applied sciences

MDPI

Article

Unsupervised Learning-Based Spectrum Sensing Algorithm with Defending Adversarial Attacks

Xinyu Li [1], Shaogang Dai [1] and Zhijin Zhao [1,2,*]

[1] School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310020, China; 212080139@hdu.edu.cn (X.L.)
[2] State Key Lab of Information Control Technology in Communication System of No. 36, Jiaxing 314000, China
* Correspondence: zhaozj03@hdu.edu.cn

Abstract: Although the spectrum sensing algorithms based on deep learning have achieved remarkable detection performance, the sensing performance is easily affected by adversarial attacks due to the fragility of neural networks. Even slight adversarial perturbations lead to a sharp deterioration of the model detection performance. To enhance the defense capability of the spectrum sensing model against such attacks, an unsupervised learning-based spectrum sensing algorithm with defending adversarial attacks (USDAA) is proposed, which is divided into two stages: adversarial pre-training and fine-tuning. In the adversarial pre-training stage, encoders are used to extract the features of adversarial samples and clean samples, respectively, and then decoders are used to reconstruct the samples, and comparison loss and reconstruction loss are designed to optimize the network parameters. It can reduce the dependence of model training on labeled samples and improve the robustness of the model to attack perturbations. In the fine-tuning stage, a small number of adversarial samples are used to fine-tune the pre-trained encoder and classification layer to obtain the spectrum sensing defense model. The experimental results show that the USDAA algorithm is better than the denoising autoencoder and distillation defense algorithm (DAED) against FGSM and PGD adversarial attacks. The number of labeled samples used in USDAA is only 11% of the DAED. When the false alarm probability is 0.1 and the SNR is -10 dB, the detection probability of the USDAA algorithm for the fast gradient sign method (FGSM) and the projected gradient descent (PGD) attack samples with random perturbations is above 88%, while the detection probability of the DAED algorithm for both attack samples is lower than 69%. Additionally, the USDAA algorithm has better robustness to attack with unknown perturbations.

Keywords: spectrum sensing; security; adversarial attacks and defense; unsupervised learning; contrast loss; reconstruction loss

check for updates

Citation: Li, X.; Dai, S.; Zhao, Z. Unsupervised Learning-Based Spectrum Sensing Algorithm with Defending Adversarial Attacks. *Appl. Sci.* **2023**, *13*, 9101. https://doi.org/10.3390/app13169101

Academic Editor: Gianluigi Ferrari

Received: 19 July 2023
Revised: 6 August 2023
Accepted: 8 August 2023
Published: 9 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The exponential growth of wireless services leads to an increasing shortage of spectrum resources. Cognitive radio (CR) [1] allows secondary users (SU) to use idle authorized spectrum, alleviating the distress situation of a short supply of spectrum resources. Spectrum sensing (SS) [2] plays a crucial role in CR, and extensive research has been conducted in this area. Classical spectrum sensing algorithms use detection statistics designed according to the difference between signal and noise statistical features to determine whether the authorized spectrum is occupied, but the detection performance depends on artificial feature parameters designed and prior information about primary user (PU). With the powerful feature extraction capability of neural networks, deep learning [3] can automatically extract the features of signal samples received in complex communication scenarios without any prior information. The detection performance of the trained spectrum sensing networks are better than that of the classical spectrum sensing algorithms. At present, the spectrum sensing algorithms based on deep learning have achieved remarkable results [4–7]. In [4],

the residual network was used to extract the sample features of the primary user signal and noise, and the obtained sensing model has good generalization performance for the static primary user signal. The residual network and attention mechanism network were used to extract sample features and realize the spectrum sensing of dynamic primary users [5]. However, supervised training was adopted in [4,5], which relied too much on labeled samples. To overcome the dependence of model training on sample labels, a semi-supervised algorithm was used to label samples with false labels and expanded the original data set to train the spectrum sensing model [6]. To reduce the dependence of spectrum sensing model training on labeled samples, a comparative learning framework was also proposed and the residual network was designed to extract rich features from unlabeled samples [7]. Although the spectrum sensing algorithms based on deep learning have presented a series of achievements in different scenarios, the performance of the spectrum sensing models are susceptible to malicious attacks [8–11]. Enhancing the defense capability of the spectrum sensing model against attacks and improving the robustness and security of the models have become important issues to be solved urgently. As a result, the issue of spectrum sensing security has attracted the attention of researchers. Primary user emulation attack ($PUEA$) and spectrum sensing data falsification ($SSDF$) are two classic attack modes that attack traditional spectrum sensing algorithms, and the defense algorithms against these two attacks have been widely studied. However, there are few researches on the defense of primary user adversarial attack ($PUAA$), which attacks spectrum sensing algorithms based on deep learning. Therefore, the unsupervised learning-based spectrum sensing algorithm with defending adversarial attacks (USDAA) that defends white box $PUAA$ is proposed in this paper. The contributions of this paper are as follows:

- An unsupervised learning spectrum sensing algorithm is proposed to effectively defend the fast gradient sign method (FGSM) and the projected gradient descent (PGD) attacks. The unsupervised adversarial pre-training is carried out by using clean samples and adversarial samples with different perturbation values. The defense model is fine-tuned by a small number of adversarial samples.
- In the adversarial pre-training stage, the encoders extract the features of clean samples and adversarial samples, respectively; the decoders reconstruct clean samples and adversarial samples according to the feature latent representations, and the contrast loss and reconstruction loss are used to optimize the network parameters of the model. It reduces the dependence of model training on labeled samples and improves the robustness of the model to attack perturbations.
- A large number of simulation experiments are conducted to evaluate the performance of the algorithms. Experimental results show that the USDAA algorithm has a better defense effect against PGD and FGSM white box attacks than the DAED algorithm [12]. The number of labeled samples used in USDAA is only 11% of the DAED algorithm. When the false alarm probability is 0.1 and the signal-to-noise ratio (SNR) is greater than -8 dB, the detection probability of the USDAA algorithm for the FGSM and PGD attack samples can reach more than 96%, which is more than 10% higher than that of the DAED algorithm. And the USDAA algorithm is more robust than the DAED algorithm; when the perturbation value increases from 0 to 0.14, the detection probability of the USDAA algorithm for attack samples decreases by 10% at most, while the detection probability of the DAED algorithm for attack samples decreases by 29% at most.

