterns, leading to a higher detection rate for unknown jamming patterns compared to known jamming patterns.

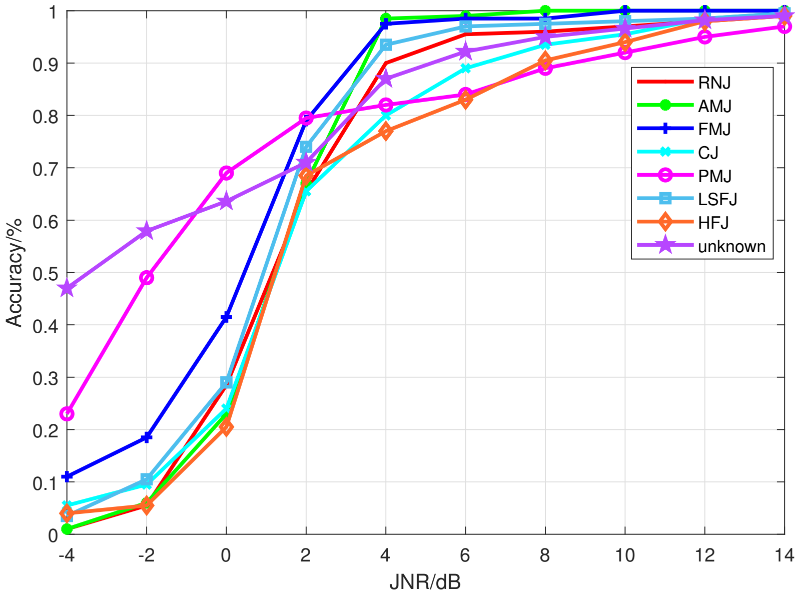

Figure 8. Open set recognition of RCAE-OWR.

Table 5 presents the accuracy of the RCAE-OWR algorithm and the supervised-based algorithm based on seven known jamming patterns when the JNR is set to 10dB. For a fair comparison, the network structure of the supervised-based algorithm is consistent with the encoder of the RCAE-OWR network in this paper. The difference lies in the fact that the supervised-based algorithm directly outputs the probabilities of the seven jamming patterns as the recognition results, while RCAE-OWR does not have prior knowledge of the specific number of patterns. Therefore, Algorithm 2 is used for jamming pattern recognition in RCAE-OWR. As the supervised-based algorithm is designed for closed-set recognition, it achieves higher detection rates. On the other hand, the RCAE-OWR algorithm is designed for open-set recognition, which means that known jamming patterns can be wrongly recognized as unknown jamming patterns, and unknown jamming patterns can be misclassified as known jamming patterns. As a result, the recognition rate of RCAE-OWR is lower than that of the supervised-based algorithm.

Table 5. Comparison of detection performance between two algorithms.

Method	RNJ	AMJ	FMJ	CJ	PMJ	LSFJ	HFJ
supervised	98%	100%	99%	95%	92%	98%	95%
RCAE-OWR	96%	100%	98.5%	93.5%	89%	97.5%	90.5%

The confusion matrix of RCAE-OWR for known jamming patterns and unknown jamming patterns is shown in Figure 9 when JNR is set to 18 dB. It can be observed from the figure that almost all samples are correctly classified into their respective patterns. This is because with higher levels of jamming, the extracted signal features better represent the original signal, resulting in increased intra-class cohesion and expanded inter-class separation. However, it is also evident that there are cases where known jamming patterns are recognized as unknown jamming patterns, and vice versa. Therefore, our algorithm needs to ensure a high detection rate for unknown jamming patterns while minimizing the impact on the recognition of known jamming patterns.

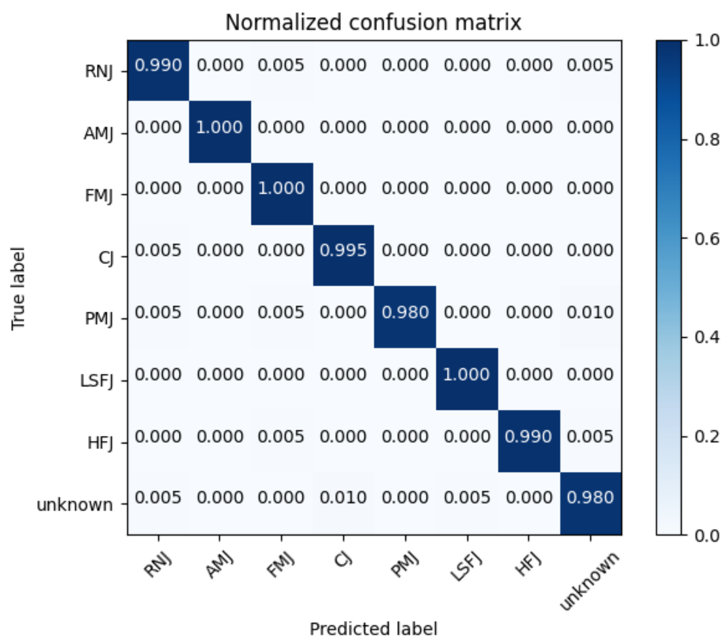

Figure 9. RCAE-OWR open set recognition results when JNR = 18 dB.

