Adversarial Attack and Defense on Deep Learning for Air Transportation Communication Jamming

Mingqian Liu, *Member, IEEE*, Zhenju Zhang, Yunfei Chen, *Senior Member, IEEE*, Jianhua Ge, Nan Zhao, *Senior Member, IEEE*

***Abstract*—Air transportation communication jamming recognition model based on deep learning (DL) can quickly and accurately identify and classify communication jamming, to improve the safety and reliability of air traffic. However, due to the vulnerability of deep learning, the jamming recognition model can be easily attacked by the attacker's carefully designed adversarial examples. Although some defense methods have been proposed, they have strong pertinence to attacks. Thus, new attack methods are needed to improve the defense performance of the model. In this work, we improve the existing attack methods and propose a double level attack method. By constructing the dynamic iterative step size and analyzing the class characteristics of the signals, this method can use the adversarial losses of feature layer and decision layer to generate adversarial examples with stronger attack performance. In order to improve the robustness of the recognition model, we use adversarial examples to train the model, and transfer the knowledge learned from the model to the jamming recognition models in other wireless communication environments by transfer learning. Simulation results show that the proposed attack and defense methods have good performance.**

***Index Terms*—Adversarial attack, adversarial defense, air transportation, communication jamming recognition, deep learning.**

I. Introduction

WIRELESS communication technology has been blooming in recent years. It has the advantages of high flexibility, wide coverage and rapid information transmission, and is applied in many key areas of society. In wireless communication systems, wireless security has been valued and improved by researchers. For example, for the problem that the information of users may be easily intercepted and eavesdropped in non-orthogonal multiple access (NOMA) systems, researchers have improved NOMA technology to ensure secure transmission between secure users and base stations and improve the security of wireless communication systems [2]–[4]. In the field of air transportation, wireless communication is the key technology for communication between aircraft and tower with direct effect on the safe landing and takeoff of aircraft in the air traffic control system. However, in an increasingly complex electromagnetic environment, various communication devices could be interfered by communication jamming [5]. In order to ensure the reliable transmission of information, various anti-jamming technologies emerge. Researchers in different fields have tried to use deep learning networks to complete the identification tasks in wireless communication, and use intelligent anti-interference technology to effectively deal with communication jamming [6]–[8]. Efficient communication anti-jamming technology requires the jammer's jamming mode, and then chooses the corresponding technical means to eliminate or reduce the jamming, so as to ensure normal communication. Therefore, the recognition of jamming signal type is the foundation of communication anti-jamming technology.

In order to automatically monitor different types of communication jamming signals in the air to facilitate anti-jamming processing, researchers have used machine learning (ML) to automatically identify and classify jamming signals in the whole electromagnetic environment for speed and accuracy of classification [9]. Interestingly, researchers have also studied the vulnerability of automatic recognition model, to make the model vulnerable to attacks. Nguyen et al. proved that the deep neural network (DNN) model is easy to be fooled, because they can classify many examples not in the class set as identifiable class members with high confidence [10]. By adding carefully designed small disturbances to the input, Szegedy et al. successfully changed the prediction results of the classifier for the input examples, and proposed the concept of adversarial examples [11]. Sadeghi et al. used receiver classifier and attack algorithms to conduct adversarial attacks against wireless communication related processes [12]. Such attack is fatal for scenarios with high requirements on wireless communication security, such as communication between aircraft and tower. As shown in Fig. 1, the jamming recognition model can correctly identify the original communication jamming, and then the signal receiver uses the corresponding anti-jamming technology to suppress the jamming. However, if the adversarial examples of communication jamming are carefully designed by attackers, the recognition model is likely to incorrectly identify them, which means that the receiver may take wrong anti-jamming measures, leading to accidents. In order to avoid flight accidents, it is necessary to improve the defense ability

This work was supported by the National Natural Science Foundation of China under Grant 62071364 and 62231027, in part by the Key Research and Development Program of Shaanxi under Grant 2023-YBGY-249, in part by the Fundamental Research Funds for the Central Universities under Grant JB210104, and in part by the 111 Project under Grant B08038. An earlier version of this paper was presented in part at the IEEE INFOCOM WKSHPS 2022 [1]. *(Corresponding author: Zhenju Zhang.)*

M. Liu, Z. Zhang and J. Ge are with the State Key Laboratory of Integrated Service Networks, Xidian University, Shaanxi, Xi'an 710071, China (e-mail: mqliu@mail.xidian.edu.cn; zhenjuzhang@stu.xidian.edu.cn; jhge@xidian.edu.cn).

