Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6309958, 14 pages
https://doi.org/10.1155/2022/6309958

Research Article

Virtual Adversarial Training-Based Semisupervised Specific Emitter Identification

CunXiang Xie,[1] **LiMin Zhang**,[1] **and ZhaoGen Zhong**[2]

[1]*Department of Information Fusion, Naval Aviation University, Yantai 264001, China*
[2]*The School of Aviation Basis, Naval Aviation University, Yantai 264001, China*

Correspondence should be addressed to ZhaoGen Zhong; zhongzhaogen@163.com

Received 21 November 2021; Revised 15 December 2021; Accepted 17 December 2021; Published 7 January 2022

Academic Editor: Junjuan Xia

Deep learning is a new direction of research for specific emitter identification (SEI). Radio frequency (RF) fingerprints of the emitter signal are small and sensitive to noise. It is difficult to assign labels containing category information in noncooperative communication scenarios. This makes network models obtained by conventional supervised learning methods perform unsatisfactorily, leading to poor identification performance. To address this limitation, this paper proposes a semisupervised SEI algorithm based on bispectrum analysis and virtual adversarial training (VAT). Bispectrum analysis is performed on RF signals to enhance individual discriminability. A convolutional neural network (CNN) is used for RF fingerprint extraction. We used a small amount of labelled data to train the CNN in an adversarial manner to improve the antinoise performance of the network in a supervised model. Virtual adversarial samples were calculated for VAT, which made full use of labelled and large unlabelled training data to further improve the generalization capability of the network. Results of numerical experiments on a set of six universal software radio peripheral (USRP; model B210) devices demonstrated the stable and fast convergence performance of the proposed method, which exhibited approximately 90% classification accuracy at 10 dB. Finally, the classification performance of our method was verified using other evaluation metrics including receiver operating characteristic and precision-recall.

1. Introduction

Specific emitter identification (SEI) refers to the technology used to identify individual emitters using distinctive external features of the signal called the radio frequency (RF) fingerprints [1]. This is obtained from the differences between the hardware characteristics of individual emitters and is hard to reproduce and/or eliminate. Moreover, the RF fingerprints of an emitter have unique characteristics that are independent of the content of the signal and may show consistency in different parts of the signal from the same emitter. Even for emitters produced by the same manufacturer and batch of equipment, the RF fingerprints are still different from each other. Therefore, the RF fingerprints can be used as a unique identification measure that enables SEI technology to be employed in military and civilian applications [2, 3].

In recent years, researchers have proposed various approaches for RF fingerprint extraction, which is a method to obtain the hardware characteristics of the emitter. For example, Padilla et al. [4] proposed methods for extracting parameters (transient waveform, instantaneous phase, and amplitude of the signal) of the RF signals, which are used as RF fingerprints, by means of the preamble in the communication. The method can identify 28 different Wi-Fi devices with an accuracy higher than 95%. However, because this method can only be applied to communication signals with preamble, its application scope is limited. Lopez-Risueno et al. [5] proposed using short-time Fourier transform to obtain the time-frequency energy spectrum of the signal, where the fingerprint features of the emitter signal are extracted based on the differences in the time-frequency energy spectrum. However, this method is based on a linear transformation, which is not suitable for nonlinear radiation source signals. Zhou et al. [6] proposed a feature extraction method based on the bispectrum-radon transform, which used bispectrum analysis to characterize the RF fingerprints and completed feature compression through radon transform. The method identified 6 ADS-B emitters with an accuracy of 90.25%. The high-order spectrum analysis method, however, is only able to extract some of the features of the emitter signal while losing some of the important subtle features, which results in a lower identification performance. Yuan et al. [7] extracted 13 types of feature parameters of emitter transient characteristics through empirical mode decomposition (EMD) and Hilbert transform to form RF fingerprints and effectively identified 8 mobile phones. Moreover, the method can theoretically be used with any type of emitters as it does not require prior information for the RF signal. However, the method only applies to the transient signal of an emitter, which is challenging to capture in practical applications. Satija et al. [8] proposed an SEI approach based on variational mode decomposition and spectral features (VMD-SF). The advantage of the method lies in its adaptability to both single hop and relaying scenarios under both AWGN and flat-fading channels. Further, the method has a low computational cost and satisfactory real-time performance. However, the performance of this method must be verified using simulation signal data, so its practicability requires further research.

Although the conventional feature extraction scheme can reflect and amplify the individual differences of emitters, it needs to blindly try all the previous manually predefined RF fingerprint features to find an effective feature extraction method for a specific task. However, the complexity of the emitter signal renders it impossible to represent the signals using a unified mathematical model. Therefore, for the target signal of interest, the choice of the feature extraction method can only depend on the subjective judgment and cognitive level of the researchers, which cannot fully reflect the differences between individual emitters.

With advancements in artificial intelligence, techniques such as machine learning (ML), deep learning (DL), and reinforcement learning (RL) have been widely used in many fields [9]. In addition to the applications in traditional computer vision (CV), natural language processing (NLP), etc., deep learning technologies have gained great success in emerging fields, such as the Internet of Things (IoT) [9], physical layer communication [10, 11], and edge intelligence [12–14]. Furthermore, deep learning is now used in SEI, which is a new research direction that can comprehensively and deeply extract the fingerprint features of the emitter signal through neural networks to improve recognition performance. Wong et al. [15] used a convolutional neural network (CNN) to estimate the gain and phase deviations of the in-phase and quadrature components of an emitter signal, achieving SEI based on the estimated gain deviation and phase deviation. Their method does not require preprocessing, such as signal synchronization and carrier frequency tracking, and can be applied to signals of multiple modulation types. He et al. [16] used a long short-term memory (LSTM) network to learn RFFs from preprocessed RF signals. Compared with CNN, LSTM is more suitable for processing time series such as one-dimensional signals. This implies that it can achieve better identification performance. However, LSTM also introduces significant computational costs and training difficulties, resulting in poor real-time performance. Based on network models such as CNN and LSTM, some new deep learning-based network models and algorithms for SEI have been proposed recently. Qian et al. [17] proposed an approach of multilevel sparse representation-based identification for SEI, which comprehensively used the CNN for RF fingerprint extraction and principal component analysis for sparse representation. The method can classify nine transmitters with a classification accuracy of over 90% using a small number of training samples. A complex-valued neural network (CVNN) proposed in [18] was used to process complex baseband signals to perform SEI. In addition, a network compression algorithm was proposed to reduce the model size and decrease the training complexity. The CVNN-based SEI method can achieve nearly 100% recognition accuracy at high SNRs. Furthermore, the network size can be compressed by nearly 70%–90%, and the training complexity decreases at different SNR levels. The fixed neural network structure typically has the problem of poor flexibility when processing complex RF signals at different scenarios. Hence, a balanced network architecture search (NAS) mechanism proposed in [19] was applied to conduct SEI. The framework uses a recurrent neural network (RNN) as the controller and cooperates with a balance function to search for the optimal network structure, thereby providing a suitable scheme for processing RF signals at current SEI tasks.

SEI based on deep learning has been extensively studied; however, most studies focus on supervised learning models, which assume that all training samples have label information. SEI is used in noncooperative scenarios in most cases, and a typical application is radio surveillance, which is the application scenario for the SEI system considered in this study. SEI can be used to distinguish between legal and illegal radio stations, identify different types of illegal radio stations (which can be targeted to eliminate their interferences), and effectively control the utilization of spectrum resources. It is difficult for illegal radio stations to be assigned label information representing the signal category. Thus, only a few signal samples contain label information. Using a large number of unlabelled signal samples combined with a few labelled signal samples to conduct semisupervised SEI-based deep learning is a worthwhile research area. To address this problem, we propose a semisupervised SEI scheme based on bispectrum analysis and virtual adversarial training (VAT). Bispectrum analysis, a signal preprocessing method, is used as the representation of RF fingerprints. On this basis, we used the CNN to extract the RF fingerprints, and VAT is performed to fully utilize the labelled and large amount of unlabelled signal samples to conduct semisupervised SEI. Compared with other existing supervised learning-based SEI methods, our research focuses on semisupervised learning-based SEI. On the one hand, our method improves the antinoise performance of the SEI system by changing the traditional network training mode. On the other hand, more importantly, our method makes full use of many of unlabelled signal samples to carry out semisupervised training through VAT, thus applying to SEI in noncooperative scenarios.