2. Related Work

$PUEA$ [13] and $SSDF$ [14] attack traditional spectrum sensing algorithms, while $PUAA$ attacks the spectrum sensing models based on deep learning [15]. As shown in Table 1, some effective defense algorithms have been proposed according to the characteristics of different attacks. $PUEA$ is a typical spectrum sensing malicious attack mode used to prevent other secondary users from employing the authorized spectrum for proper or other purposes. The attacker sends a signal that is similar to that of the primary user

to increase the false alarm probability of the secondary users, making the SU mistakenly think that the current frequency band is occupied by the PU, thus losing the opportunity to use the spectrum hole, resulting in the waste of spectrum resources. In addition to $PUEA$, the $SSDF$ attack also deteriorates the spectrum sensing performance, and malicious nodes send forged data to adjacent secondary users or fusion centers, causing incorrect decisions. At present, the defense algorithms against $PUEA$ [16–19] and $SSDF$ [20–23] have been widely studied as shown in Table 1. Because of the vulnerability of neural networks, $PUAA$ only needs to add small perturbations to the input samples of the spectrum sensing model, which can cause the sensing model fail to accurately detect the PU signal. $PUAA$ can be divided into white box $PUAA$ and black box $PUAA$ according to the different ways of obtaining adversarial samples [24]. The attacker of white box $PUAA$ knows all the information such as the internal structure and parameters of the target model, and can directly use the target model to obtain adversarial samples. However, the attacker of black box $PUAA$ knows nothing about the structure and parameters of the target model, and can only obtain adversarial samples by interacting with the target model or training the proxy model, requiring a large number of experiments to find effective attack strategies. Therefore, the attack effect of white box $PUAA$ is better than that of black box $PUAA$. $PUAA$ uses a neural network to add perturbations to clean samples to achieve the purpose of attack, so the defense algorithms targeting $PUEA$ and $SSDF$ cannot effectively defend $PUAA$. Several defense algorithms were designed against black box attacks [25–27]. The attacker was tricked into receiving the wrong training data set by SU deliberately carrying out a small number of false transmissions [25], but this defense approach requires SU transmitters to balance performance and defense effectiveness. In [26], three black box $PUAA$s were designed to attack the spectrum sensing model based on residual network, and the vulnerability of spectrum sensing network was verified; convolutional neural network (CNN) and long short-term memory (LSTM) algorithms were used to design a Deep Filter (DF) encoder to extract fusion features of clean samples and adversarial samples, and the decoder reconstructed clean samples according to fusion features. The proposed algorithm can effectively defend black box adversarial attacks without affecting the detection performance of clean samples. Additionally, a proxy model was used to obtain adversarial samples, and a wireless receiver was designed to combine time-domain and frequency-domain characteristics [27] which can effectively defend black box adversarial attacks. Currently, most research focuses on the defense algorithms of spectrum sensing against black box $PUAA$, but there is a limited number of studies on defense algorithms against white box $PUAA$. FGSM [28] and PGD [29] are two classic white box $PUAA$ methods. To defend these two attacks, the defense algorithm was designed based on a denoising autoencoder and distillation (DAED) [12]. CNN and transposed convolutional neural networks were used to design the denoising autoencoder to filter out the perturbations, and clean samples were obtained in this algorithm. The obtained clean samples and the defense distillation algorithm were used to train the residual network to obtain the spectrum sensing model. However, the robustness and generalization of this model to attack perturbations are poor and algorithm performance relies on a large number of labeled samples. To improve the robustness of the defense model for the attack perturbations and reduce the dependence of the model training on labeled samples, we propose the unsupervised spectrum sensing algorithm with defending adversarial attacks (USDAA).The USDAA algorithm uses less labeled samples, and the algorithm has a better defense effect against PGD and FGSM white box $PUAA$ than the DAED algorithm [12]. In addition, the robustness of the USDAA algorithm is better than that of the DAED algorithm.

Table 1. References to defend different attacks.

References	Attacks	Types
[16–19]	PUEA	-
[20–23]	SSDF	-
[25–27]	PUAA	Black Box
[12]	PUAA	white Box

3. Problem Description

Spectrum sensing can be modeled as a binary hypothesis testing problem as shown in Equation (1):

$$y(n) = \begin{cases} u(n) & H_0 \\ x(n) + u(n) & H_1 \end{cases} \quad n = 0, 1, 2, \ldots, \mathrm{N} - 1, \tag{1}$$

where H_0 represents the assumption that there is no primary user signal on the frequency band; H_1 represents the assumption that the frequency band is occupied by a primary user; $y(n)$ represents the received baseband signal; $u(n)$ is independent identical distributed Gaussian white noise with mean 0 and variance σ_u^2; $x(n) = h(n)s(n)$ is the primary user signal, and $x(n)$ and $u(n)$ are independent of each other, where $h(n)$ represents the channel gain between the primary user and the secondary user and N indicates the signal length. Two indexes to measure the spectrum sensing performance of the algorithm are detection probability P_d and false alarm probability P_f , which are defined as shown in Equation (2).

$$P_{\mathrm{f}} = P(H_1|H_0) \qquad \mathrm{P_d} = \mathrm{P}(\mathrm{H_1}|\mathrm{H_1}). \tag{2}$$

Single-node primary user adversarial attack is to add a perturbation to the received clean signal $y(n)$, making the spectrum sensing classifier unable to classify correctly. Its mathematical expression is shown in Equation (3),

$$F_\beta(y(n)) \neq F_\beta(y(n) + \delta(n)), \tag{3}$$

where F_β represents the attacked spectrum sensing classifier, that is, the target model; $\delta(n)$ represents the perturbation. The USDAA algorithm proposed in this paper improves the robustness of the model against adversarial samples through unsupervised adversarial pre-training, and the mathematical expression of the defense algorithm is shown in Equation (4), where R_θ represents the defense model trained by the USDAA algorithm.

$$R_\theta(y(n)) = \mathrm{R}_\theta(y(n) + \delta(n)). \tag{4}$$

4. Implementation of USDAA Defense Algorithm

The framework of the USDAA algorithm is shown in Figure 1. The USDAA algorithm consists of two stages: adversarial pre-training and fine-tuning.

In the adversarial pre-training stage, two encoders extract the features of clean samples and adversarial samples, respectively, and the two decoders reconstruct the clean samples and adversarial samples according to the potential representation of the features $Z_{\mathrm{i}}(i = 1, 2)$, respectively. After the adversarial pre-training, the parameters of the encoder module of the feature extractor are saved as ξ. In the fine-tuning stage, the feature dimension is reduced to the classification dimension, a linear layer is added to the encoder module obtained by pre-training, the model parameters are trained with only a small number of adversarial samples, and the features and labels of the samples are matched. The fine-tuned model is the final defense model R_θ.