Y. Chen is with the Department of engineering, University of Durham, South Road, dh1 3le, Durham, UK (e-mail: yunfei.chen@durham.ac.uk).

N. Zhao is the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: zhaonan@dlut.edu.cn).

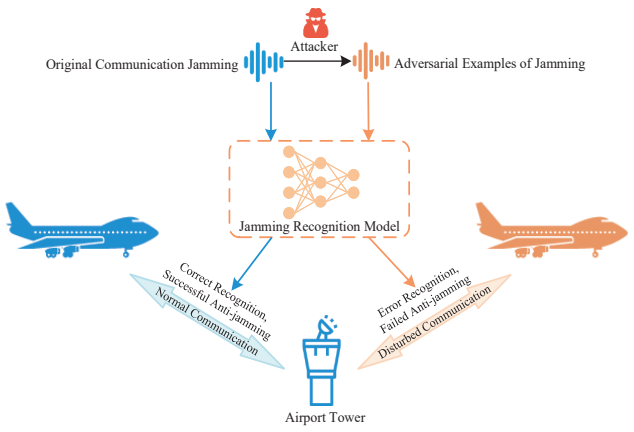


Fig. 1. Communication system model between aircraft and tower.

of the recognition model against adversarial attacks. Therefore, studying the attack and defense of intelligent recognition of communication jamming is of great significance to improve the safety of air traffic.

The existence of adversarial examples has threatened the security and reliability of the jamming recognition model in the field of wireless communication, which shows the urgency of improving the robustness of the model. For different attacks, although various defense methods have been proposed, their good defense performances are limited to certain types of attacks. Yuan et al. introduced some common defense methods in recent years, and pointed out that the performance of these defense methods will change with the change of attack methods and environments [13]. As new attack methods are constantly proposed, the defense model that has been proposed before will become fragile, which prompted researchers to propose new defense methods. For example, Carlini et al. proposed three norm attack methods and successfully broke the robust distillation defense model [14]. Since adversarial attack and defense are two parts of a game process, many researchers have designed a defense method against the attack after proposing a new attack method to improve the defense ability of the classifier [15]. For example, in the field of intelligent transportation, researchers have studied the adversarial attack and defense methods of the intelligent recognition model of urban road conditions, and improved the robustness of the model in automatic driving in the field of image recognition [16], [17]. Today, compared with the image field, there is little work to apply adversarial examples to wireless communication, which means that the communication jamming recognition model will be easily attacked. [18], [19] introduced four label-based adversarial methods, including fast gradient sign method (FGSM), basic iterative method (BIM), projected gradient descent method (PGD) and momentum iterative method (MIM), and used them to generate the adversarial examples, which verified that the recognition model was vulnerable to the attack of the adversarial examples in the field of wireless communication. However, the structure of the network models used by the above methods is relatively simple, and the recognition accuracy of these models on the test set is not high, so the attack effect of the adversarial examples generated by them on other high performance recognition models is not ideal.

In this paper, we evaluate the attack performances of several important attack methods on high performance recognition model, and propose an attack method based on double level confrontation. In the attack, compared with the traditional methods, we construct a dynamic iterative step size, and generate adversarial examples with stronger attack performance by integrating feature layer confrontation and decision layer confrontation, which have better attack effects on high performance model. Adversarial examples have model details, and the goal is to make the recognition model misclassify these examples with high confidence. Then, the generated adversarial examples are used to train the recognition model, and the knowledge learned by the model is transferred to the jamming recognition models in other wireless communication environments by using transfer learning.

The main motivations and contributions of this paper are summarized as follows:

1) The jamming recognition models are trained in different wireless communication environments to identify the types of communication jamming to achieve high recognition accuracy.

2) New attack methods are proposed using dynamic iterative step to improve the attack method by analyzing the class characteristics of signals in the model, and generate double level adversarial examples by using the generated decision level confrontation and feature level confrontation to enhance the attack performance of the adversarial examples.

3) The adversarial examples are used to train the jamming recognition model, and use the transfer learning method to enhance the defense performance of the jamming recognition models in different environments.