The main contributions of this work are summarized as follows:

(1) To enhance the individual discriminability of different emitters, the bispectrum distribution is used as a characteristic representation of the RF signal, which lays the foundation for RF fingerprint extraction based on the CNN

(2) Considering that RF fingerprints are susceptible to noise, we perform adversarial training [20, 21] on the CNN to improve the antinoise performance of the network. In addition, we propose an algorithm for calculating the minimum number of adversarial samples for use during adversarial training to maximize the antinoise performance of the network

(3) After improving the antinoise performance of the network through adversarial training, we further propose a VAT method to train the CNN using labelled and large amounts of unlabelled signal samples collected in noncooperative scenarios, which can greatly enhance the generalization capacity of the network and improve the identification performance of the system

(4) Various experimental results are presented. We first evaluate the convergence proposed method. Then, we measure the classification accuracy of the proposed framework along several factors, including SNRs, ratio of labelled to unlabelled samples, communication propagation channel, and number of emitters. Simulation results show that our method performs well for SEI in noncooperative scenarios

The remainder of this paper is organized as follows. In Materials and Methods, we introduce the signal preprocessing method based on bispectrum analysis and the basic concepts of CNNs, adversarial training, and VAT. Our proposed method of VAT-based semisupervised SEI is also introduced in this section, together with the experiments conducted on a real-world RF dataset generated through a software-defined radio (SDR) platform. Results and Discussion present and discuss the experimental results. Finally, we conclude the paper.

2. Materials and Methods

2.1. Bispectrum Distribution of RF Signal. Bispectrum analysis is a special case of higher-order spectral analysis, which has demonstrated its superiority in processing non-Gaussian and nonstationary signals [22, 23]. The bispectrum distribution of the signal can be obtained by calculating the two-dimensional Fourier transform of the third-order cumulant of the signal:

$$B(\omega_1, \omega_2) = \sum_{\tau_1} \sum_{\tau_2} c_{3x}(\tau_1, \tau_2) e^{-j(\omega_1 \tau_1 + \omega_2 \tau_2)}, \quad (1)$$

where ω_1 and ω_2 represent the two-dimensional frequency and the third-order cumulant $c_{3x}(\tau_1, \tau_2)$ can be expressed as

$$c_{3x}(\tau_1, \tau_2) = E[x^*(t) x(t + \tau_1) x(t + \tau_2)]. \quad (2)$$

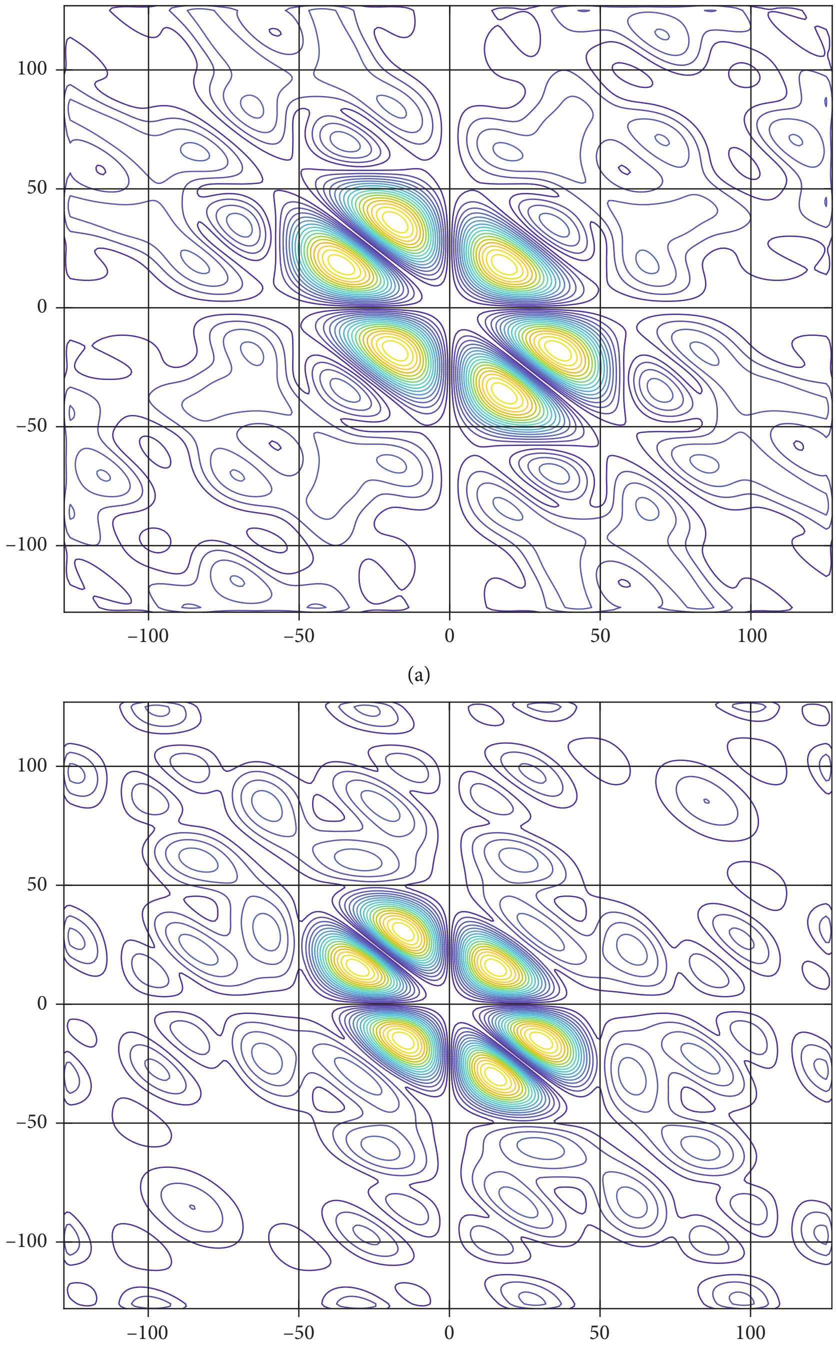
Figure 1 shows the bispectrum distributions of two different RF signals $x_1(t)$ and $x_2(t)$. It shows the visible differences in bispectrum distribution features, which demonstrate that the bispectrum analysis is an effective method of signal preprocessing to enhance individual discriminability.

2.2. Convolutional Neural Network. It is difficult to fully extract RF fingerprints using the methods mentioned earlier. In this study, we employed neural networks which have shown excellent performance in processing large quantities of data and extracting deep features [24, 25]. We used a CNN for RF fingerprint feature extraction and signal classification. A CNN is mainly composed of three structures: a convolution layer, an activation function, and a pooling layer. The function of the convolution layer is mainly to extract the features of the signals. In the convolution process, multiple convolution cores are used to convolve the respective feature maps of the previous layer to realize feature extraction and mapping. Each convolution layer includes many convolution cores, thus realizing multifeature extraction of signals. In a CNN, each feature map uses the same weight parameter, which is called weight sharing. Weight sharing reduces the number of parameters in the model and maintains displacement invariance for the position and size of the input. The activation function is used to introduce nonlinear factors into the convolutional layer for it to avoid becoming a linear combination of input vectors which helps in extracting complex and deep-level features [26]. The pooling layer is used for subsampling, whose main goal is to reduce the resolution of feature maps and thus facilitate the extraction of deep-level features.

When using a CNN to achieve signal classification, the feature space output by the CNN is used as the input of the fully connected neural network (FCN). The features extracted from the convolution layer are vectorized through the fully connected layer, thus mapping the distributed feature representation to the label space of samples. Finally, the output probability is mapped through the SoftMax function, and the maximum output probability corresponds to the recognized signal category.

2.3. Adversarial Training. Adversarial training is an important method to enhance the robustness of neural networks. The training samples mixed with some minor disturbances, which are subtle but can cause misidentification, are fed to the neural network to enable it to adapt to changes and be robust to interference. This method is widely used to defend against adversarial samples [27], which is one of the most effective means of adversarial defence. In contrast, adversarial training can be used for adversarial attacks [28]. The attacker constructs the adversarial input vector based on the adversarial sample so that the machine learning model misjudges, which is called an adversarial attack. Adversarial training can also be extended as VAT for semisupervised learning based on unlabelled data [29].

The RF fingerprints of the emitter signal caused by the subtle hardware differences of emitters are not generally evident and can be easily interfered by noise and be misidentified. To solve this problem, we used the adversarial training



(a)

(b)

Figure 1: Bispectrum distributions of two different signals.

method to train the CNN by adding adversarial samples to the training data, which improves the network's recognition of adversarial samples and improves the robustness of the network. The key to adversarial training is the generation of adversarial samples. For a trained CNN, a subtle perturbation Δx is added to the input vector x. When the network loss reaches its maximum value, the corresponding $x + \Delta x$ becomes the adversarial sample [27]. The subtle perturbation Δx most likely results in the misjudgment of the original neural network. Therefore, the CNN needs to be trained with the input vector $x + \Delta x$ and the real label y_{true} to improve its recognition performance for adversarial samples, which will result in enhanced network robustness [28].