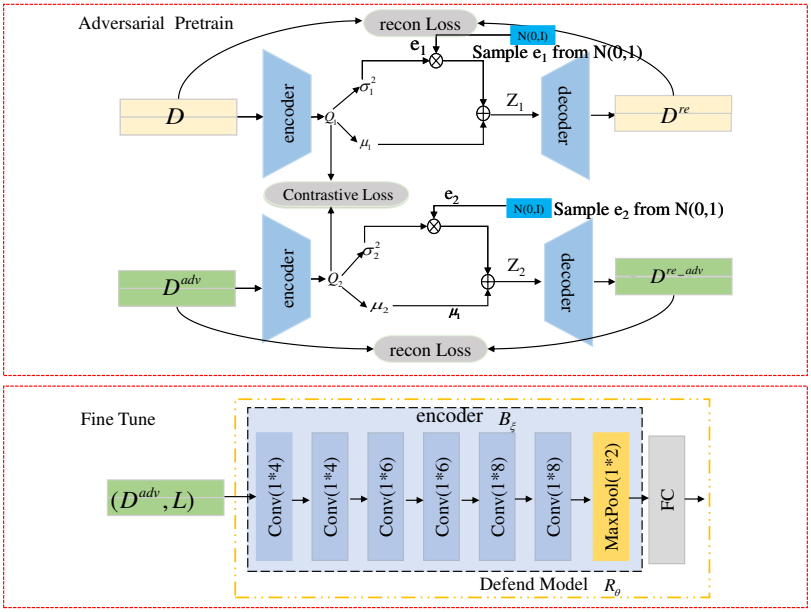


Figure 1. USDAA algorithm framework.

4.1. Adversarial Pre-Training

Adversarial pre-training uses two encoders and two decoders with the same structure and shared parameters. The encoder is composed of six convolution layers and one maximum pooling layer, and each convolution layer is followed by a batch normalization layer (BN) and a nonlinear activation layer (Relu). The decoder is composed of six deconvolution networks and a maximum pooling layer. Similarly, behind each deconvolution layer, there are a BN layer and a Relu layer. The deconvolution layer is the inverse operation of the convolutional layer, which inversely maps the feature potential representation $Z_i(i = 1, 2)$ to the original signal space. A feature extractor encoder is obtained by the adversarial pre-training, which can extract rich features of adversarial samples and clean samples.

To enhance the effectiveness of adversarial pre-training and achieve the goal of defending the primary user adversarial attacks in spectrum sensing, the loss function in the adversarial pre-training shown in Equation (5) is designed:

$$loss = \mathrm{loss_c} + \mathrm{loss_{kl}} + \mathrm{loss_{re}}, \tag{5}$$

where loss_c represents contrast loss, loss_{kl} represents KL divergence and losss_{re} represents sample reconstruction loss. The labels of adversarial samples and clean samples are the same, and the adversarial samples and clean samples are taken as a pair of positive samples. Encoders extract the clean sample features Q_1 and the adversarial sample features Q_2, respectively, and the Euclidean distance between Q_1 and Q_2 is calculated using Equation (6) as the contrast loss value $loss_c$,

$$\mathrm{loss_c} = \| Q_1 - Q_2 \|_2 = \Big(\sum_{\mathrm{i}=1}^{\mathrm{v}} |q_{1i} - q_{2i}|^2 \Big)^{1/2}, \tag{6}$$

where the characteristic dimension of $Q_i(i = 1, 2)$ is $v \times 1$. The contrast loss between adversarial samples and clean samples is calculated, guiding the model to gather the same sample features together according to the extracted features, improving the classification and clustering ability of the encoder, and enhancing the robustness of the model to the adversarial samples and clean samples. According to Equation (7), the potential representation of features Z_1 and Z_2 is obtained by using the reset parameters [30]

$$Z_1 = \mu_1 + e_1 * \sigma_1^2 \qquad Z_2 = \mu_2 + e_2 * \sigma_2^2, \tag{7}$$

where e_1 and e_2 are samples from a matrix following the $N(0,1)$ distribution, and $\mu_i, \sigma^2{}_i (i = 1,2)$ are the mean and variance from the encoder output. In order to enhance the diversity of the reconstructed samples D^{re} and D^{re_adv}, forcing the potential feature representation Z_1 and Z_2 obtained from the encoder to approximate a normal distribution, the KL divergence [31] $loss_{kl}$ between $N(\mu_1, \sigma_1^2 I)$, $N(\mu_2, \sigma_2^2 I)$ and the standard normal distribution $N(0, I)$ are calculated, respectively, and their sum is shown in Equation (8).

$$loss_{kl} = KL\{N(\mu_1, \sigma_1^2 I)||N(0, I)\} + KL\{N(\mu_2, \sigma_2^2 I)||N(0, I)\}. \tag{8}$$

To make the model display better generalization performance for adversarial samples with different perturbation values, the reconstruction loss is designed as shown in Equation (9), where D^{re} and D^{re_adv} represent the clean samples and adversarial samples reconstructed by the decoder from Z_1 and Z_2, respectively, and t represents the sample dimension.

$$\begin{aligned} loss_{re} &= \|D - D^{re}\|_2 + \left\|D^{adv} - D^{re_adv}\right\|_2 \\ &= (\sum_{i=1}^{t} \left|d_i - \mathrm{d}_i^{re}\right|^2)^{1/2} + (\sum_{i=1}^{t} \left|d_i^{adv} - \mathrm{d}_i^{re_adv}\right|^2)^{1/2} \end{aligned}. \tag{9}$$

Here, d_i, d_i^{re}, d_i^{adv} and $\mathrm{d}_i^{re_adv}$ are the ith elements of D, D^{re}, D^{adv} and D^{re_adv}, respectively.