4.8. Analysis of Unknown Jamming Patterns Recognition Performance in RCAE-OWR

To evaluate the recognition performance of RCAE-OWR for unknown jamming patterns, *Precision*, *Recall*, and $F1$ are introduced as performance metrics [6]. *Precision* represents the ratio of the number of correctly classified samples to the sum of correctly classified samples and incorrectly classified samples. *Recall* represents the ratio of the number of correctly classified samples to the total number of samples. $F1$ is defined as the harmonic mean of *Precision* and *Recall*, given by $F1 = 2 \times Precision \times Recall/(Precision + Recall)$ [40].

4.8.1. Performance Comparison of Different Methods

In order to validate the effectiveness of our proposed RCAE-OWR, two other OWR-based methods, IOmSVM [41] and SNN-OWR [35] are implemented for comparison. The features extracted in the IOmSVM method are referenced from [12]. We utilize Openness [42] to measure the complexity of the open world task to describe the weight of the number of unknown patterns. Openness is defined as :

$$\text{Openness } = 1 - \sqrt{\frac{N_{\text{train}}}{N_{\text{test}}}} \tag{27}$$

where N_{train} denotes the number of patterns contained in the training set and N_{test} denotes the number of patterns contained in the test set. When the Openness is 0 , the task will move to closed-set recognition. In this paper, we set N_{test} to 9 and keep changing the value of N_{train} for experiments; $F1$ score is used to measure the performance of the algorithm and the results are shown in Figure 10. From the figure, it can be seen that as the Openness increases, the classification accuracy of the three methods decreases, which is attributed to the fact that more information about unknown patterns is fed into the model, posing a serious challenge to the classification of the model. Meanwhile, RCAE-OWR has the highest recognition accuracy at the same Openness, e.g., when the Openness is 0.25, the average $F1$ score of RCAE-OWR is 4.29% and 10.26% higher than that of SNN-OWR and IOmSVM, respectively.

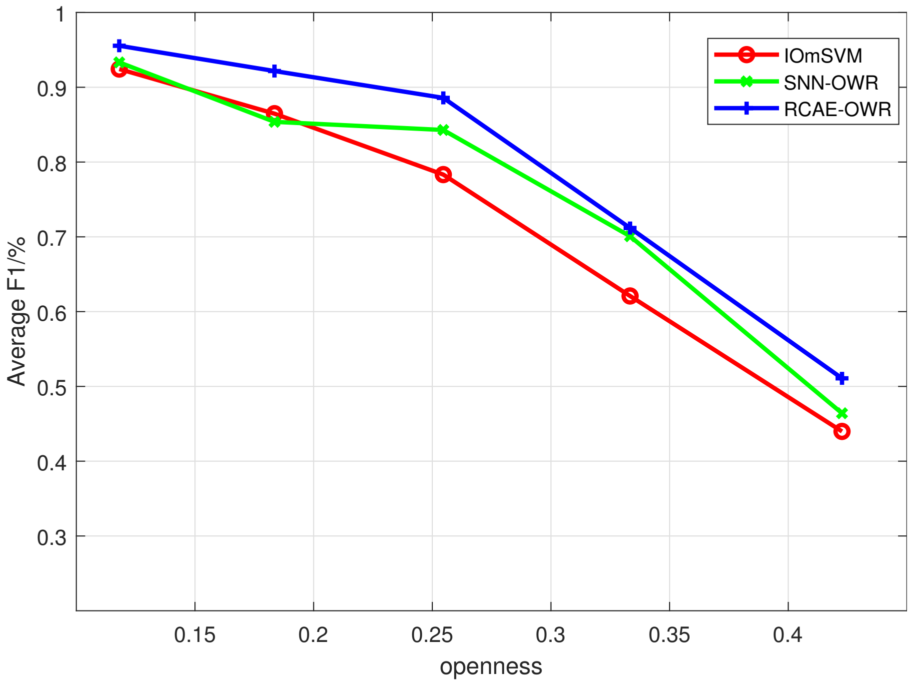

Figure 10. $F1$ scores against varying Openness under different methods.