Let x_l, y_l, and x_{ul} represent the labelled training samples, their corresponding labels, and unlabelled training samples, respectively. We also assumed that x_l and x_{ul} are uniformly represented by x. The loss function of the adversarial training is defined as [28]

$$L_{\text{adv}}(x_l, y_l\,; \theta) = D[q(y_l \mid x_l), p(y_l \mid x_l + r_{\text{adv}}, \theta)], \tag{3}$$

```
Required:
$\boldsymbol{x}$: signal data, including labelled data $\boldsymbol{x}_l$ and unlabelled data $\boldsymbol{x}_u$
$\boldsymbol{y}_l$: labels corresponding to the labelled signal data
iter: number of training iterations
$\lambda_{\mathrm{adv}}, \lambda_{\mathrm{vadv}}$: regularization coefficients of (virtual) adversarial training
$L(\cdot)$: supervised loss function of network model
$L_{\mathrm{adv}}(\cdot)$: loss function of adversarial training
$LDS_{\mathrm{adv}}(\cdot)$: full loss function of adversarial training
$L_{\mathrm{vadv}}(\cdot)$: loss function of virtual adversarial training
$LDS_{\mathrm{vadv}}(\cdot)$: full loss function of virtual adversarial training
$\theta$: parameters of network model
$\eta$: learning rate
$\varepsilon_{\max}^{(0)}$: the upper limit of the perturbation weighting coefficient
$\varepsilon_{\min}^{(0)}$: the lower limit of the perturbation weighting coefficient
$t = 0$: number of iterations of the perturbation weighting coefficient
for $i = 1$ to iter do
    $r_{\mathrm{adv}} \longleftarrow \arg\max_{r:\|r\|\le\varepsilon} D[q(y|x_l), p(y|x_l + r, \theta^i)]$
    do
        $\varepsilon_{\mathrm{ave}}^{(t)} = (\varepsilon_{\max}^{(t)} + \varepsilon_{\min}^{(t)})/2$
        $x_l^{(t)} \longleftarrow x_l + \varepsilon_{\mathrm{ave}}^{(t)} r_{\mathrm{adv}}$
        $\begin{cases} \varepsilon_{\min}^{(t+1)} = \varepsilon_{\mathrm{ave}}^{(t)}, & y_{\mathrm{pre}}(x_l^{(t)}) = y_l \\ \varepsilon_{\max}^{(t+1)} = \varepsilon_{\mathrm{ave}}^{(t)}, & y_{\mathrm{pre}}(x_l^{(t)}) \ne y_l \end{cases}$
        $t \longleftarrow t + 1$
    while $\varepsilon_{\max}^{(t)} - \varepsilon_{\min}^{(t)} \ge \varepsilon_{\mathrm{acc}}$.
    end while
    $\varepsilon^* \longleftarrow \varepsilon_{\max}^{(t)}$, $r_{\mathrm{adv}} \longleftarrow \varepsilon^* \cdot r_{\mathrm{adv}}$.
    Initialize $\varepsilon_{\max}^{(0)}$, $\varepsilon_{\min}^{(0)}$, and $t$
    $L_{\mathrm{adv}}(x_l, y_l; \theta) = D[q(y_l|x_l), p(y_l|x_l + r_{\mathrm{adv}}, \theta)]$
    $LDS_{\mathrm{adv}}(\theta) = L(x_l, y_l; \theta) + \lambda_{\mathrm{adv}} L_{\mathrm{adv}}(x_l, y_l; \theta)$
    $\theta^{i+1} \longleftarrow \theta^i - \eta \nabla_\theta LDS_{\mathrm{adv}}(\theta)$
end for
for $j = \mathrm{iter} + 1$ to 2iter do
    $r_{\mathrm{vadv}} = \arg\max_{r:\|r\|\le\varepsilon} D[p(y|x, \theta^j), p(y|x + r, \theta^j)]$
    $L_{\mathrm{vadv}}(x, \theta) = D[p(y \mid x, \theta^j), p(y \mid x + r_{\mathrm{vadv}}, \theta^j)]$
    $LDS_{\mathrm{vadv}}(\theta) = L(x_l, y; \theta) + \lambda_{\mathrm{vadv}} L_{\mathrm{vadv}}(x, \theta)$
    $\theta^{j+1} \longleftarrow \theta^j - \eta \nabla_\theta LDS_{\mathrm{vadv}}(\theta)$
end for
```

Algorithm 1: Semisupervised SEI based on VAT.

where $D[q, p]$ is used to measure the divergence between two distributions p and q; $q(y_l \mid x_l)$ is the true distribution of the output label, which can be approximated by a one-hot vector; $p(y_l \mid x_l + r_{\mathrm{adv}}, \theta)$ is the data distribution predicted by the CNN; and r_{adv} represents an adversarial sample, which is equivalent to Δx and can be expressed as follows:

$$r_{\mathrm{adv}} = \arg\max_{r:\|r\|\le\varepsilon} D[q(y_l \mid x_l), p(y_l \mid x_l + r, \theta)]. \tag{4}$$

After obtaining the adversarial samples of all signal types, we regarded the loss value generated by the adversarial sample as part of the original loss value and added it to the original loss function in the form of regularization, which is expressed as

$$LDS_{\mathrm{adv}}(x_l, y_l; \theta) = \sum_{(x_l, y_l) \in D_l} L(x_l, y_l, \theta) + \lambda_{\mathrm{adv}} \sum_{(x_l, y_l) \in D_l} L_{\mathrm{adv}}(x_l, y_l; \theta), \tag{5}$$

where D_l represents the labelled training sets, λ_{adv} represents the weighting coefficient of regularization, and $L(x_l, y_l; \theta)$ represents the supervised loss function for the CNN, which can be calculated through the cross-entropy function.

$$L(x_l, y_l; \theta) = -y_l \log y_{\mathrm{pre}}(x_l; \theta). \tag{6}$$

The parameter θ of the CNN is tuned using the backpropagation algorithm:

$$\theta^{i+1} \longleftarrow \theta^{i} - \eta \nabla_{\theta} LDS_{\text{adv}}(\theta), \tag{7}$$

where η represents the learning rate.

2.4. Virtual Adversarial Training (VAT). The adversarial training algorithm trains a CNN in a supervised learning model, where all the training samples must be labelled. However, in noncooperative communication scenarios, only a small number of signal samples are labelled. Using a small number of labelled samples to train the CNN through adversarial training results in poor generalization capacity.

To exploit the information in the unlabelled signals, we adopted VAT [30] to employ the labelled and unlabelled training data and to smoothen the output space of the neural network. This minimized the change in the output of the neural network where its input was locally perturbed. Therefore, VAT proved effective for semisupervised learning [31].

However, in the semisupervised learning model, there are many unlabelled training samples such as $q(y \mid x_{\text{ul}})$. Therefore, unlabelled training samples cannot be used to train the CNN through adversarial training algorithms. Note that, for a large amount of labelled training samples, $p(y \mid x, \theta)$ approaches $q(y \mid x)$. We can use "virtual" labels that are probabilistically generated from $p(y \mid x, \theta)$ rather than labels unknown to the user. We then compute the adversarial direction based on these virtual labels [32, 33]. The loss function for VAT can be expressed as

$$L_{\text{vadv}}(x, \theta) = D[p(y \mid x, \theta), p(y \mid x + r_{\text{vadv}}, \theta)], \tag{8}$$

where θ represents the weight parameters of the neural network in the current training state and r_{vadv} represents the virtual adversarial sample:

$$r_{\text{vadv}} = \underset{r:\|r\| \le \varepsilon}{\arg\max} D[p(y \mid x, \theta), p(y \mid x + r, \theta)]. \tag{9}$$

After obtaining the virtual adversarial samples, the full loss function is given by

$$LDS_{\text{vadv}}(x, \theta) = \sum_{(x_l, y_l) \in D_l} L(x_l, y_l, \theta) + \lambda_{\text{vadv}} \sum_{x \in D_l, D_{\text{ul}}} L_{\text{vadv}}(x, \theta), \tag{10}$$

where D_l and D_{ul} represent the labelled and unlabelled training sets, respectively; $\lambda_{\text{vadv}} > 0$ represents the regularization coefficient that needs to be set in advance. $L(x_l, y_l; \theta)$ represents the supervised loss function of the CNN, which is equivalent to Equation (6).

Equation (10) shows that both labelled data and a large amount of unlabelled data are used to carry out semisupervised training. The labelled data was combined with the unlabelled data to conduct virtual adversarial training. Supervised learning can use labelled data to guide network training. The loss function of VAT $L_{\text{vadv}}(x, \theta)$ can be regarded as a measure of the local smoothness of the current network, and its optimization can smooth the network output space. λ_{vadv}, as the regularization coefficient, is used to control the relative balance between supervised learning and virtual adversarial training, ensuring the effect of semisupervised training.