Finally, Equation (10) is used to calculate the batch loss gradient, where I is the batch size and Equation (11) is used to complete the network parameter update.

$$\nabla\xi \leftarrow \frac{1}{I}\sum_{k=1}^{I} \frac{\partial loss}{\partial \xi}, \tag{10}$$

$$\xi \leftarrow optimizer(\xi, \nabla\xi). \tag{11}$$

4.2. Fine Tuning

The pre-trained encoder is used as a feature extractor, and a classification layer is added to the encoder to form defense model R_θ. To prevent fine-tuning from destroying pre-training parameters, only a small number of corresponding adversarial samples and labels are used to fine-tune. Two fine-tuning methods are used in this paper. The first method is to fine-tune the encoder network parameters ξ obtained by pre-training and the classification layer, and the defense model is denoted as USDAA. The second way is to only fine-tune the classification layer and freeze ξ, and the fine-tuned defense model is denoted as USDAA-F.

Both fine-tuning methods use the cross-entropy loss function shown in Equation (12) to calculate the loss value of each batch of the adversarial samples,

$$\mathrm{loss}_{FT} = -\frac{1}{I}\sum_{k=1}^{I} [L_k \log prob(D_k^{adv}, H_1) + (1 - L_k) \log prob(D_k^{adv}, H_0)], \tag{12}$$

where L denotes the true label of the samples, $L = 1$ denotes the presence of PU on the frequency band, $L = 0$ denotes the presence of only noise on the frequency band, respectively, $prob(D, H_1), prob(D, H_0)$ denote the probability of the presence of PU on the detection channel and the probability of only noise on the channel, and $prob(D, H_1) + prob(D, H_0) = 1$. The gradient of the loss function per lot is shown in Equation (13),

$$\nabla\theta \leftarrow \frac{1}{I}\sum_{k=1}^{I} \frac{\partial \mathrm{loss}_{FT}}{\partial \theta}, \tag{13}$$

and the network parameters are updated by Equation (14).

$$\theta \leftarrow optimizer(\theta, \nabla\theta). \tag{14}$$

4.3. Implementation of Primary User Adversarial Attacks and Defense

The process of the primary user adversarial attacks and defenses is shown in Figure 2. The target model F_β is trained using clean samples D and labels, and its network structure is identical to the encoder. The generation process of adversarial samples of two kinds of white box primary user adversarial attacks is shown in the left part of the network in Figure 2. According to FGSM and PGD, two kinds of white box attack algorithms, clean samples D are input into the target model F_β to obtain adversarial samples D^{adv}. The right side of Figure 2 shows the defense process, and the defense network R_θ is obtained by the methods in Sections 4.1 and 4.2. When the adversarial samples D^{adv} are input to the defense model R_θ, it can correctly sense whether there is a primary user signal.

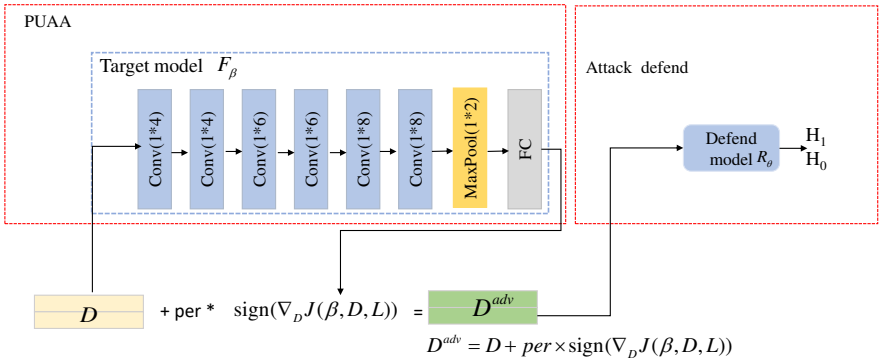


Figure 2. PUAA and defense.

4.3.1. FGSM

FGSM is a classic gradient-based attack algorithm whose principle is to add perturbations Δ_D along the gradient direction of the network model changes and obtain the adversarial samples $D^{adv} = D + \Delta_D$. The specific process is shown in Figure 2. The clean samples D are input into the trained target network F_β, the gradient direction signs $(\nabla_D J(\beta, D, L))$ of each sample are obtained by the sign symbol function, and then Δ_D= per $\times$ sign $(\nabla_D J(\beta, D, L))$, where per is the perturbation value, β is the target model parameter, $J(\beta, D, L)$ represents the loss function used for training, and L represents the sample label. The adversarial samples D^{adv} are obtained by Equation (15).

$$D^{adv} = \mathrm{D} + \mathrm{per} \times \mathrm{sign}(\nabla_D J(\beta, D, L)). \tag{15}$$

It can be seen that only one gradient update is needed to obtain a single FGSM adversarial sample.

4.3.2. PGD

Projected gradient descent PGD is an iterative attack algorithm which adds some small random perturbations each time in the iteration process and limits the perturbations to a certain range during the iteration process whose attack iteration mode is shown in Equation (16).

$$\begin{aligned} D_0^{adv} &= D + U(-\gamma, \gamma), \\ D_{\mathrm{i}+1}^{adv} &= \prod_{D,\gamma} \left\{ \mathrm{D}_i^{\mathrm{adv}} + \mathrm{per} \times \mathrm{sign}(\nabla_D J(\beta, \mathrm{D}_i^{\mathrm{adv}}, L) \right\} \end{aligned}, \tag{16}$$

where $U(-\gamma, \gamma)$ represents the random perturbation generated in between $[-\gamma, \gamma]$, each iteration adds a random perturbation $U(-\gamma, \gamma)$ to the sample; $\prod_{D,\gamma}$ indicates that the perturbation is confined within the sphere range; D_{i}^{adv}represent the adversarial samples obtained by iteration i times; per $\times$ sign$(\nabla_D J(\beta, D_i^{\mathrm{adv}}, L)$ is obtained by the same way as FGSM, except that D_{i}^{adv} are used instead of D.

To prevent the amplitude of the adversarial samples D^{adv} from being too large or too small, Equation (17) is used to limit the amplitude A^{adv} of the adversarial samples to the range of the clean samples,

$$A^{adv}=Clip(A^{adv}, A_{\min,} A_{\max}), \tag{17}$$

where A_{max} and A_{min} represent the largest and smallest values of the clean samples, respectively.

4.3.3. Adversarial Attack Defense

By adversarial pre-training and fine-tuning, the USDAA defense models are obtained to defend two types of white box adversarial attacks: FGSM and PGD. The decision threshold λ is determined by the given false alarm probability. The detection criteria of the defense model against adversarial samples D^{adv} are shown in Equation (18), where $prob(D^{adv}, H_1)$ is the output of the defense model R_θ, indicating the probability of D^{adv} being judged H_1; thus, the defense against white box attack is realized.

$$\begin{matrix} H_1 : prob(D^{adv}, H_1) \geq \lambda \\ H_0 : prob(D^{adv}, H_1) < \lambda \end{matrix} . \tag{18}$$

In summary, the USDAA algorithm is shown in Algorithm 1.