Figure 11 illustrates the performance of the three methods as the JNR varies, with N_u set to 3. Notably, the detection performance of all three methods exhibits improvement as JNR increases, and the differences in $F1$ scores diminish with higher JNR values. Among the three OWR-based methods, RCAE-OWR stands out with the highest classification accuracy, particularly at low JNR. For instance, when the JNR is 8 dB, the average $F1$ score of RCAE-OWR surpasses that of SNN-OWR and IOmSVM by 6.85% and 12.23%, respectively.

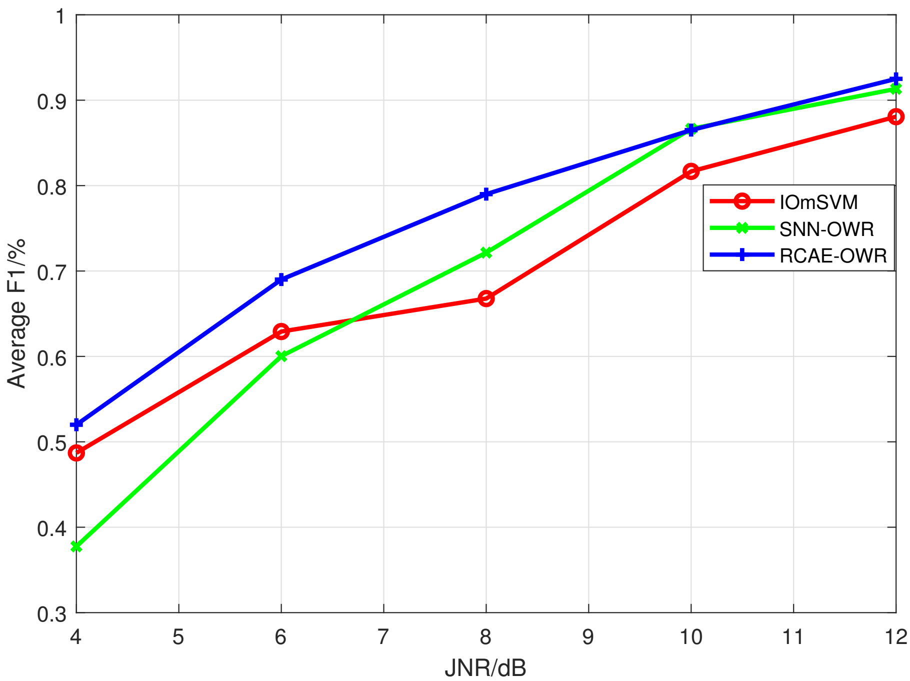

Figure 11. $F1$ scores against JNR under different methods.

Table 6 presents the offline training time and online recognition time of the three methods. It can be observed from the table that the IOmSVM method has the shortest training time, and likewise the recognition accuracy is the worst within the three methods.

Both SNN-OWR and RCAE-OWR require longer training times; however, once the models are trained effectively, they still achieve real-time jamming recognition.

Table 6. Runtime (in seconds) of different methods.

Method	Offline Training	Online Recognition
IOmSVM	753	0.0075
SNN-OWR	1506	0.0071
RCAE-OWR	1427	0.0062

4.8.2. Impact of Unknown Patterns on RCAE-OWR Performance

To investigate the influence of the number of unknown jamming patterns on the algorithm's recognition performance, we randomly selected two, three, four, five and six kinds of patterns out of the nine kinds of jamming patterns as unknown jamming patterns, while the remaining patterns were considered known jamming patterns. The experimental results are presented in Table 7, with JNR set to 12 dB. From the table, it can be observed that as the number of unknown jamming patterns decreases, the algorithm's detection performance improves, as indicated by an increase in the $F1$ score. For example, when the number of unknown jamming patterns is reduced from four to two, the average $F1$ score increases by 0.070. However, when the number of unknown patterns surpasses the number of known patterns, it poses a challenge to the algorithm's detection performance. For example, when increasing the number of unknown patterns from four to five, the average $F1$ score decreases by 0.174.

Table 7. Effect of Unknown jamming patterns on RCAE-OWR Performance.

Number of Patterns	Pattern	*Precision*	*Recall*	*F*1
2	Pattern 1	0.941	0.964	0.952
	Pattern 2	0.979	0.940	0.959
3	Pattern 1	0.916	0.940	0.927
	Pattern 2	0.929	0.928	0.928
	Pattern 3	0.922	0.900	0.910
4	Pattern 1	0.861	0.930	0.894
	Pattern 2	0.907	0.920	0.913
	Pattern 3	0.912	0.820	0.863
	Pattern 4	0.909	0.841	0.873
5	Pattern 1	0.841	0.724	0.778
	Pattern 2	0.781	0.696	0736
	Pattern 3	0.701	0.680	0.690
	Pattern 4	0.623	0.713	0.663
	Pattern 5	0.542	0.692	0.692
6	Pattern 1	0.779	0.598	0.677
	Pattern 2	0.712	0.545	0.613
	Pattern 3	0.603	0.536	0.567
	Pattern 4	0.481	0.533	0.505
	Pattern 5	0.365	0.452	0.402
	Pattern 6	0.243	0.405	0.301