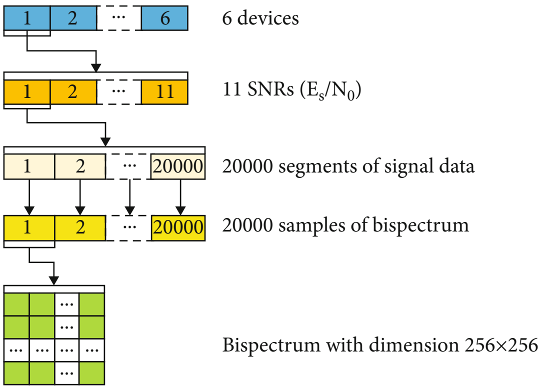

1
2
···
6
6 devices
1
2
···
11
11 SNRs (E$_s$/N$_0$)
1
2
···
20000
20000 segments of signal data
1
2
···
20000
20000 samples of bispectrum
···
Bispectrum with dimension 256×256

Figure 2: Dataset structure.

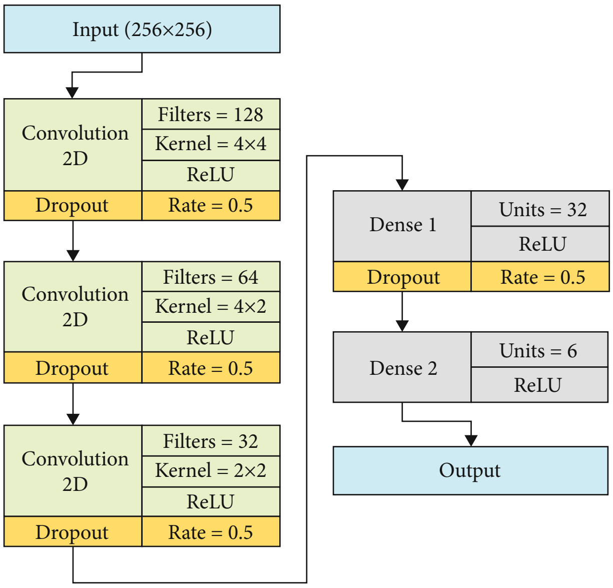


Figure 3: Architecture details of the network model.

Finally, the parameter θ of the CNN is tuned according to the backpropagation algorithm.

$$\theta^{j+1} \longleftarrow \theta^{j} - \eta \nabla_{\theta} LDS_{\text{vadv}}(\theta). \tag{11}$$

2.5. Semisupervised SEI Based on VAT. VAT is a semisupervised learning model that can be applied to SEI in noncommunication scenarios. However, networks trained with a considerably small amount of labelled signal data have poor generalization capability; these networks then probably assign the wrong virtual label for the unlabelled signal data, which can cause severely harmful effects for the subsequent classification. To solve this problem, we first trained the network model with labelled signal data via adversarial training, which improves the generalization capability of the network. Furthermore, we calculated a specific value ε^* for the

Table 1: Architecture details of the CNN on different depths.

Depth	Conv-1	Conv-2	Conv-3	Conv-4	Conv-5
1	Filters = 128 Size = 4 × 4				
2	Filters = 128 Size = 4 × 4	Filters = 64 Size = 4 × 2			
3	Filters = 128 Size = 4 × 4	Filters = 64 Size = 4 × 2	Filters = 32 Size = 2 × 2		
4	Filters = 128 Size = 4 × 4	Filters = 64 Size = 4 × 2	Filters = 32 Size = 2 × 2	Filters = 32 Size = 2 × 2	
5	Filters = 128 Size = 4 × 4	Filters = 64 Size = 4 × 2	Filters = 32 Size = 2 × 2	Filters = 32 Size = 2 × 2	Filters = 16 Size = 2 × 2

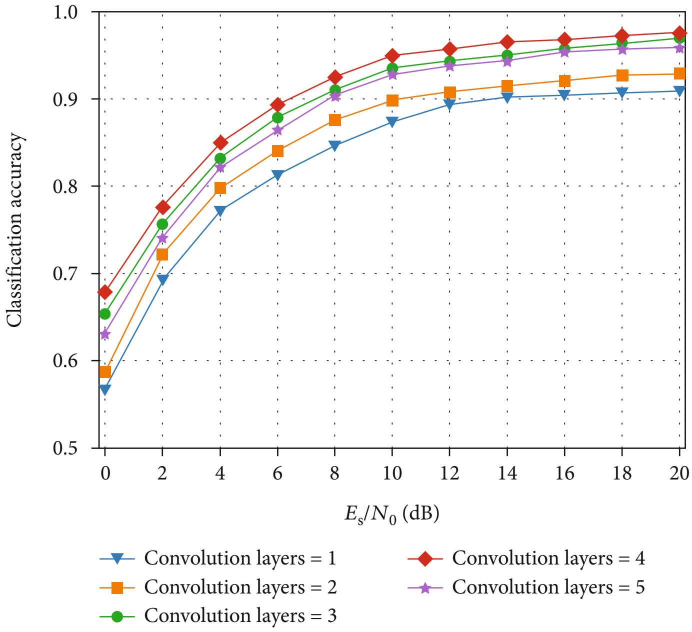


Figure 4: Classification accuracy on different network depths.

perturbation weighting coefficient ε of the adversarial sample, which is the minimum value of ε that causes misidentification. At this time, the corresponding r_{adv} is the smallest perturbation that can lead to misidentification. Using this as an adversarial sample for adversarial training maximizes the antinoise performance of the neural network. The network parameters obtained through adversarial training will be directly used in VAT to conduct semisupervised SEI. Therefore, the procedure of semisupervised SEI based on VAT is concluded in Algorithm 1.

2.6. Implementation Details. The calculation of (virtual) adversarial samples is essential for the (virtual) adversarial training algorithm. However, in practice, we cannot obtain a closed form of r_{adv} or r_{vadv} calculated in Equation (4) or Equation (8). Therefore, in this section, we provide the core implementation details of the proposed algorithm: calculating (virtual) adversarial samples in an approximate manner.

The calculation of r_{adv} in this study can be approximated with a linear approximation of D with respect to r in Equation (4), which can be expressed as

$$r_{\text{adv}} \approx \frac{\nabla_{x_l} L(x_l, y_l; \theta)}{\left\| \nabla_{x_l} L(x_l, y_l; \theta) \right\|_2}. \tag{12}$$

For a neural network model, the calculation of $\nabla_{x_l} L(x_l, y_l; \theta)$ can be computed through forward- and backpropagation.

Furthermore, the calculation of r_{vadv} is performed in an approximate manner, which can be described as follows: for an input training sample x, a random unit vector d of the same size that obeys the standard Gaussian distribution is generated. Then, r_{vadv} is obtained by taking the gradient of D with respect to r on $r = \xi d$ for x.

$$r_{\text{vadv}} \approx \frac{\nabla_r D[p(y \mid x, \theta), p(y \mid x + r, \theta)]|_{r=\xi d}}{\left\| \nabla_r D[p(y \mid x, \theta), p(y \mid x + r, \theta)]|_{r=\xi d} \right\|_2}. \quad (13)$$

For the neural network model, the calculation of $\nabla_r D$ can be computed through forward- and backpropagation.

2.7. Signal Data Collection and Experimental Setup. We demonstrate our network model based on a software-defined radio platform composed of GNU Radio and seven USRP model B210 devices. By combining GNU Radio with the USRP, we then define the transceiver of radio signals through the PC to form a complete communication system composed of software and hardware. This platform realizes communication functions, and signal modulation and demodulation can be done at the software level.

A computer, running on Ubuntu 18.04, was connected to USRP to build a communication system. We used six USRPs as the transmitters and one USRP as the receiver, and then, we collected six types of RF signals through six USRPs, which operated at a 2.4 GHz centre frequency, and the received signals were sampled at a rate of 16 MHz. The signal modulation mode was QPSK, and the bandwidth was 1.2 MHz.

For each emitter, 20,000 segments of RF signals were collected from the lab environment; thus, the signal to noise ratio (SNR) of the signals was high. After the preliminary measurement, the SNR was found to be more than 50 dB. Therefore, we assumed that the signal was not affected by noise. For each class sampled signal, we calculated the average symbol energy within the input frame as E_s and used MATLAB to add different levels of simulated additive white Gaussian noise (AWGN) to set the ratio of symbol energy to noise density (E_s/N_0) as 0 dB, 2 dB, …, 20 dB, respectively. The signal data polluted by noise were processed by bispectrum analysis to obtain bispectrum distributions, which have a uniform size of 256 × 256. Figure 2 shows the dataset structure of our experiments. The dataset contains six classes of sampled signal data. Each class signal includes 20,000 segments at a specific SNR (E_s/N_0), each of which is transformed to a bispectrum distribution with the dimension of 256 × 256.

For 20,000 bispectrum distributions of each class signal at a specific SNR, 80%, 10%, and 10% were allocated to training, validation, and testing, respectively. For the training and validation datasets, we set the ratio of labelled data to unlabelled data at 10%.

The proposed semisupervised SEI network architecture was built on the Keras framework based on TensorFlow, and the network was trained on a Windows 10, Intel (R) core (TM) i9-10900 CPU, 16 GB RAM, and NVIDIA GeForce RTX 3090 system.