Algorithm 1 USDAA Algorithm.

Require: adversarial pre-training sample set D and D^{adv}, adversarial sample set (D^{adv}, L) for fine-tuning, number of rounds k_0 for pre-training and number of rounds k_1 for fine-tuning.

Ensure: the parameters of feature extraction network B_ξ and spectrum sensing classifier R_θ are optimal;

1: **for** epoch = 1 **to** k_0 **do**
2: D and D^{adv} are input into two encoders, respectively, intermediate features $Q_i(i = 1, 2)$ are extracted, and the potential representation of features $Z_i(i = 1, 2)$ are obtained according to Equation (7);
3: The hidden variable $Z_i(i = 1, 2)$ are fed to the decoders, which reconstruct D and D^{adv} according to potential feature representation $Z_i(i = 1, 2)$;
4: The contrast loss, KL divergence and reconstruction loss are calculated by using Equations (6), (8) and (9), the total loss is obtained by using Equation (5);
5: The parameters of encoder and decoder are optimized according to Equations (10) and (11) and the parameters ξ of encoder are saved.
6: **end for**
7: The parameters of the encoder B_ξ are loaded, which is obtained by adversarial pre-training as the feature extractor of the defense model;
8: After the feature extractor, a classification layer is added to obtain complete defense model.
9: **for** epoch = 1 **to** k_1 **do**
10: The fine-tuning samples (D^{adv}, L) are input into the defense model;
11: The parameters θ of defense model are fine-tuned by using Equations (12)–(14).
12: **end for**
13: Return a trained defense model R_θ.

5. Experimental Results

The defense performance of the USDAA algorithm, USDAA-F algorithm and DAED algorithm in [12] is analyzed by simulation. The defense models of USDAA and USDAA-F are obtained by unsupervised pre-training of adversarial samples and clean samples, and fine-tuned by adversarial samples. The adversarial samples with perturbations (*per*) range per $\in [0.06, 0.1]$ with a step size of 0.01 are used for pre-training and fine-tuning, and PGD attacks 10 times iteratively, and the value of γ is 0.007.

The algorithm hyperparameter settings are shown in Table 2, the batch size of the three algorithms is 64, the learning rate is set to 0.0001, and the Adam optimizer is used to train the network. The USDAA algorithm and the USDAA-F algorithm have 30 rounds of adversarial pre-training and 10 rounds of fine-tuning, and the hyperparameters, as well as feature extraction network structure used in DAED algorithm, are the same as those of USDAA. The USDAA algorithm uses a few labeled samples in the fine-tuning stage and the DAED algorithm uses large quantities of labeled samples to train the sensing classifier.

Table 2. The setting of the hyperparameters.

Parameters	Values
batch size	64
optimizer	Adam
number of adversarial pre-training round	30
fine-tuning round	10
learning rate	0.0001
per	0.06–0.1
PGD iteration attack times	10

5.1. Data Sets and Simulation Environment

Four primary user signals including BPSK, QPSK, 64QAM and 16QAM are obtained from RML [32]. The signal length is 512 and the background noise is additive white Gaussian noise. In the training set, the SNR of signal samples ranges from -20 dB to 20 dB with an interval of 2 dB. Each signal has 250 samples per dB, totaling 21,000 signal samples, and 21,000 noise samples. In fine-tuning data set and the test data, the SNR of signal samples ranges from -20 dB to 10 dB with an interval of 2 dB. A total of 30 signal samples per dB of each signal and 1920 noise samples are generated as the fine-tuning data set. A total of 150 samples of each signal per dB are generated as the test set.

The MATLAB software was used to obtain datasets and PyCharm was used to train the defense models. The experimental environment and computer configuration are shown in Table 3.

Table 3. Experimental environment and computer configuration.

Experimental Environment	Environment Configuration
Operating System	Win 11
CPU	AMD Ryzen 7 5800 H with Radeon Graphics
GPU	NVIDIA GeForce RTX 3060
Memory	32 GB
Programming Languages	Python3.9
Deep Learning Framework	PyTorch1.12.0

5.2. Algorithm Performance Analysis

5.2.1. Influence of the Perturbation Attacks on the Target Model F_β

The perturbation value of FGSM and PGD white box attacks is set per $\in [0, 0.1]$ with an interval of 0.02, and the number of iterations of PGD attacks is 10. When the false alarm probability is 0.05 and 0.1, the relationship curves between the perturbation values and the detection probability of the target model F_β trained only with clean samples for the two adversarial samples with the SNR of -12 dB, -10 dB and -8 dB are shown in Figure 3 and Figure 4, respectively.

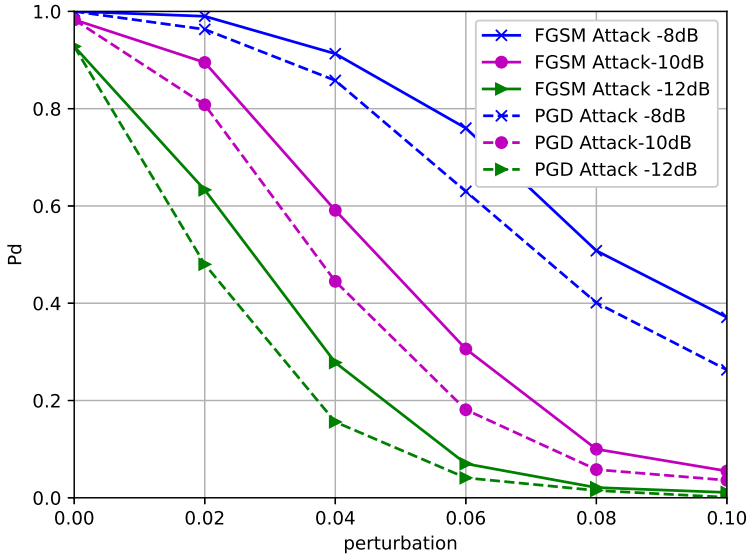


Figure 3. P_f is 0.05, the detection probability of the target model.