4.8.3. Recognition Performance of RCAE-OWR for Unknown Jamming Patterns

The recognition rates of RCAE-OWR for unknown jamming patterns with JNR of 4∼14 dB are shown in Figure 12. It can be observed from the graph that when the number of unknown jamming patterns is fixed, the average $F1$ score increases with the increase in JNR. This is because at lower JNR levels, a large number of known jamming patterns are incorrectly classified as unknown, resulting in poor classification performance. As the JNR increases, the characteristics of different jamming signals become more distinct, and the samples of known patterns wrongly classified as unknown patterns decrease, leading to improved performance in classifying unknown patterns. Additionally, at lower JNR levels, there is a significant difference in the average $F1$ score among the three different jamming patterns. However, as the JNR increases, this difference decreases. For example, at JNR = 4 dB, the average $F1$ score for two unknown patterns is 0.3 and 0.65 higher than that for three and four unknown patterns, respectively. At JNR = 10 dB, the average $F1$ score for two unknown patterns is 0.08 and 0.15 higher than that for three and four unknown patterns, respectively. This also highlights the importance of improving the JNR of the dataset as a prerequisite and guarantee for open world recognition of radar jamming signals.

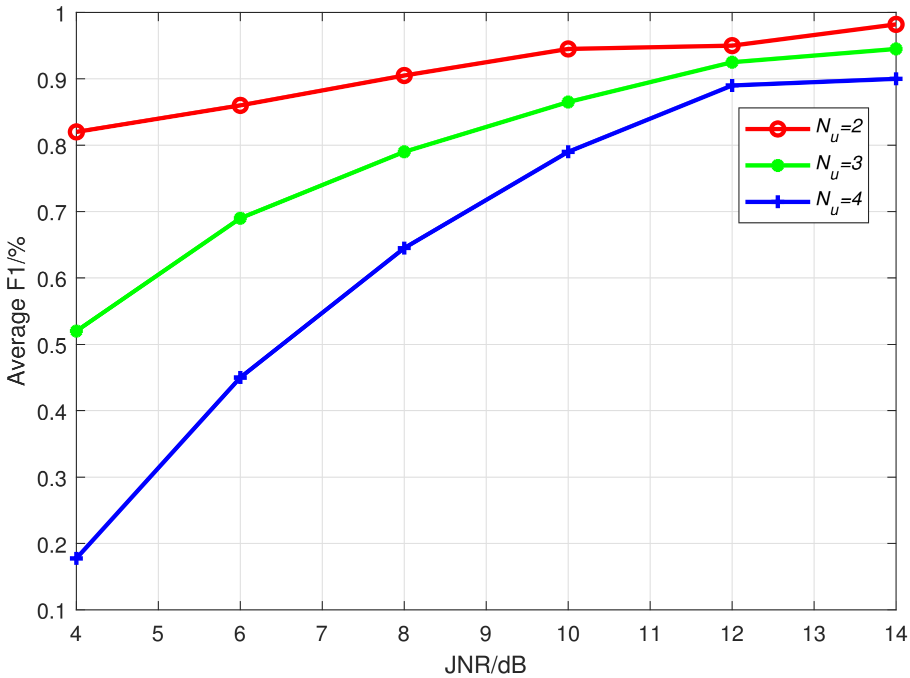

Figure 12. Recognition performance of RCAE-OWR for different unknown jamming patterns.

4.8.4. Performance of RCAE-OWR for Unknown Patterns under Different Combinations

To investigate the influence of JNR on the algorithm's performance with varying combinations of unknown patterns, we randomly extract three or four kinds of signals from the dataset to form the unknown patterns set, while the remaining are considered as known patterns for experimental evaluation. The results of these experiments are presented in Figure 13. From the figure, it can be seen that as the JNR increases, the overall classification accuracy also improves; this is because as the JNR increases, the signal is less affected by noise and is more discriminative. Notably, when the number of unknown patterns is fixed, the classification accuracy consistently increases with higher JNR values. Additionally, regardless of the specific combination of unknown patterns data sets, the classification accuracy exhibits minimal variation under the same JNR condition, highlighting the robustness of the RCAE-OWR algorithm.