The structure and the detailed architecture parameters of the CNN are shown in Figure 3. Three convolutional layers are utilized, and the ReLU function is used as the activation function of the convolutional layer, which has stable output and no vanishing gradient problem. The features extracted from the convolution layer are vectorized through the fully connected layer (Dense I), thus mapping the distributed feature representation to the label space of samples. Finally, the output probability is mapped through the SoftMax function of the last fully connected layer (Dense II). Dense II contains K neurons, corresponding to K transmitters. By comparing the output probabilities of K neurons, the maximum probability corresponds to the recognized emitter category.

Table 2: Hyperparameters of the network model.

Hyperparameters	Value
Optimizer	Adam
Learning rate	0.0004
Epochs	500
Batch size	1024
Dropout rate	0.5

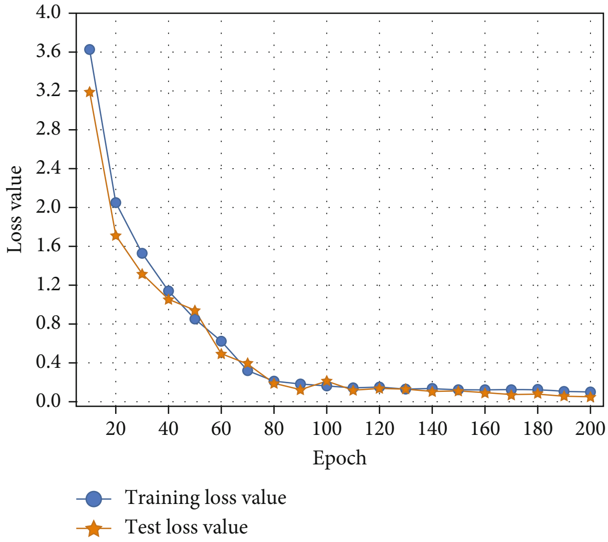


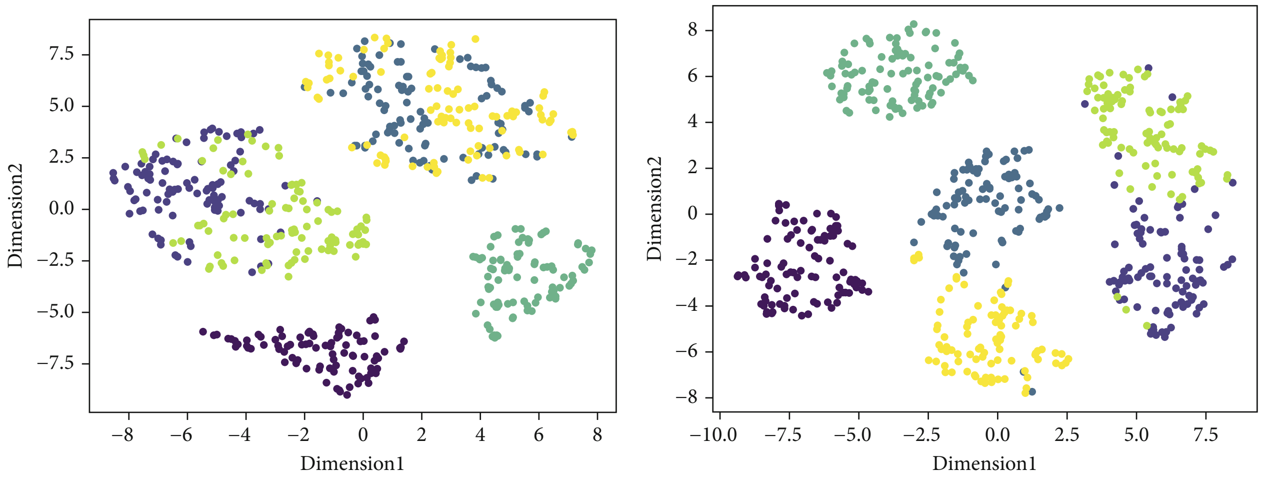
Figure 5: Training times and loss function graph of VAT.

In practice, the network depth (the number of convolution layers) is determined through simulations. Table 1 summarizes the architecture details of the CNN for different depths.

We evaluated the classification accuracy for different network depths based on the RF signal dataset mentioned above. Figure 4 shows that the classification accuracy improved with the increase in the number of convolution layers, and it remained at a high level when the number of convolution layers was 3. However, when the number of convolution layers reached 5, the network was too deep to fit the input signal data, and the classification accuracy slightly decreased. Considering classification accuracy and network complexity, the optimal network depth is 3 convolutional layers.

We also list the hyperparameters used to train the network model as summarized in Table 2.

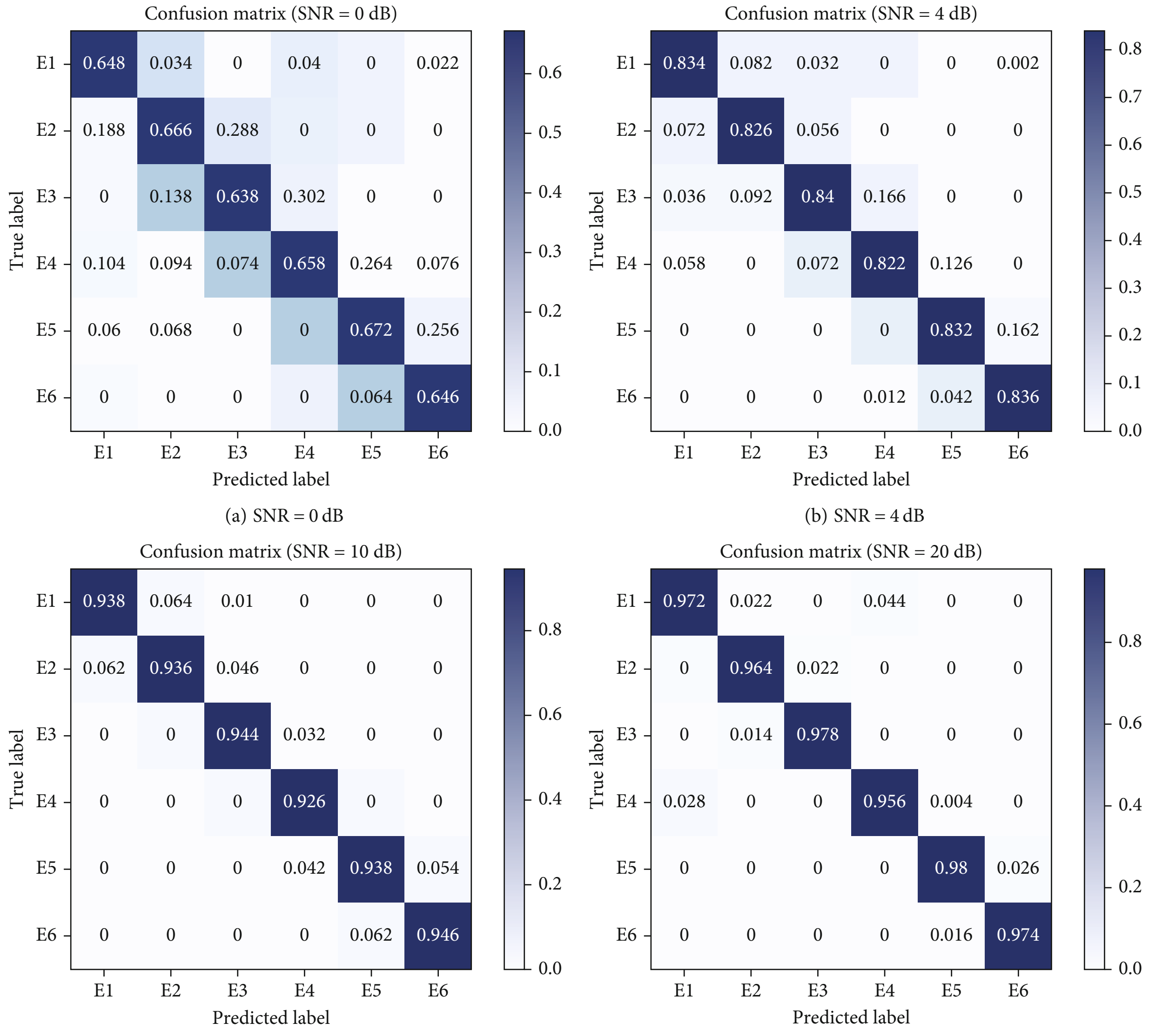
We used the validation dataset for hyperparameter tuning. Learning rate is the most significant hyperparameter, which directly controls the magnitude of the network gradient update during training and affects the effective tolerance



(a) Algorithm using only labelled data to train the CNN

(b) Proposed method

Figure 6: t-SNE dimensionality reduction distribution diagram.