The rest of the paper is organized as follows: Section II briefly summarizes the related work in this field. Section III proposes an attack method based on double level confrontation, and describes its implementation principle and process. In Section IV, a method of transferring adversarial knowledge is proposed to improve the defensive performance of recognition models in other environments for adversarial examples. Section V shows a series of experimental results and analyses to evaluate the performance of the proposed adversarial attack and defense methods. Finally, Section VI gives the conclusion.

II. Attack and Jamming Recognition

A. Adversarial Attacks

Since DL is vulnerable to adversarial examples, the early inferential explanation is the highly nonlinear feature of DNN. However, Goodfellow et al. pointed out that due to the high feature dimension of the input and the linearity of the model, adversarial examples can be generated to attack the recognition model, and they proposed the FGSM attack method [20]. After FGSM was proposed, many new attack methods were developed, such as BIM, PGD and MIM. They belong to the L_∞ norm attack. When generating adversarial example, the norm is regarded as a measure of the perceived similarity

between the original example and the adversarial example, and is often used to constrain the generation of adversarial perturbation. The L_p norm of the adversarial perturbation η with n_e elements is defined as

$$\|\eta\|_p = \left(\sum_{i=1}^{n_e} \|\eta_i\|^p \right)^{\frac{1}{p}}, \tag{1}$$

it represents the constraint on the number of non-zero perturbation vectors when $p = 0$, the Euclidean distance between the adversarial example and the original example when $p = 2$, and the maximum variation of all sample values in the adversarial example when $p = \infty$.

1) FGSM: FGSM is a simple and fast method to generate adversarial examples. By calculating the gradient of the loss function to the input, the attack direction can be determined. Then, by adding a fixed step in this direction as an adversarial perturbation, and adding the perturbation to the original input, the adversarial example is generated. FGSM only needs a single iteration to generate adversarial examples, but it cannot update these examples by querying model parameters multiple times to enhance attack performance. FGSM can be expressed as

$$\begin{cases} \eta = \varepsilon \cdot \operatorname{sign}\left(\nabla_x J\left(x, l\right)\right), \\ x^* = x + \eta, \end{cases} \tag{2}$$

where x is the original input, l is the real label, J is the loss function, $\nabla_x J$ is the gradient of the loss function to the input, η is the adversarial perturbation, ε is the maximum perturbation level allowed to generate the adversarial example, and x^* is the generated adversarial example. For L_∞ norm attack, $\|\eta\|_\infty \leq \varepsilon$ should be satisfied.

2) BIM: Compared with FGSM, BIM generates adversarial examples with multiple iterations, so that the gradient direction can be adjusted after each iteration [21]. The number of iterations is denoted as N, and the iteration step is $\alpha = \varepsilon/N$ during the iteration. BIM can be expressed as

$$\begin{cases} {x_0}^* = x, \\ {x_{n+1}}^* = \operatorname{Clip}_{x,\varepsilon}\left\{ {x_n}^* + \alpha \cdot \operatorname{sign}\left(\nabla_{{x_n}^*} J\left({x_n}^*, l\right)\right)\right\}, \end{cases} \tag{3}$$

where n denotes the nth iteration and its value is $0, 1, \cdots, N-1$. $\operatorname{Clip}_{x,\varepsilon}\{\cdot\}$ denotes that after each iteration, the adversarial examples are cut to meet the L_∞ constraint.

3) PGD: On the basis of BIM, PGD generates random perturbation within the neighborhood of the original example, and uses it as the initial input of the algorithm. After several iterations, it generates the adversarial example with stronger attack performance [22]. Before generating adversarial examples, PGD will add random noise to the original examples. The essence of this method is that the projection gradient on the negative loss function decreases, and its iterative process is

$${x_{n+1}}^* = \prod_{x+S} \left({x_n}^* + \alpha \cdot \operatorname{sign}\left(\nabla_{{x_n}^*} J\left({x_n}^*, l\right)\right)\right), \tag{4}$$

where $\prod_{x+S}(\cdot)$ means that the adversarial perturbation is limited to the sphere. The iterative process represents performing gradient descent after randomly selecting projection points in $x + S$.