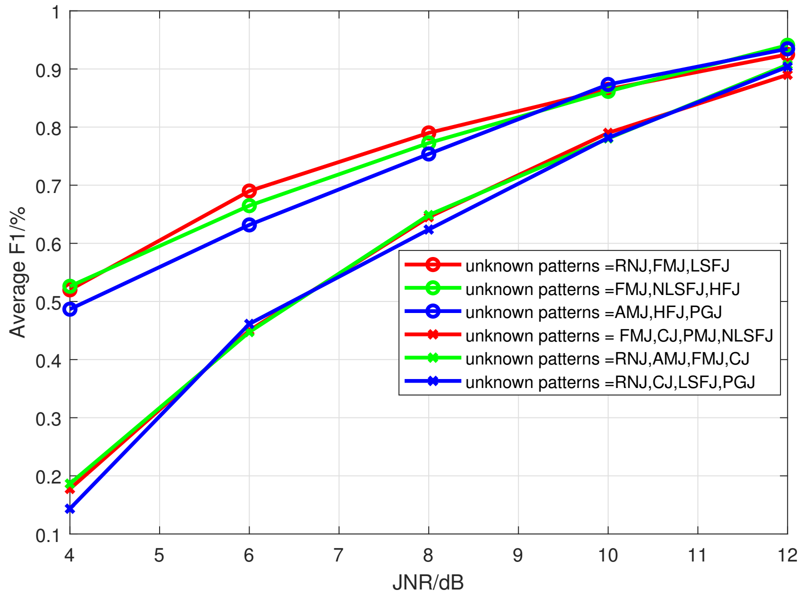

Figure 13. Performance under different unknown patterns.

5. Conclusions

In this study, we addressed the problem of open world recognition for radar jamming signals. Leveraging the idea of zero-shot learning, we proposed an open world recognition method called RCAE-OWR based on residual autoencoders. This method utilizes the features extracted by the encoder network to form the semantic centers of known jamming patterns. Then, a distance-based approach is proposed to classify known and unknown jamming patterns. Experimental results demonstrate that the RCAE-OWR algorithm can effectively recognize unknown jamming patterns, especially under high JNR. Although the proposed RCAE-OWR algorithm has demonstrated excellent classification performance in suppression jamming signals, there are still some limitations and drawbacks to address. For instance, this paper focused only on common jamming signals, while there are other jamming patterns, such as agile noise jamming and deceptive jamming. In the future, these patterns should be incorporated into our method. Additionally, deep learning has been widely applied to intelligent radar jamming signal recognition with a balanced training set. However, in most cases, an imbalanced training set is inevitable. Therefore, considering open world recognition of radar jamming under class-imbalanced conditions will be a challenging yet valuable task in future research.

Author Contributions: Conceptualization Z.Z. and Y.Z.; methodology, Y.Z. and Z.Z.; software, Y.B. and Y.Z.; validation, Z.Z.; formal analysis, Z.Z.; investigation, Z.Z. and Y.Z.; resources, Z.Z.; data curation, Y.B.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.B. and Y.Z.; visualization, Z.Z.; supervision, Z.Z.; project administration, Y.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the State Key Program of National Natural Science of China (Grant No. U19B2016) and Zhejiang Provincial Key Lab of Data Storage and Transmission Technology, Hangzhou Dianzi University.

Acknowledgments: The authors would like to thank an Editor who handled the manuscript and the anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1

Lemma A1. *Assuming the $\nabla_\theta L_{ce}$, $\nabla_\theta L_{cl}$ and $\nabla_\theta L_{le}$ are gradients of L_{ce}, L_{cl} and $L_{e\varepsilon}$, respectively, then there is:*

$$\begin{aligned}\nabla_\theta L &= \nabla_\theta L_{ce} + \lambda_{cl}\nabla_\theta L_{cl} + \lambda_{re}\nabla_\theta L_{re} \\ &= \frac{1}{M}\left(\sum_i (S(E(\mathbf{x}_i^c)) - y_i^c) + \sum_i E(\mathbf{x}_i^c) - \bar{\mathbf{z}}_{y_i^c}^c + \sum_i \tilde{\mathbf{x}}_i^c - \mathbf{x}_i^c\right)\end{aligned} \tag{A1}$$

Proof. because $L = L_{ce} + \lambda_{cl}L_{cl} + \lambda_{re}L_{re}, \therefore\ \nabla_e L$ can be notated as:

$$\nabla_\theta L_t = \nabla_\theta L_{ce} + \lambda_{rl}\nabla_\theta L_{kl} + \lambda_{ct}\nabla_\theta L_{ct} \tag{A2}$$