(a) SNR = 0 dB

(b) SNR = 4 dB

(c) SNR = 10 dB

(d) SNR = 20 dB

Figure 7: Confusion matrix for the classification at different SNRs.

capability of the model. For tuning the learning rate, we first used a small amount of data to train the network to determine the magnitude of the learning rate. We then chose a specific value within this magnitude range as the initial learning rate. During the training process, as the number of training datasets increased, the learning rate decayed exponentially until the verification loss value converged, and the learning rate at this point was optimal. Batch size is a relatively independent hyperparameter that determines the direction in which the gradient decreases. Regarding the choice of the batch size, we determined several candidate values of 128, 256, 512, 1024, and 2048, and then evaluated how the classification accuracy in the validation dataset changed over time, and finally selected the batch size corresponding to the fastest improvement in classification performance over time. An epoch represents the training time. After each training epoch, the classification performance of the model on the validation dataset was evaluated, and the training was stopped when the classification performance stopped increasing. Therefore, generally, a large value was assigned to the epoch. The dropout rate was generally set to 0.5, which was used to prevent overfitting. By tuning hyperparameters, the network model can achieve better performance.

3. Results and Discussion

3.1. Convergence Performance. We evaluated the convergence performance of the neural networks trained using the proposed approach. We first collected RF signal data with an SNR of 10 dB from six USRPs, with each device representing a class of signal. The maximum epoch of network training was set at 200. Moreover, we chose the training loss value and test loss value as metrics to evaluate the convergence performance.

Figure 5 shows that the loss function of the network tended to be stable after approximately 80 training epochs, which means our approach has a relatively fast convergence speed. Moreover, the training loss function curve and the test loss function curve are relatively smooth with no noticeable fluctuation. This indicates that the training process was stable. The loss value of training and test procedure decreased as the number of iterations increased, i.e., the two curves exhibit a downward trend, which indicates that the network model performs well in both the training dataset and the test dataset and that no overfitting or underfitting problem occurred. The results show that our approach can be used for semisupervised training of neural networks.

3.2. Classification Accuracy. We first proved the superiority of the proposed method compared with that of the method using only labelled data to train CNN. The algorithm of t-distributed stochastic neighbour embedding (t-SNE), [34] one of the best dimensional reduction methods, was used to visually display the feature parameters extracted from the neural network model. Figure 6 shows the t-SNE dimension reduction distribution diagram of feature parameters extracted through two different methods.

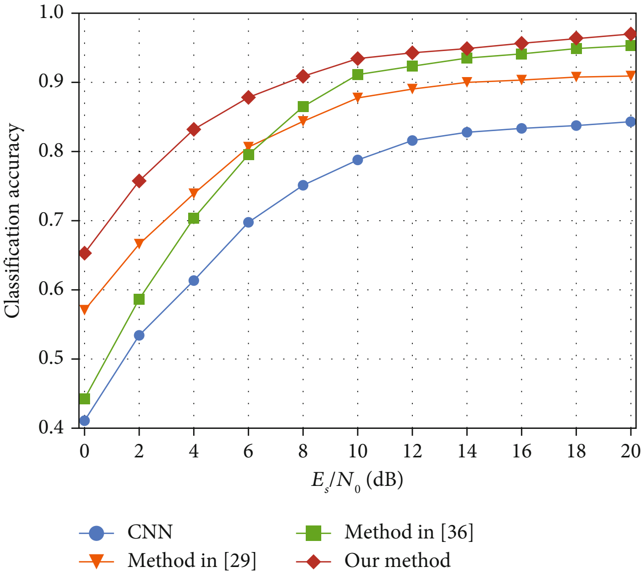


FIGURE 8: Classification accuracy under different SNRs.

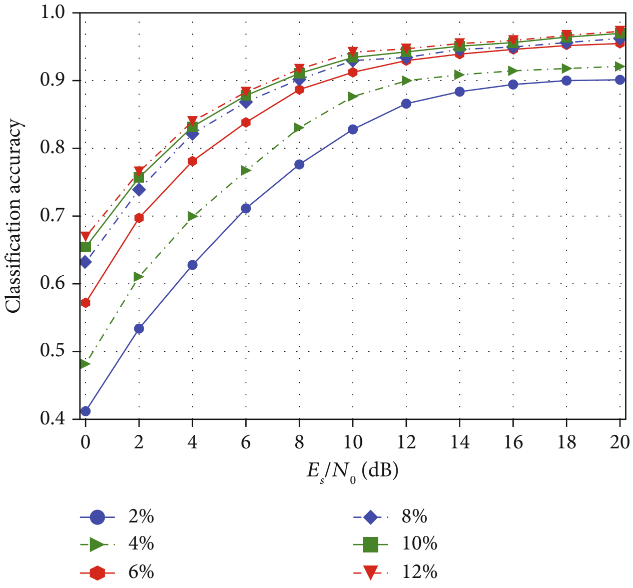


FIGURE 9: Classification accuracy under different ratios.

As shown in Figure 6, compared with the algorithm using only labelled data to train the CNN, the feature parameters of the RF fingerprints extracted by the proposed algorithm have stronger clustering within classes and greater differentiation between classes. This proves that the proposed VAT algorithm, which uses a large amount of unlabelled signal samples for training, can improve the neural network's generalization ability. This allows the network to extract the characteristic parameters of an individual emitter more comprehensively, thus improving the classification of RF signals.

We then considered four factors that significantly affect classification accuracy: (1) SNR, (2) ratio of labelled to unlabelled samples, (3) communication propagation channel, and (4) number of emitters.

Classification accuracy vs. SNR: first, we tested the classification accuracy on different SNRs. Both the labelled and

Table 3: Classification accuracy at different ratios.

Method	2%	4%	6%
Method in [29]	78.9%	82.2%	88.3%
Method in [35]	84.1%	87.5%	89.6%
Our proposed method	90.3%	91.8%	94.6%

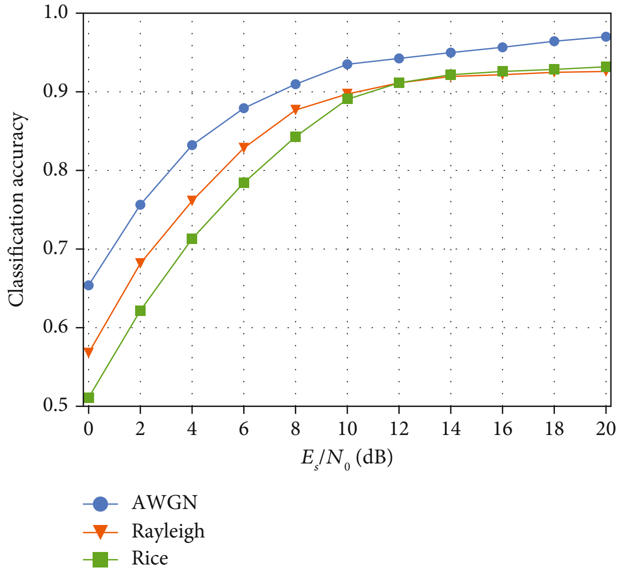


Figure 10: Classification accuracy on different propagation channels.

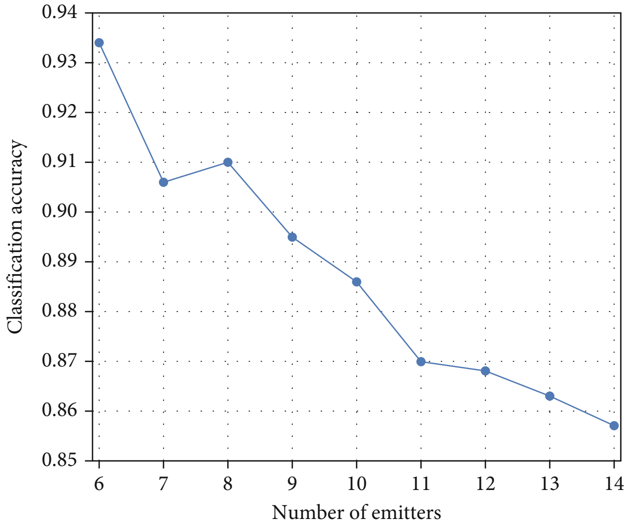


Figure 11: Classification accuracy under different number of emitters.

unlabelled data were used to conduct semisupervised training on CNN based on the proposed method. Figure 7 shows the confusion matrix of classification at different SNRs.

As shown in Figure 7, all emitters maintain a relatively average classification accuracy, and no serious confusion occurs in the classification between individual emitters. This shows that our method can fully extract RF fingerprints and effectively distinguish between individual emitters, thus proving the effectiveness of the deep learning-based SEI. In addition, our method can classify six emitters with an average accuracy of more than 83% at 4 dB and 93% at 10 dB, which demonstrates that our proposed VAT-based semisupervised SEI is robust to noise interference.