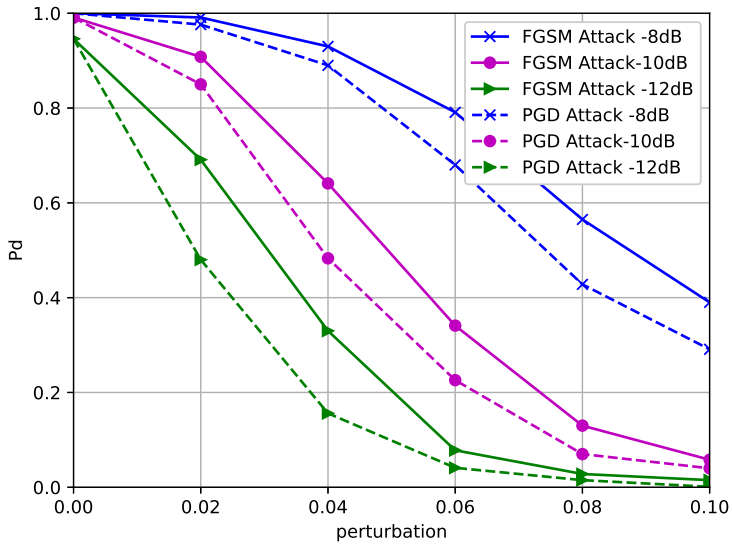


Figure 4. P_f is 0.1, the detection probability of the target model.

The folliwing can be seen from Figures 3 and 4:

(1) When the attack perturbations are 0, that is, the spectrum sensing network is not attacked, the detection probability of the target model F_β for clean samples can reach more than 93%. With the increase in the attack perturbations, the detection probability of the target model F_β for adversarial samples gradually decreases.

(2) The reduction in SNR and false alarm probability decreases the detection probability of target model F_β for adversarial samples, and PGD attack has a greater impact on target model F_β than the FGSM attack. When the SNR is -12 dB, -10 dB and -8 dB, per is 0.06 and 0.1, and the false alarm probability is 0.05 and 0.1, respectively, the detection probabilities of the target model for FGSM and PGD adversarial samples are shown in Table 4.

(3) When the per increases to 0.1, the detection performance of the target model deteriorates sharply, and its robustness is poor. To make the defense model have certain robustness to the adversarial samples with different attack perturbations, three defense models arec trained by adversarial samples with the perturbation value ranging from 0.06 to 0.1 in the following experiments.

Table 4. Detection performance of F_β for adversarial samples.

Attacks	*SNR* (dB)	Per P_f	0.06		0.1
			0.05	0.1	0.1
FGSM Attack	−12 dB		7.4%	7%	1.5%
	−10 dB		30.9%	30.6%	5.8%
	−8 dB		77.2%	76%	39%
PGD Attack	−12 dB		5%	4.1%	1%
	−10 dB		18.6%	18.1%	4%
	−8 dB		63.1%	63%	29.1%

5.2.2. Algorithm Defense Effect Comparison

The false alarm probability is set to 0.1, the perturbation value of FGSM and PGD adversarial samples is per $\in [0.06, 0.1]$, and the detection probabilities of the USDAA algorithm, USDAA-F algorithm and DAED algorithm on FGSM and PGD adversarial samples are shown in Figure 5 and Figure 6, respectively. Without Attack indicates the detection probability for clean samples by the target model F_β.

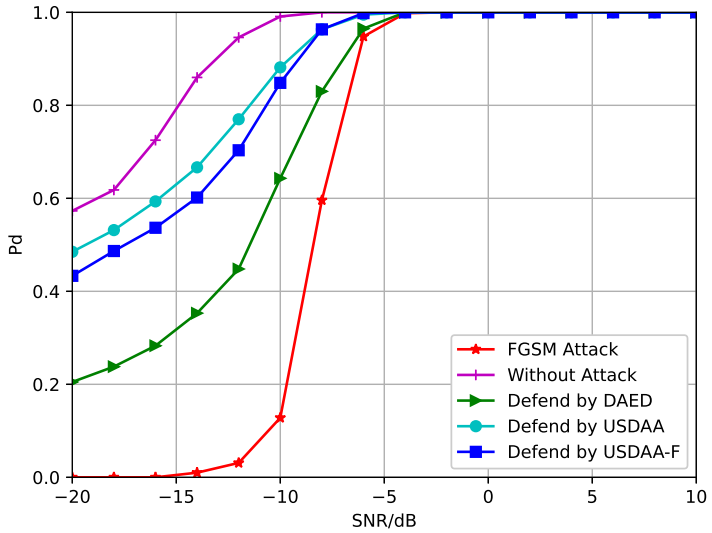


Figure 5. Defense effect of different defense algorithms against FGSM attack.

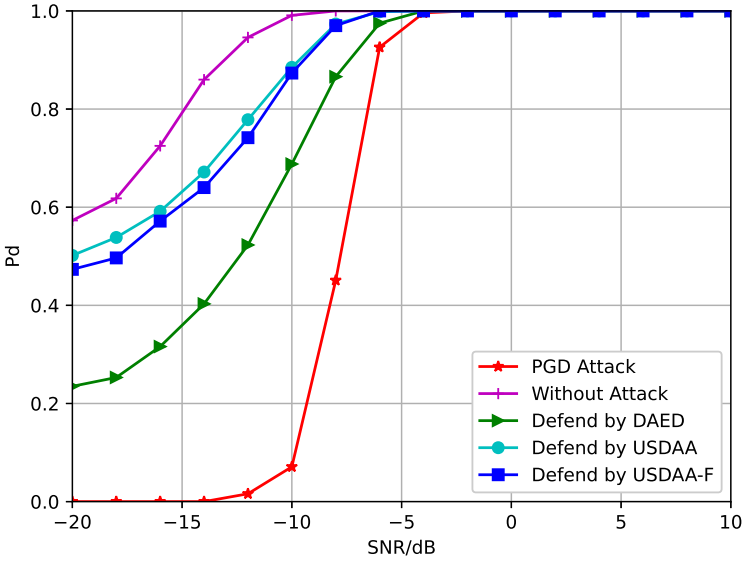


Figure 6. Defense effect of different defense algorithms against PGD attack.

The following can be seen from Figures 5 and 6:

(1) The slight perturbations of the input samples cause the detection performance of the target model to decline sharply. In the case of the absence of attack, when the SNR is -12 dB, the detection probability of the target model F_β for clean samples is 94.6%, but the detection probability of the model for FGSM adversarial samples and PGD adversarial samples is 3.1% and 1.6%, respectively, and the detection effect deteriorates sharply. This is because the target model trained only with clean samples is not robust to the adversarial samples.