4) MIM: Compared with the above three methods, MIM introduces momentum into the iterative attack [23]. By accumulating velocity vectors in the gradient direction of the loss function, MIM can accelerate the gradient descent. This method can well solve the problem of local optimal solution and over-fitting in the iterative process, and has strong generalization ability. MIM can be expressed as

$$\begin{cases} {x_0}^* = x, g_0 = 0, \\ g_{n+1} = \mu \cdot g_n + \frac{\nabla_{{x_n}^*} J({x_n}^*, l)}{\left\|\nabla_{{x_n}^*} J({x_n}^*, l)\right\|_1}, \\ {x_{n+1}}^* = \operatorname{Clip}_{x,\varepsilon}\left\{ {x_n}^* + \alpha \cdot \operatorname{sign}\left(g_{n+1}\right)\right\}, \end{cases} \tag{5}$$

where g_n denotes the accumulated gradient generated by the previous n iterations, μ is the attenuation factor of g_n, and $\|\cdot\|_1$ is the sum of the absolute values of the elements in the vector.

B. Communication Jamming Signals Model

Communication jamming can destroy or disturb the information transmission of communication system, which can be classified into blanket jamming and deception jamming according to jamming style. Blanket jamming refers to the transmission of jamming signals to cover the spectrum of each other's signal to reduce the signal-to-noise ratio (SNR) at the communication receiver to interfere with the normal operation of the receiver. Deception jamming is a jamming method that imitates the wireless signal used by the other party to make it unable to extract effective information.

The typical types of blanket jamming include single-tone jamming, multi-tone jamming, noise band jamming, noise frequenncy modulation (FM) jamming, and linear frequency modulation (LFM) jamming.

Single-tone jamming is the simplest type of jamming, which is essentially a sinusoidal signal composed of a single frequency component. Its time domain expression is

$$J(t) = A\exp\left(j\left(2\pi f_c t + \varphi_0\right)\right), \tag{6}$$

where A denotes the amplitude of the jamming signal, f_c stands for the carrier frequency, and φ_0 is the initial phase.

Multi-tone jamming is composed of multiple single-tone jamming, which can be expressed as

$$J(t) = \sum_{m=1}^{M} A_m \exp\left(j\left(2\pi f_m t + \varphi_m\right)\right), \tag{7}$$

where M represents the tone number of multi-tone jamming, A_m, f_m and φ_m are the amplitude, carrier frequency and initial phase of the mth single-tone jamming that constitutes multi-tone jamming, respectively.

Noise band jamming means that the energy of noise is concentrated in the specified band range, and the time domain expression is

$$J(t) = U_n(t)\exp\left(j\left(2\pi f_c t + \varphi_0\right)\right), \tag{8}$$

where $U_n(t)$ is a Gaussian white noise with mean value of 0 and variance of σ_n^2.

The frequency of noise FM jamming is affected by the modulation noise, which can be expressed as

$$J(t) = A\exp\left(j\left(2\pi f_c t + k_{fm}\int_0^t \xi\left(t'\right)dt'\right)\right), \quad (9)$$

where k_{fm} is the frequency modulation coefficient, $\xi(t)$ is the Gaussian white noise with mean value of 0 and variance of σ_n^2.

LFM jamming frequency and time is linear, at a certain time only contains a single frequency, in a certain time range with broadband scanning characteristics. This jamming can be expressed as

$$J(t) = A\exp\left(j\left(2\pi f_c t + \pi k t^2 + \varphi_0\right)\right), \quad (10)$$

where k is the frequency of linear frequency modulation.

When the jamming signal uses Pseudo Noise code which has a certain correlation with the real spread spectrum code and can achieve good interference effect in good synchronization, this jamming is called random binary code modulation jamming, also known as BPSK jamming. It can be expressed as

$$J(t) = s(t)cos\left(2\pi f_c t + \varphi_0\right), \quad (11)$$

where $s(t) = A\sum_n a_n g(t - nT_s)$, $a_n = -1$ or $a_n = +1$, $g(t)$ is a rectangular pulse with T_s pulse width. According to the code rate, the jamming can be divided into narrowband interference and broadband jamming. When the symbol rate T_s is less than the real symbol rate, it is narrowband BPSK jamming (BPSK_NBJ), and when T_s is greater than the real symbol rate, it is broadband BPSK jamming (BPSK_WBJ).