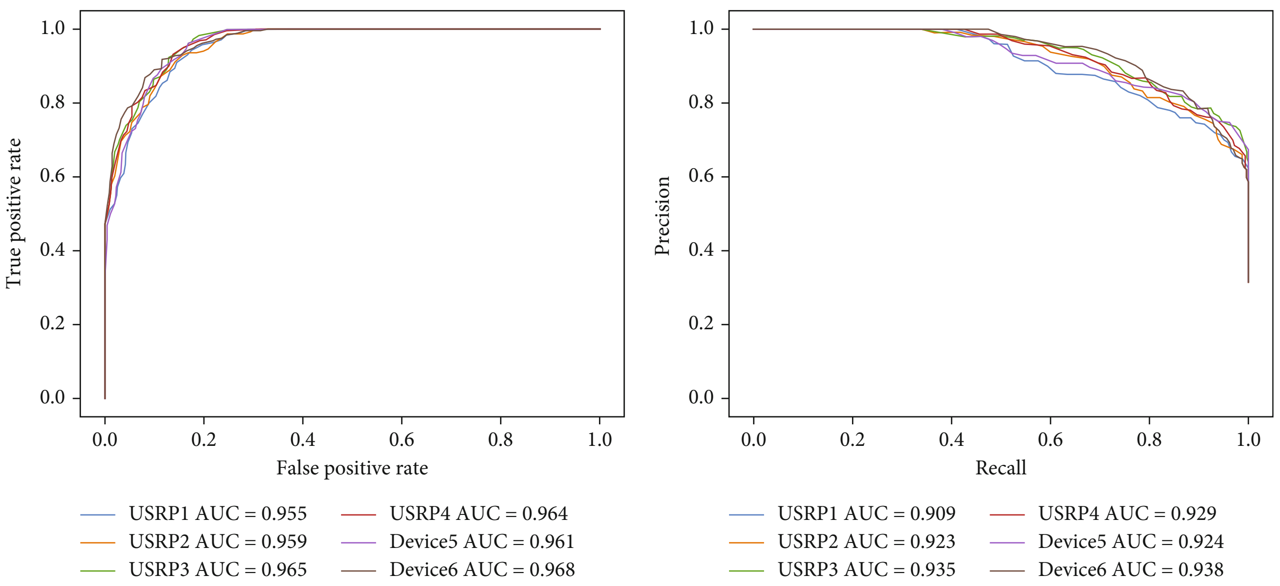
Then, we compared the classification accuracy of the proposed method with that of the method using only labelled data to train CNN and those previously proposed methods in [29, 35]. The experimental results are shown in Figure 8.

Figure 8 shows that our approach achieves the highest classification accuracy compared to other SEI schemes. Compared to the method using only labelled data to train the CNN, our method can improve classification accuracy by 15%–20% on average. This is because VAT can smoothen the output space of the network to enhancing its generalization ability. This effectively overcomes noise interference and improves classification accuracy. Compared to the method proposed in [29], our method improves classification accuracy by 5%–10% on average. This is because our improved VAT method augments the antinoise performance and generalization capability of the network through adversarial training in the pretraining process. This results in the assignment of accurate virtual labels for the unlabelled signal data, thus contributing to the effectives of VAT. Compared to the method based on metalearning proposed in [35], our method shows an advantage in terms of classification accuracy at low SNRs, although both methods a have similar classification accuracy only when SNR > 10 dB. Therefore, the experimental result shows that our method can adapt well to the task of SEI at low SNRs.

Classification accuracy vs. ratio of labelled to unlabelled samples: we evaluated classification accuracy on different labelled to unlabelled ratios. Similarly, we fixed the unlabelled data sample at 10,000 for each class signal and the labelled data sample from 200 to 1200, which means that the ratio of labelled samples to unlabelled samples was 2 to 12%. These labelled and unlabelled training samples with SNRs from 0 dB to 20 dB were used to train the CNN through the method of VAT.

Figure 9 shows that the classification accuracy improved when ratio of labelled to unlabelled samples increased from 2% to 6%, but stabilized when the ratio reached approximately 8%. Furthermore, even in the worst case of 2%, the classification performance did not deteriorate significantly. The classification accuracy was more than 80% at 10 dB and approximately 90% at 20 dB. According to these results, our approach needs only a small amount of labelled data samples to achieve a high and stable classification accuracy, giving it the ability to handle practical situations.

We further evaluated the classification performance of our method compared with that in [29, 35] at a ratio of 2%, 4%, and 6%, respectively. To eliminate the factor of noise interference, the SNR for the signal dataset was set to 20 dB. The experimental results are shown in Table 3. For all the ratios, our proposed method outperformed the existing



(a) ROC curves for each of the six devices

(b) Precision-recall curves for each of the six devices

Figure 12: ROC and precision-recall curves for each of the six devices.

methods, which further proves that the proposed method is adaptable and advanced in noncooperative scenarios.

Classification accuracy vs. communication propagation channel: we evaluated the classification accuracy of the proposed algorithm in different propagation channels. The collected RF signals were transmitted over an AWGN channel, Rayleigh channel, and Rice channel. Figure 10 shows the classification accuracy of different channels as a function of the SNR. With the Rayleigh and Rice channels, the classification accuracy was lower than with the AWGN channel; this is because the RF fingerprints of the signal emitted by the emitters were not apparent, and it was more difficult to distinguish after being affected by multiplicative noise. The proposed method essentially constructs (virtual) adversarial samples against additive noise, but not multiplicative noise, limiting the adaptiveness of the SEI system in terms of Rayleigh and Rice channels. However, the decline in classification accuracy, which is 5%–10% on average, is not significant. This demonstrates that the proposed approach can still calculate approximate and relatively effective (virtual) adversarial samples for signal samples according to the current network model under interference from multiplicative noise, which can improve the antinoise performance and generalization ability of the network to some extent. The experimental result further demonstrates the excellent classification performance of the proposed method.

Classification accuracy vs. number of emitters: we evaluated the classification accuracy on a different number of emitters. We collected RF signals from up to 14 different individual emitters and evaluated the classification performance of the network on different number of emitters, which varied from 6 to 14 and needed to be classified. The experiment was conducted based on an AWGN channel with SNR = 10 dB.

Figure 11 shows how the classification accuracy changes with the number of emitters. The experimental result indicates that although classification performance deteriorates as the number of emitters increases, the network can classify up to 14 emitters with an accuracy of more than 85%. In general, with the increase in the number of emitters to be identified, the network scale should be increased also, resulting in a higher computation cost. Nevertheless, our method can classify more emitter individuals at a high classification accuracy, maintaining the existing network scale, which benefits from the elaborate training algorithm based on VAT and indicates that the proposed method has good scalability for large emitter populations.

3.3. *Other Evaluation Metrics for Classification Performance.*

Classification accuracy is the ratio of the number of correctly classified samples to the total number of test samples, which can reflect only the overall classification performance. However, it is difficult to determine whether each class of RF signals is correctly classified, particularly when each class is in a minority with respect to the rest of the RF signals, leading to the class imbalance problem. In this case, classification accuracy is not a comprehensive evaluation measure of classification performance.

We used the receiver operating characteristic (ROC) and the precision-recall as metrics to further evaluate the classification performance of the proposed approach. To identify six USRPs, we chose one device as the positive class with a weight of five and the remaining five devices as a single negative class with a weight of one. The six classes of RF signals were collected with SNR of 10 dB, which were used for training and identification. The ROC and precision-recall curves for each of the six devices are shown in Figure 12.

As shown in Figure 12(a), the ROC curve for each device is distributed in the upper left of the figure. This implies that the system achieves a high true-positive rate with a low false-positive rate. Furthermore, precision and recall are two evaluation indices that balance each other. Figure 12(b) shows

Table 4: Mean ROC AUC and mean precision-recall AUC at different SNRs.

SNR	ROC-AUC	Precision-recall AUC
0 dB	0.924	0.811
4 dB	0.939	0.878
8 dB	0.957	0.907
12 dB	0.978	0.928
16 dB	0.981	0.935
20 dB	0.981	0.937

that the precision-recall curve for each device is distributed in the upper right of the figure, indicating that precision occupies a larger proportion. We also calculated the area under the curve (AUC) for each device. The ROC AUC and precision-recall AUC for each device is more than 90%, which further proves the excellent classification performance of the proposed method. Table 4 summarizes the mean ROC AUC and mean precision-recall AUC at SNRs of 0 to 20 dB. As expected, our proposed framework achieves a better mean AUC on higher SNRs and has a drop-off at lower SNRs, which is not significant.

4. Conclusion

To address the shortcomings of traditional SEI based on deep learning, this paper proposes an SEI method based on bispectrum analysis and VAT. First, bispectrum analysis is performed on the RF signals as a way of signal preprocessing. Noting that the emitter signal is susceptible to noise interference and only a small number of labelled training samples are available with many of the samples being unlabelled in a noncooperative communication scenario, we calculated the virtual adversarial samples for both labelled and unlabelled signal samples, using which we then calculated the corresponding loss functions. Loss functions based on the labelled samples were also calculated. Using the two loss functions and the preset harmonic parameters, the objective function of the neural network was calculated. Through iterative tuning, the neural network model corresponding to the minimum loss function value of the verification dataset was obtained as the optimal output. Finally, the neural network model could be used for SEI.