□

Notice that $L_{ce} = -\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{ce}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$, where $\mathcal{L}_{ce}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \sum_{j=1}^{N_c} y_{i,j}^c \log(\hat{y}_{i,j}^c)$, Let $\mathbf{u} = [u_1, u_2, \ldots, u_{N_c}]$ denote the output vector of the last layer of the network. After softmax operation, $\mathbf{u}$ yields $\hat{\mathbf{y}}_i$. That is, $\hat{\mathbf{y}}_i^j = S\big(E\big(\mathbf{x}_i^c\big)\big)_j = \frac{e^{u_j}}{\sum_k e^{u_k}}$. where $S(\cdot)$ represents the softmax operation, and $S(E(\mathbf{x}_i))_j$ and u_j are the j-th elements of $S\big(E\big(\mathbf{x}_i^c\big)\big)$ and $\mathbf{u}$, respectively. According to the chain derivative rule:

$$\frac{\partial \mathcal{L}_{ce}}{\partial \mathbf{u}} = \frac{\partial \mathcal{L}_{ce}}{\partial p}\frac{\partial p}{\partial \mathbf{u}} \tag{A3}$$

We have:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ce}}{\partial u_j} &= \sum_m \frac{\partial \mathcal{L}_{cem}}{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j} \\ &= \sum_m \frac{\partial - y_{i,m}^c \log S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial E\big(\mathbf{x}_i^c\big)_m}\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j} \\ &= \sum_m -\frac{y_{i,m}^c}{S\big(E\big(\mathbf{x}_i^c\big)\big)_m}\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j}\end{aligned}$$

Simplify to get:

$$\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j} = \begin{cases} S\big(E\big(\mathbf{x}_i^c\big)\big)_j\Big(1 - S\big(E\big(\mathbf{x}_i^c\big)\big)_j\Big), m = j \\ -S\big(E\big(\mathbf{x}_i^c\big)\big)_j S\big(E\big(\mathbf{x}_i^c\big)\big)_m, m \neq j \end{cases} \tag{A4}$$

Hence, $\frac{\partial \mathcal{L}_{ce}}{\partial \mathbf{u}}$ can be rewritten as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ce}}{\partial u_j} &= \sum_m \frac{\partial \mathcal{L}_{cem}}{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j} \\ &= \sum_m \frac{\partial - y_{i,m}^c \log S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial E\big(\mathbf{x}_i^c\big)_m}\frac{\partial S\big(E\big(\mathbf{x}_i^c\big)\big)_m}{\partial u_j} \\ &= S(E(\mathbf{x}_i^c))_j\Big(1 - S(E(\mathbf{x}_i^c))_j\Big) + \\ &\quad \sum_{m\neq j} -\frac{y_{i,m}^c}{S\big(E\big(\mathbf{x}_i^c\big)\big)_m}\Big(-S(E(\mathbf{x}_i^c))_j S(E(\mathbf{x}_i^c))_m\Big) \\ &= S(E(\mathbf{x}_i^c))_j - y_{i,j}^c\end{aligned}$$

therefore, $\frac{\partial \mathcal{L}_{ce}}{\partial \mathbf{u}} = S\left(E\left(\mathbf{x}_i^c\right)\right) - y_i^c$, according to $L_{ce} = -\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{ce}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$, we have:

$$\frac{\partial L_{ce}}{\partial \mathbf{u}} = \frac{1}{M}\sum_i S(E(\mathbf{x}_i^c)) - y_i^c \tag{A5}$$

Similarly, $\frac{\partial L_{cl}}{\partial \mathbf{u}} = \frac{\partial L_{cl}}{\partial E\left(\mathbf{x}_i^c\right)}$, and $L_{cl} = \frac{1}{M}\sum_{i=1}^{M}\left\|E\left(\mathbf{x}_i^c\right) - \overline{\mathbf{z}}_{y_i^c}^c\right\|_2^2$, we have:

$$\frac{\partial L_{cl}}{\partial \mathbf{u}} = \frac{\partial L_{cl}}{\partial E(\mathbf{x}_i)} = \frac{1}{M}\sum_i E\left(\mathbf{x}_i^c\right) - \overline{\mathbf{z}}_{y_i^c}^c \tag{A6}$$

Similarly,

$$\frac{\partial L_{re}}{\partial \mathbf{u}} == \frac{\partial L_{re}}{\partial E\left(\mathbf{x}_i^c\right)} = \frac{1}{M}\sum_i \tilde{\mathbf{x}}_i^c - \mathbf{x}_i^c \tag{A7}$$

In summary, it can be concluded that:

$$\begin{aligned}\nabla_\theta L &= \nabla_\theta L_{ce} + \lambda_{cl}\nabla_\theta L_{cl} + \lambda_{re}\nabla_\theta L_{re} \\ &= \frac{1}{M}\left(\sum_i (S(E(\mathbf{x}_i^c)) - y_i^c) + \sum_i E\left(\mathbf{x}_i^c\right) - \overline{\mathbf{z}}_{y_i^c}^c + \sum_i \tilde{\mathbf{x}}_i^c - \mathbf{x}_i^c\right)\end{aligned} \tag{A8}$$