All the above seven kinds of communication jamming are considered. Under the assumption of perfect sampling synchronization, we set the sampling frequency as 10 MHz, and randomly set the carrier frequency and initial phase of the interference. We set the tone number of multi-tone jamming as 4, set the roll-off coefficient of the shaping function of BPSK jamming as 0.35. We randomly set the bandwidth factor of noise band jamming in $[0.1, 0.7]$, and the frequency modulation coefficient of noise FM jamming in $[0.125, 0.933]$. When generating the communication jamming example data set, we set the jamming noise ratio (JNR) from -20 dB to 18 dB and the interval is 2 dB, each jamming generates 1000 examples under each JNR. Then, we set the number of sampling points to 128 and store the jamming in the data set after sampling in the form of in-phase component and orthogonal component. Thus, the dataset contains 140,000 jamming examples.

C. Intelligent Recognition Networks Model of Communication Jamming

In recent years, many researchers have successfully applied neural networks to the automatic recognition of radio signal categories. Wang et al. combined two convolutional neural networks trained on different data sets, and used dropout instead of pooling to achieve higher modulation recognition accuracy [24]. Tang et al. proposed a programmed data enhancement method by using the auxiliary classifier to generate adversarial networks (ACGANs), and obtained better classification accuracy of communication signal modulation [25]. Rajendran et al. introduced Long Short-Term Memory (LSTM), which can solve the problem of gradient disappearance and gradient explosion in the process of long sequence training, into modulation recognition, and proved the excellent recognition performance of the network [26]. O'Shea et al. used ResNet to classify radio signals, which proved that ResNet had good recognition effect in the field of wireless communication [27]. In order to better reflect the attack effect of different attack methods, this paper uses ResNet as the target model of adversarial attack.

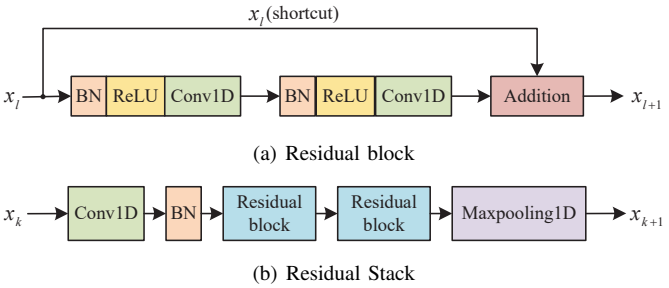


Fig. 2. Partial structure of ResNet.

TABLE I
RESNET STRUCTURE

Layer	Output dimensions
Reshape	128×2
Residual Stack	64×32
Residual Stack	32×32
Residual Stack	16×32
Residual Stack	8×32
Residual Stack	4×32
Residual Stack	2×32
Flatten	64
FC/Dropout	128
FC/Dropout	128
FC/Softmax	7

Before ResNet appeared, the layers of the neural network were not very deep, because with the deepening of the network, the network was prone to gradient saturation or gradient explosion in the training process, resulting in larger errors. ResNet is a convolutional neural network proposed by He et al., using "shortcut connection" [28]. This makes it easy for each residual block to learn identity mapping and solve the degradation problem that is common in DNNs, which means we can add a lot of residual blocks without compromising the performance of the training set. Later, He et al. used identity mapping in residual block and after-addition activation to further enhance the learning efficiency and effectiveness of the network. In this paper, the improved ResNet is used to identify and classify the jamming signals, and is used to generate adversarial examples. The partial network structure of ResNet is shown in Fig. 2.

The target model of this paper includes six Residual Stack structures, and there are 33 layers of convolution layer and full connection layer of the model. The components of the network and the output shape of each part are shown in Table I.

When training the network, we use 80% of the examples in the dataset as the training set and 20% of the examples as the

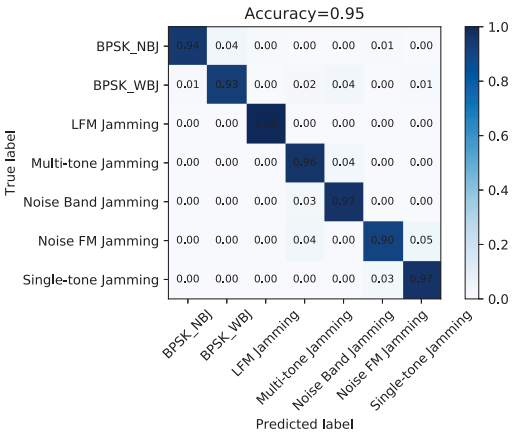


Fig. 3. Confusion matrix of communication jamming recognition with JNR=6dB.

test set. We set the number of epochs and the size of training batches of the network to 100 and 1024, respectively, and set the initial learning rate to 0.001. In order to make the network converge faster, we adopt an automatic updating mechanism for learning rate: if the loss value of the test set during the iteration of the training network does not decrease after three consecutive iterations, the learning rate is halved.