(2) The DAED algorithm uses 42000 labeled samples and the USDAA algorithm only uses 3840 labeled samples. Even though the USDAA algorithm uses only 11% of the labeled samples of the DAED algorithm, the defense effects of the USDAA algorithm and the USDAA-F algorithm are better than those of the DAED algorithm.When the false alarm probability is 0.1 and the SNR is -10 dB, the detection probabilities of the USDAA, USDAA-F and DAED for FGSM adversarial samples are 88.1%, 84.8% and 64.3%, respectively; accordingly, the detection probabilities for PGD adversarial samples by the three defense models are 88.5%, 88.7% and 68.8%, respectively. This is because the USDAA algorithm and the USDAA-F algorithm extract the features of clean samples and adversarial samples by using the contrast loss and the reconstruction loss training model in the unsupervised pre-training stage; however, the denoising filter of the DAED algorithm can only remove part of the adversarial perturbations in the adversarial samples, so the defense capability of the DAED algorithm is limited.

(3) The defense effect of the USDAA algorithm is better than that of the USDAA-F algorithm. This is because the USDAA algorithm fine-tunes all network layer parameters, including the parameters of the feature extractor, while USDAA-F fine-tunes only the classification layer parameters.

5.2.3. Robustness Comparison of Algorithms to Attack Perturbations

The false alarm probability is 0.05, and the perturbation values of FGSM and PGD primary user adversarial attack samples are set at 0, 0.06, 0.1, 0.12 and 0.14, respectively, where a perturbation value of zero is equivalent to an absence of attack. The detection probabilities of USDAA and DAED models for FGSM and PGD adversarial attack samples of different perturbation values are shown in Figure 7 and Figure 8, respectively.

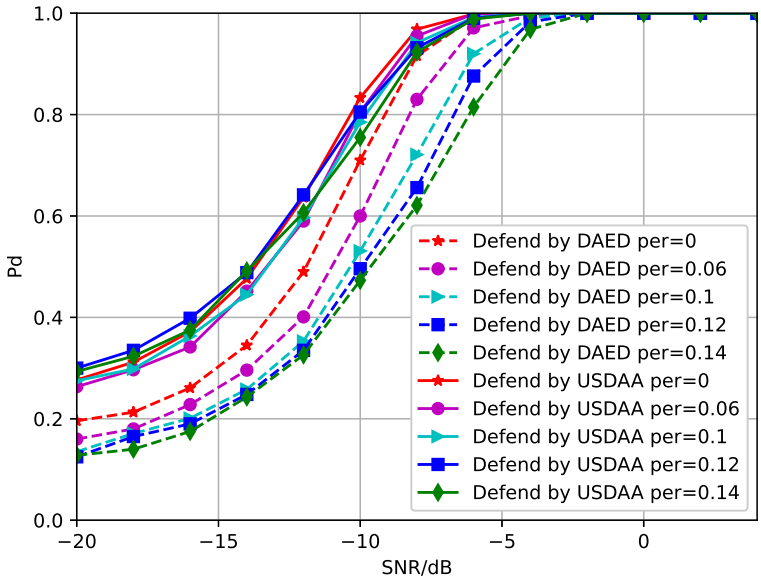


Figure 7. Robustness of FGSM defense model to attack perturbations.

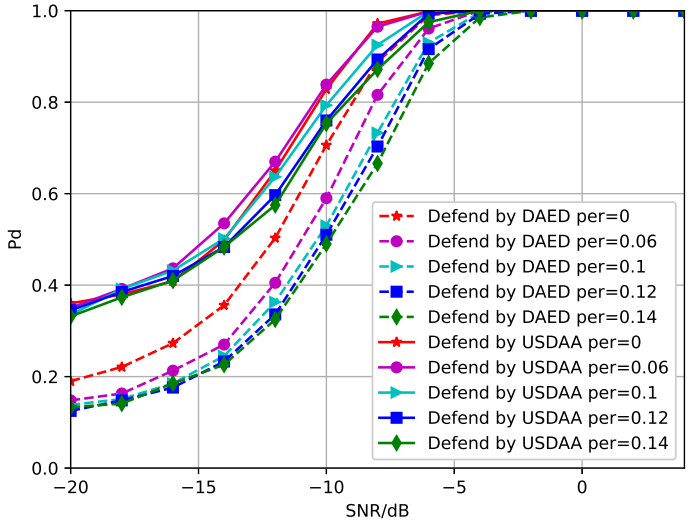


Figure 8. Robustness of PGD defense model to attack perturbations.

According to Figures 7 and 8, the following can be asserted:

(1) For the same kind of attack, with the same perturbation values, the detection probability of the USDAA algorithm is higher than that of the DAED algorithm.

(2) The robustness of the USDAA algorithm to attack perturbation values is better than that of the DAED algorithm. When the false alarm probability is 0.05 and the SNR is greater than -2 dB, the detection probability of the USDAA and DAED models can reach 100% for the FGSM and PGD adversarial attack samples with five perturbation values. When the SNR is lower than -6 dB, the reduction in the detection probability of the USDAA algorithm for two kinds of adversarial attacks is less than 10%; however, the maximum reduction in the detection probability of the DAED algorithm is 29.5%. This is because the USDAA algorithm adopts contrast loss and reconstruction loss to optimize the network in the adversarial pre-training stage and uses KL divergence to constrain the distribution of potential features $Z_i(\mathrm{i} = 1, 2)$ which makes the decoder reconstruction samples more diverse on the basis of the original input samples. Therefore, the defense effect of the USDAA algorithm is not sensitive to the change in perturbations. However, the distillation used in the DAED algorithm affects the generalization of the model and reduces the robustness of the sensing model to adversarial attack samples with different perturbation values.

6. Conclusions

The spectrum sensing algorithms based on deep learning have significantly improved the spectrum sensing performance. However, due to the vulnerability of neural networks, even small primary user adversarial attack perturbations can lead to a sharp decline in algorithm performance. To improve the robustness of the spectrum sensing model for FGSM and PGD attacks, the USDAA algorithm was proposed. In the unsupervised pre-training stage, contrast loss and reconstruction loss are used to extract features from unlabeled samples to obtain the feature extractor. Then, the USDAA defense model is obtained by fine-tuning the feature extractor and classification layer using a small number of labeled samples. Experimental results demonstrate that the USDAA algorithm outperforms the DAED algorithm in terms of defense effectiveness and robustness to attack perturbations. However, PGD and FGSM are both gradient-based attack algorithms, but not all attacks depend on gradients. Therefore, a way to improve the defense capability of spectrum sensing algorithms for non-gradient attacks needs further research.