References

1. Du, C.; Cong, Y.; Zhang, L.; Guo, D.; Wei, S. A Practical Deceptive Jamming Method Based on Vulnerable Location Awareness Adversarial Attack for Radar HRRP Target Recognition. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2410–2424. [CrossRef]
2. Xu, H.; Quan, Y.; Zhou, X.; Chen, H.; Cui, T.J. A Novel Approach for Radar Passive Jamming Based on Multiphase Coding Rapid Modulation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–14. [CrossRef]
3. Liu, Z.; Zhao, S. Unsupervised Clustering Method to Discriminate Dense Deception Jamming for Surveillance Radar. *IEEE Sensors Lett.* **2021**, *5*, 1–4. [CrossRef]
4. Lv, Q.; Quan, Y.; Feng, W.; Sha, M.; Dong, S.; Xing, M. Radar Deception Jamming Recognition Based on Weighted Ensemble CNN With Transfer Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *17*, 1–11. [CrossRef]
5. Hua, Q.; Zhang, Y.; Wei, C.; Ji, Z.; Jiang, Y.; Wang, Y.; Xu, D. A Self-Supervised Method Based on CV-MUNet++ for Active Jamming Suppression in SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [CrossRef]
6. Luo, Z.; Cao, Y.; Yeo, T.S.; Wang, Y.; Wang, F. Few-Shot Radar Jamming Recognition Network via Time-Frequency Self-Attention and Global Knowledge Distillation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–12. [CrossRef]
7. Huang, S.; Feng, Z.; Zhang, Y.; Zhang, K.; Li, W. Feature based modulation classification using multiple cumulants and antenna array. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–5.
8. Chen, T.; Gao, S.; Zheng, S.; Yu, S.; Xuan, Q.; Lou, C.; Yang, X. EMD and VMD Empowered Deep Learning for Radio Modulation Recognition. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *9*, 43–57. [CrossRef]
9. Zhu, F.; Jiang, Q.Q.; Lin, C.; Xiao, Y. Typical wide band EMI identification based on support vector machine. *Syst. Eng. Electron.* **2021**, *43*, 2400–2406.
10. Lu, Y.; Li, S. CFAR detection of DRFM deception jamming based on singular spectrum analysis. In Proceedings of the 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xiamen, China , 22–25 October 2018; pp. 1–6.
11. Yang, S.; Tian, B.; Zhou, R. A jamming identification method against radar deception based on bispectrum analysis and fractal dimension. *J. Xian Jiaotong Univ.* **2016**, *50*, 128–135.
12. Xu, C.; Yu, L.; Wei, Y.; Tong, P. Research on Active Jamming Recognition in Complex Electromagnetic Environment. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019; pp. 1–5.
13. Wei, Y.; Li, Y.; Zhang, J.; Tong, P. Radar Jamming Recognition Method Based on Fuzzy Clustering Decision Tree. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019; pp. 1–5.
14. Yu, Y.; Li, J.; Li, J; Xia, Y.; Ding, Z.; Samali, B. Automated damage diagnosis of concrete jack arch beam using optimized deep stacked autoencoders and multi-sensor fusion. *Dev. Built Environ.* **2023**, *14*, 100128. [CrossRef]
15. Zhang, Y.; Zhao, Z. Limited Data Spectrum Sensing Based on Semi-Supervised Deep Neural Network. *IEEE Access* **2021**, *9*, 1813–1817. [CrossRef]
16. Zhang, Y.P.; Zhao, Z.J. Semi-supervised deep learning using pseudo labels for spectrum sensing. *J. Nonlinear Convex Anal.* **2022**, *23*, 1913–1929.