After the training, we use the jamming recognition model to identify the jamming signals with JNR=6dB in the test set, and use the confusion matrix to show the recognition effect. The confusion matrix is a matrix used to observe the classification results of the recognition model. The elements on the diagonal of the confusion matrix represent the correct recognition rate of the model for each category signal in the test set. The recognition results are shown in Fig. 3. As can be seen from Fig. 3, when JNR=6dB, the recognition accuracy of the jamming recognition model trained in this paper is 0.95, and the recognition accuracy for each type of jamming signal is not less than 0.9, which has a good recognition effect.

III. Double Level Adversarial Attack

A. Dynamic Iterative Adversarial Attack

In multi-classification tasks, cross entropy loss function can be used to characterize the difference between predicted values and real labels of neural networks. At the decision-making level of the recognition model, the cross entropy loss of the model for the input jamming signal can be expressed as

$$L_d = -\frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} l_{ij}\left(x_i\right)\log_2\left(p_{ij}\left(x_i\right)\right), \tag{12}$$

where N_1 represents the number of input signals x_i, N_2 represents the number of signal classification labels, $l_{ij}\left(x_i\right)$ is the real label of input signals, and $p_{ij}\left(x_i\right)$ is the prediction probability distribution of recognition model for input signals.

In FGSM, BIM, PGD and MIM, the iterative step length is fixed. Hence, when adversarial examples are generated, only the perturbation direction changes, while the perturbation size does not change. In some cases, a fixed iteration step may cause the iteration process to be unable to move forward or over iterate. In order to make this step change with iteration, a dynamic iteration step can be designed to dynamically change the perturbation size in the disturbance direction, so that the adversarial example can better approach the optimal point of the loss function within a limited number of iterations.

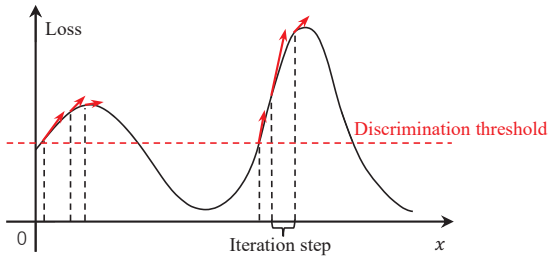


Fig. 4. Dynamic iterative step process.

The implementation process of dynamic iterative step size is shown in Fig. 4. When the iteration point is close to the extreme point of the loss function of the model, the absolute value of the gradient at the point of the loss function is small. In order to avoid the iterative point skipping the extreme point directly when the step size is too large, the iterative step size should be reduced. On the contrary, when the iterative point is close to the middle of the two adjacent extreme points, the absolute value of the gradient is larger and the projection of the value on the transverse axis is smaller. In order to quickly reach the extreme point nearest to the iterative point, the iterative step size should be appropriately increased. Therefore, the gradient size $\left|\nabla_{x_n} L_d\right|$ of the loss function at the input point can positively adjust the iteration step size. In addition, in order to reflect the direction and intensity of the loss function change, we use the gradient difference between the current iteration point and the previous iteration point as the supplementary information of the current iteration step size, so as to achieve the purpose of using historical information to correct the iteration process. In summary, we set the iterative step size $\left|\nabla_{x_n} L_d + \nabla_{x_n} L_d - \nabla_{x_{n-1}} L_d\right| = \left|2\nabla_{x_n} L_d - \nabla_{x_{n-1}} L_d\right|$. In practice, in order to narrow the scope of the step size, it needs to be normalized to

$$\alpha_n = \frac{\left|2\nabla_{x_n} L_d - \nabla_{x_{n-1}} L_d\right|}{\left\|2\nabla_{x_n} L_d - \nabla_{x_{n-1}} L_d\right\|_1}. \tag{13}$$