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

Funding: This research was funded by Research on Intelligent Spread Spectrum Anti-interference Technology in Complex Electromagnetic Environment, U19B2016 National Natural Science Foundation of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding authors upon reasonable request.

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

References

1. Haykin, S.; Setoodeh, P. Cognitive Radio Networks: The Spectrum Supply Chain Paradigm. *IEEE Trans. Cogn. Commun. Netw.* **2015**, *1*, 3–28. [CrossRef]
2. Qu, Z.; Song, Q.; Yin, S. Optimal spectrum sensing in energy harvesting cognitive radio systems. *China Commun.* **2014**, *11*, 83–90. [CrossRef]
3. Peng, Q.; Gilman, A.; Cosman, P.; Milstein, L. Robust deep sensing through transfer learning in cognitive radio. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 38–41. [CrossRef]
4. Zheng, S.; Chen, P.; Qi, H.;Yang, X. Spectrum sensing based on deep learning classification for cognitive radios. *China Commun.* **2020**, *17*, 138–148. [CrossRef]
5. Zhang, P.; Ding, R.; Jiang, T. Spectrum sensing method of deep learning based on attention mechanism. *Wirel. Commun. Tech.* **2022**, *31*, 1–6.
6. Zhang, Y.; Zhao, Z. Limited data spectrum sensing based on semi-supervised deep neural network. *IEEE Access* **2021**, *9*, 166423–166435. [CrossRef]
7. Li, X.; Zhao, Z.; Zhang, Y.; Zheng, S.; Dai, S. Spectrum sensing algorithm based on self-supervised contrast learning. *Electronics* **2023**, *12*, 1317. [CrossRef]
8. Turkyilmaz, Y.; Senturk, A.; Bayrakdar, M.E. Employing machine learning based malicious signal detection for cognitive radio networks. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7457. [CrossRef]
9. Cicioğlu, M; Çalhan, A. Channel aware wireless body area network with cognitive radio technology in disaster cases. *Int. J. Commun. Syst.* **2020**, *33*, e4565.
10. Tian, Q.; Zhang, S.; Mao, S. Adversarial attacks and defenses for digital communication signals identification. *Dig. Commun. Net.* **2022**, *in press*. [CrossRef]
11. Ye, Y.; Chen, Y.; Liu, M. Multiuser adversarial attack on deep learning for OFDM detection. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 2527–2531. [CrossRef]
12. Yang, X.; Li, Z.; Zhang, S. Denoising autoencoder for spectrum sensing adversarial defense model. *Mob. Commun.* **2023**, *47*, 28–36.
13. Chen, R.; Park, J. Ensuring trustworthy spectrum sensing in cognitive radio networks. In Proceedings of the 2006 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks, Reston, VA, USA, 25 September 2006; pp. 110–119.
14. Chen, H.; Zhou, M.; Xie, L.; Li, J. Cooperative spectrum sensing With m-ary quantized data in cognitive radio networks under SSDF attacks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 5244–5257. [CrossRef]
15. Luo, Z.; Zhao, S.; Lu, Z. When attackers meet AI: Learning-empowered attacks in cooperative spectrum sensing. *IEEE Trans. Mob. Comput.* **2020**, *21*, 1892–1908. [CrossRef]
16. Elghamrawy, S. Security in cognitive radio network: Defense against primary user emulation attacks using ge-netic artificial bee colony algorithm. *Future Generation Comput. Syst.* **2020**, *109*, 479–487. [CrossRef]
17. Khaliq, S.; Amjad, M. Defence against pue attacks in ad hoc cognitive radio networks: A mean field game approach. *Telecommun. Syst.* **2019**, *70*, 123–140. [CrossRef]
18. Luo, Z.; Lou, C. Specific primary user sensing for wireless security in ieee 802.22 network. In Proceedings of the 2011 11th International Symposium on Communications and Information Technologies, Hangzhou, China, 12–14 October 2011; pp. 18–22.
19. Yuan, Z.; Niyato, D. Defeating primary user emulation attacks using belief propagation in cognitive radio networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 1850–1860. [CrossRef]
20. Tephilla, S.; Manickam, L. An SETM algorithm for combating SSDF attack in cognitive radio networks. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 9047809. [CrossRef]
21. Parhizgar, N.; Jamshidi, A.; Setoodeh, P. Defense against spectrum sensing data falsification attack in cognitive radio networks using machine earning. In Proceedings of the ICEE, Tehran, Iran, 17–19 May 2022; pp. 974–979.

22. Paul, A.; Mishra, A.K.; Shreevastava, S.; Tiwari, A.K. Deep reinforcement learning based reliable spectrum sensing under SSDF attacks in cognitive radio networks. *J. Netw. Comput. Appl.* **2022**, *205*, 1084–8045. [CrossRef]
23. Kumar, R.; Parmar, A.; Captain, K. Reinforcement learning for performance improvement in cooperative spectrum sensing. *Phys. Commun.* **2023**, *59*, 102112. [CrossRef]
24. Feinman, R.; Shintre, S. Detecting adversarial samples from artifacts. *arXiv* **2017**, arXiv:1703.00410.
25. Sagduyu, Y.; Shi, Y. Adversarial deep learning for over-the-air spectrum poisoning attacks. *IEEE Trans. Mob. Comput.* **2019**, *20*, 306–319. [CrossRef]
26. Zheng, S.; Ye, L.;Wang, X. Primary user adversarial attacks on deep learning-based spectrum sensing and the defense method. *China Commun.* **2021**, *18*, 94–107. [CrossRef]
27. Sahay, R.; Brinton, C.; Love, D. A deep ensemble-based wireless receiver architecture for mitigating adversarial attacks in automatic modulation classification. *IEEE Trans. Commun. Netw.* **2022**, *8*, 71–85. [CrossRef]
28. Goodfellow, J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
29. Madry, A.; Makelov, A.; Schmidt, L. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
30. Kingma, D.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
31. Van Erven, T.; Harremos, P. Rényi divergence and kullback-leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
32. Shea, J.; West, N. Radio machine learning dataset generation with GNU radio. In Proceedings of the GNU Radio Conference, Virtually, 26–30 September 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.