17. Yu, Y.; Hoshyar, A.N.; Samali, B.; Zhang, G.; Rashidi, M.; Mohammadi, M. Corrosion and coating defect assessment of coal handling and preparation plants (CHPP) using an ensemble of deep convolutional neural networks and decision-level data fusion. *Neural Comput. Appl.* **2023**, *35*, 18697–18718. [CrossRef]
18. Lv, Q.; Quan, Y.; Sha, M.; Feng, W.; Xing, M. Deep Neural Network-Based Interrupted Sampling Deceptive Jamming Countermeasure Method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 9073–9085. [CrossRef]
19. Zhao, Q.; Liu, Y.; Cai, L.; Lu, Y. Research on electronic jamming identification based on CNN. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019.
20. Li, M.; Ren, Q.; Wu, J. Interference classification and identification of TDCS based on improved convolutional neural network. *J. Phys. Conf. Ser.* **2020**, *16*, 121–155. [CrossRef]
21. Liu, Q.; Zhang, W. Deep learning and recognition of radar jamming based on CNN. In Proceedings of the 2019 12th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 14–15 December 2019; pp. 208–212.
22. Shao, G.; Chen, Y.; Wei, Y. Convolutional neural network-based radar jamming signal classification with sufficient and limited samples. *IEEE Access* **2020**, *8*, 80588–80598. [CrossRef]
23. Shao, G.; Chen, Y.; Wei, Y. Deep fusion for radar jamming signal classification based on CNN. *IEEE Access* **2020**, *8*, 117236–117244. [CrossRef]
24. Tian, X.; Chen, B.; Zhang, Z. Multiresolution Jamming Recognition with Few-shot Learning. In Proceedings of the 2021 CIE International Conference on Radar (Radar), Haikou, Hainan, China, 15–19 December 2021; pp. 2267–2271.
25. Luo, H.; Liu, J.; Wu, S.; Nie, Z. A Semi-Supervised Deception Jamming Discrimination Method. In Proceedings of the 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), Xi'an, China, 7–8 November 2021; pp. 428–432.
26. Zhou, Y.; Shang, S.; Song, X.; Zhang, S.; You, T.; Zhang, L. Intelligent Radar Jamming Recognition in Open Set Environment Based on Deep Learning Networks. *Remote Sens.* **2022**, *14*, 2410–2424. [CrossRef]
27. Wang, L.; Yang, X.; Tan, H.; Bai, X.; Zhou, F. Few-Shot Class-Incremental SAR Target Recognition Based on Hierarchical Embedding and Incremental Evolutionary Network. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–11. [CrossRef]
28. Wei, Q. R.; He, H.; Zhao, Y.; Li, J.A. Learn to Recognize Unknown SAR Targets From Reflection Similarity. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
29. Song, Q.; Chen, H.; Xu, F.; Cui, T.J. EM Simulation-Aided Zero-Shot Learning for SAR Automatic Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1092–1096. [CrossRef]
30. Bernardino, R.P.; Torr, P.H.S. An embarrassingly simple approach to zero-shot learning. In Proceedings of the 32nd international conference on Machine learning (ICML'15), Lille, France, 6–11 July 2015.
31. Kodirov, E.; Xiang, T.; Gong, S. Semantic Autoencoder for Zero-Shot Learning. In Proceeding of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4447–4456.
32. Zhang, L.; Xiang, T.; Gong, S. Learning a Deep Embedding Model for Zero-Shot Learning. In Proceeding of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3010–3019.
33. Ma, B.J.; Qi, J.; Su, H.Y.; Zhang, X.F. One-dimensional Radar Active Jamming Signal Recognition Method Based on Bayesian Deep Learning. *Signal Process.* **2023**, *39*, 235–243.
34. Qu, Q.; Wei, S.; Liu, S.; Liang, J.; Shi, J. JRNet: Jamming Recognition Networks for Radar Compound Suppression Jamming Signals. *IEEE Trans. Veh. Technol.* **2020**, *69*, 15035–15045. [CrossRef]
35. Tang, Y.; Zhao, Z.; Chen, J.; Zheng, S.; Ye, X.; Lou, C.; Yang, X. Open world recognition of communication jamming signals. *China Commun.* **2023**, *20*, 199–214. [CrossRef]
36. Wang, J.Q. Study and Implementation of Radar Active Jamming Type Discrimination. Master's Thesis, Xidian University, Xi'an, China, 2014.
37. Zhang, H.; Yu, L.; Chen, Y.; Wei, Y. Fast Complex-Valued CNN for Radar Jamming Signal Recognition. *Remote Sens.* **2021**, *13*, 2867. [CrossRef]
38. Fu, Y.; Hospedales, T.M.; Xiang, T.; Gong, S. Trans-ductive multi-view zero-shot learning. *IEEE Tran. PAMI* **2015**, *37*, 2332–2345. [CrossRef]
39. Dong, Y.; Jiang, X.; Zhou, H.; Lin, Y.; Shi, Q. SR2CNN: Zero-Shot Learning for Signal Recognition. *IEEE Trans. Signal Process.* **2021**, *69*, 2316–2329. [CrossRef]
40. Yang, L.; Li, Q.; Shao, H.Z. An Open Set Recognition Algorithm of Electromagnetic Target Based on Metric Learning and Feature Subspace Projection. *Acta Electron. Sin.* **2022**, *17*, 1310–1318.
41. Jleed, H.; Martin, B. Incremental multiclass open-set audio recognition. *Int. J. Adv. Intell. Inform.* **2022**, *8*, 251–270. [CrossRef]
42. Geng, X.; Dong, G.; Xia, Z.; Liu, H. SAR Target Recognition via Random Sampling Combination in Open-World Environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 331–343. [CrossRef]