In order to avoid the problem that the dynamic iterative step size deviates from the initial step size too large, which leads to the iteration stagnation or skips the extreme point directly, we reduce the step size, so that the minimum dynamic iterative step size is not less than half of the original fixed step size ε/N, and the maximum is not more than the limitation of L_∞. The step size after truncating is distributed in the interval $[0.5 * \ \varepsilon/N,\ \varepsilon]$, avoiding $\alpha_n \ll \varepsilon$ or $\alpha_n \gg \varepsilon$. When the loss function value of the adversarial examples at the final iteration point is greater than the discriminant threshold, the recognition model will classify the examples incorrectly, which can be regarded as that the examples successfully implement the attack on the recognition model.

We call the attack method proposed in this section Dynamic Iterative Method (DIM). Compared with MIM, DIM adopts dynamic iterative step, and its implementation process can be expressed as

$$\begin{cases} {x_0}^* = x, \alpha_0 = 0, g_0 = 0, \\ \alpha_n = \frac{\left|2\nabla_{x_n{}^*} L_d - \nabla_{x_{n-1}{}^*} L_d\right|}{\left\|2\nabla_{x_n{}^*} L_d - \nabla_{x_{n-1}{}^*} L_d\right\|_1}, \\ g_{n+1} = \mu \cdot g_n + \frac{\nabla_{x_n{}^*} L_d}{\left\|\nabla_{x_n{}^*} L_d\right\|_1}, \\ {x_{n+1}}^* = \mathrm{Clip}_{x,\varepsilon}\left\{{x_n}^* + \alpha_n \cdot \mathrm{sign}\left(g_{n+1}\right)\right\}. \end{cases} \tag{14}$$

B. Dynamic Iterative Feature Adversarial Attack

Huang et al. introduced the activation vector to visualize the features extracted by different radio modulation recognition models based on DL, and proved that these features largely depend on the content carried by the radio signal [29]. Therefore, the features extracted by the recognition model well reflect the characteristics of the signal itself and the differences between signals. If confrontation can be generated from the internal characteristics of the signal, the performance of the recognition model can be better attacked.

The traditional adversarial attack methods are mainly label-based, aiming to make the recognition model produce wrong class labels. This belongs to label adversaries. Sabour et al. proposed a new feature attack method, called feature adversaries (FA), based on the characteristics of the internal layer of DNN, aiming to generate an adversarial example to fool classifier from the internal feature space [30]. The internal characteristics of the adversarial examples generated by this method are similar to those of other types of examples, but the adversarial examples themselves are difficult to be visually distinguished, making adversarial examples look like the original inputs. In a feature space within the DNN, the FA method takes the square of the Euclidean distance between the features of the input examples and the features of the target examples as the error function, and realizes the targeted attack by minimizing the loss function. Specifically, the adversarial example x^* can be defined as the solution to the following constrained optimization problem

$$\begin{cases} x^* = arg \ \min\limits_{x} \left\|f_k(x) - f_k\left(x_t\right)\right\|_2^2, \\ subject \ to \ \ \left\|x - x_s\right\|_\infty < \varepsilon \ , \end{cases} \tag{15}$$

where x_s and x_t represent the original example and the target example respectively, and f_k represents the mapping of the network model to the characteristics of the signal at the kth layer. ε is the maximum value that the adversarial perturbation can reach to limit the perception of the adversarial example.

Due to the characteristics of the algorithm, the implementation of the FA method needs to specify the input signal as the target signal to generate the corresponding target features at the specified feature layer, so this method is only applicable to targeted attack. In addition, the selection of target signal seriously affects the performance of the method. When different individuals of the same class of jamming signals are used as target signals, the difference in attack performance may be large. In order to study the cause of this performance difference, we choose JNR of 0 dB, 6 dB, 12 dB and 18 dB

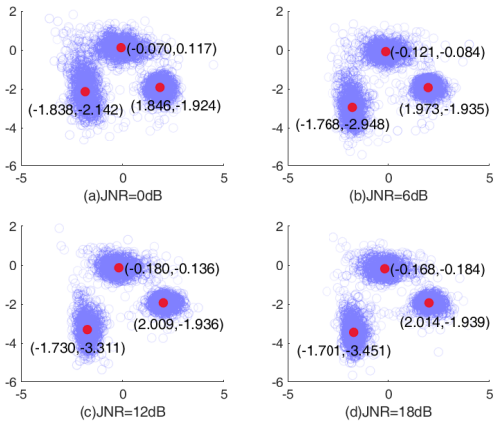


Fig. 5. Characteristic scatter plot of multi-tone jamming with different JNRs.

for multi-tone jamming as the network input, select the last residual block of the network model as the feature space to be studied, and output the characteristics of different signals.