Numerical experiments were conducted to evaluate the performance of the proposed method. First, convergence experiments showed that our approach has stable and fast convergence. Second, we considered four factors that impact the classification accuracy of our method. The classification accuracy vs. SNR experiment showed that our method is significantly robust to noise. The classification accuracy vs. ratio of labelled to unlabelled sample experiment showed that our method can handle weak labelling problems in practical situations. The classification accuracy vs. propagation channel experiment showed that our method can also resist the interference of nonlinear multiplicative noise to RF signals and maintain a relatively high classification accuracy. The classification accuracy vs. number of emitter experiment showed that our method exhibits good scalability for large emitter populations. Moreover, we used two other methods including ROC and precision-recall to further evaluate the classification performance of our method. The AUC for ROC and precision-recall curves were calculated to represent the correct recognition rate of each device, and experimental results demonstrated the excellent classification performance of our method more comprehensively.

Future research will consider the following two aspects. (1) Various emitter devices, in addition to USRP, will be used to collect more types of RF signals to avoid using one type of signal data for experimentation and verify the scalability of the proposed method for RF signals emitted by different types of devices. Furthermore, these devices work in an outdoor environment to obtain more realistic signal data to verify the practicability of our method. (2) Our method is essentially for closed-set identification. The RF signal class to be identified is the same as that in the database used for training. However, our method cannot address anomalous emitters it was not previously trained on, which is common in real-world applications. Therefore, future work will focus on detecting anomalous emitters and classifying known emitters.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 91538201, in part by Taishan Scholar Project of Shandong Province under Grant ts201511020, and in part by project supported by Chinese National Key Laboratory of Science and Technology on Information System Security under Grant 6142111190404.

References

[1] L. Li, H. B. Ji, and L. Jiang, "Quadratic time–frequency analysis and sequential recognition for specific emitter identification," *IET Signal Process*, vol. 5, no. 6, pp. 568–574, 2011.

[2] H. J. Liu, Z. Liu, W. L. Jiang, and Y. Y. Zhou, "Incremental learning approach based on vector neural network for emitter identification," *IET Signal Process*, vol. 4, no. 1, pp. 45–54, 2010.

[3] H. J. Liu, Z. Liu, W. L. Jiang, and Y. Y. Zhou, "Approach based on combination of vector neural networks for emitter identification," *IET Signal Process*, vol. 4, no. 2, pp. 137–148, 2010.

[4] P. Padilla, J. F. Valenzuela-Valdes, and J. L. Padilla, "Radiofrequency identification of wireless devices based on RF fingerprinting," *Electronics letters*, vol. 49, no. 22, pp. 1409-1410, 2013.

[5] G. Lopez-Risueno, J. Grajal, and A. Sanz-Osorio, "Digital channelized receiver based on time-frequency analysis for signal interception," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 879–898, 2005.

[6] Y. P. Zhou, X. Wang, Y. Chen, and Y. Tian, “Specific emitter identification via bispectrum-radon transform and hybrid deep model,” *Mathematical Problems in Engineering*, vol. 2020, no. 2020, Article ID 7646527, 17 pages, 2020.

[7] Y. J. Yuan, Z. T. Huang, W. Hao, and X. Wang, “Specific emitter identification based on Hilbert–Huang transform-based time–frequency–energy distribution features,” *IET Communications*, vol. 8, no. 13, pp. 2404–2412, 2014.

[8] U. Satija, N. Trivedi, G. Biswal, and B. Ramkumar, “Specific emitter identification based on variational mode decomposition and spectral features in single hop and relaying scenarios,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 581–591, 2019.

[9] S. Tang, L. Chen, K. H. J. Xia, L. Fan, and A. Nallanathan, “Computational intelligence and deep learning for next-generation edge-enabled industrial IoT,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 114–125, 2021.

[10] L. He and K. He, “Efficient memory-bounded optimal detection for GSM-MIMO systems. Accepted by,” *IEEE Transactions on Communications*, pp. 1–12, 2021.

[11] K. He, L. F. Le He, Y. Deng, G. K. Karagiannidis, and A. Nallanathan, “Learning-based signal detection for MIMO systems with unknown noise statistics,” *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3025–3038, 2021.

[12] M. Chen, T. Wang, S. Zhang, and A. Liu, “Deep reinforcement learning for computation offloading in mobile edge computing environment,” *Computer Communications*, vol. 175, pp. 1–12, 2021.

[13] X. Lai, L. Fan, and A. Nallanathan, “Outdated access point selection for mobile edge computing with cochannel interference,” *IEEE Transactions on Vehicular Technology*, 2021.

[14] L. Y. Chen, R. Zhao, K. He, Z. Zhao, and L. Fan, “Intelligent ubiquitous computing for future UAV-enabled MEC network systems,” *Cluster Computing*, vol. 2021, no. 25, pp. 1–10, 2021.

[15] L. J. Wong, W. C. Headley, and A. J. Michaels, “Specific emitter identification using convolutional neural network-based IQ imbalance estimators,” *IEEE Access*, vol. 7, pp. 33544–33555, 2019.

[16] B. X. He and F. G. Wang, “Cooperative specific emitter identification via multiple distorted receivers,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3791–3806, 2020.

[17] Y. H. Qian, J. Qi, X. Y. Kuai, G. Han, H. Sun, and S. Hong, “Specific emitter identification based on multi-level sparse representation in automatic identification system,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2872–2884, 2021.

[18] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O. A. Dobre, and H. V. Poor, “An efficient specific emitter identification method based on complex-valued neural networks and network compression,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2305–2317, 2021.

[19] M. Du, X. He, X. Cai, and D. Bi, “Balanced neural architecture search and its application in specific emitter identification,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 5051–5065, 2021.

[20] K. Lee, H. M. Yang, and S. Y. Oh, “Adversarial training on joint energy based model for robust classification and out-of-distribution detection,” in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pp. 17–21, Busan, Korea, 2020.

[21] Z. J. Wang, W. Zhang, X. W. Rong, and Y. B. Li, “Salient object detection with adversarial training,” *IET Image Processing*, vol. 13, no. 14, pp. 2859–2865, 2019.

[22] D. Zhang and W. B. Wu, “Asymptotic theory for estimators of high-order statistics of stationary processes,” *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 4907–4922, 2018.

[23] L. Saidi, J. Ben, and F. Fnaiech, “Application of higher order spectral features and support vector machines for bearing faults classification,” *ISA Transactions*, vol. 54, pp. 193–206, 2015.

[24] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[25] L. Han, J. Z. Sun, and W. Zhang, “Convolutional neural network for convective storm nowcasting using 3-D doppler weather radar data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 2, pp. 1487–1495, 2020.

[26] P. Giampaolo, M. Ludovico, and Z. Pietro, “Segmentation and semantic labelling of RGBD data with convolutional neural networks and surface fitting,” *IET Computer Vision*, vol. 11, no. 8, pp. 633–642, 2017.

[27] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, “Adversarial examples in RF deep learning: detection of the attack and its physical robustness,” in *Proceedings 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–6, Shaw Centre, Ottawa, Canada, 2019.

[28] M. Sadeghi and E. G. Larsson, “Adversarial attacks on deep-learning based radio signal classification,” *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2019.

[29] T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.

[30] H.-K. Poon, W.-S. Yap, Y.-K. Tee, W.-K. Lee, and B.-M. Goi, “Hierarchical gated recurrent neural network with adversarial and virtual adversarial training on text classification,” *Neural Networks*, vol. 119, pp. 299–312, 2019.

[31] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and J. Ian, “Realistic evaluation of deep semi-supervised learning algorithms,” in *Proceedings 32nd Conference on Neural Information Processing Systems (NIPS)*, pp. 1–19, Montréal, Canada, December 2018.

[32] T. Miyato, A. M. Dai, and I. Goodfellow, “Virtual adversarial training for semi-supervised text classification,” in *Proceedings 2017 International Conference on Learning Representations(ICLR)*, pp. 1–11, Toulon, France, 2017.

[33] T. Miyato, S. I. Maeda, M. Koyama, K. Nakae, and S. Ishii, “Distributional smoothing with virtual adversarial training,” in *ICLR 2016: 4th International Conference on Learning Representations, ICLR 2016*, pp. 1–12, San Juan, Puerto Rico, 2016.

[34] L. V. Der Maaten and G. E. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[35] N. Yang, B. Zhang, G. Ding et al., “Specific emitter identification with limited samples: a model-agnostic meta-learning approach,” *IEEE Communications Letters*, 2021.