In order to obtain the aggregation characteristics of the signal in the feature space, we project the feature points mapped by the model to the two-dimensional plane to form the feature scatter diagram. For the feature points far away from the feature aggregation area, the medians of all feature points on different coordinate axes are obtained and taken as the feature center, so that the influence of abnormal features on the value of feature center can be well reduced when there are many feature points. The classification feature center of the signal in the kth layer feature space of the network can be expressed as

$$F_k(x) = \begin{cases} f^*_{\frac{M+1}{2}}(x), M \ is \ odd, \\ \frac{1}{2} \cdot \left(f^*_{\frac{M}{2}}(x) + f^*_{\frac{M}{2}+1}(x)\right), M \ is \ even, \end{cases} \tag{16}$$

where M represents the number of feature points, and $f^*(x)$ represents the coordinate value of the input feature mapping from small to large.

According to the ResNet structure in this paper, the output characteristics of the signal in the last residual layer can be regarded as 32 feature points. In order to facilitate observation and analysis, we compare the first three feature points, and obtain some feature scatter diagrams under four JNRs, as shown in Fig. 5.

Fig. 5 shows multi-tone jamming when JNR is 0 dB, 6 dB, 12 dB and 18 dB respectively. It can be seen that in the feature space within the network model, the characteristics of jamming signals have strong aggregation. By integrating 32 feature points including those in Fig. 5, the ResNet model in this paper can classify and identify various jamming signals. At the same time, in the recognition process, a small number of signals will be identified wrongly, which is reflected in the feature map that is the scattered feature points far from the aggregation area. Therefore, when the feature points of the target signal used by FA algorithm are far away from the aggregation area, its attack performance will decrease. In addition, in Fig. 5, the feature centers and values of the signals under the four JNRs have been labeled with red dots. It can

be seen that there is no significant difference in the center points of the four groups, and the difference decreases with the increase of JNR. Thus, we infer that in the feature space of the model, the characteristics of the same class of signals under different JNRs have strong similarity, and the higher the JNR is, the stronger the similarity is. It should be noted that the feature centers in Fig. 5 are only 3 of the 32 feature centers of multi-tone jamming signals. Since the data set used in this paper contains seven kinds of jamming signals, the model will produce seven groups of feature vectors in the training process as the basis for classification, and each feature vector contains 32 feature centers.

In order to solve the problem of large deviation of attack results caused by different target signal selection in FA method, we select seven groups of feature vectors corresponding to seven types of signals under the JNR of 10 dB as class features to generate confrontation in the feature layer of the model. At this time, for different signals of the same class, they share one class feature. The difference between the real feature of the input signal and its corresponding class feature is taken as the feature loss. By increasing the feature loss or reducing the feature loss, the untargeted or targeted attack on the recognition network can be realized respectively. We use the Euclidean distance to measure the feature difference of the signal in the feature space of the recognition model, and the feature loss of the network at the kth layer can be expressed as

$$L_f = \frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_3}\left\|f_k\left(x_{ij}\right) - F_k\left(x_{ij}\right)\right\|_2^2, \tag{17}$$

where N_1 and N_3 represent the number of input signals and the number of feature points in the kth feature layer of the network, respectively. $f_k\left(x_{ij}\right)$ and $F_k\left(x_{ij}\right)$ represent the real characteristics of the signal in this feature layer and the corresponding class characteristics, respectively. The purpose of untargeted attack is to make the model identify the jamming incorrectly by maximizing the loss function, and $F_k\left(x_{ij}\right)$ represents the class characteristics corresponding to the input signal. The purpose of targeted attack is to make the model identify the target as a specified type by minimizing the loss function. At this time, $F_k\left(x_{ij}\right) = F_k\left(x_t\right)$ represents the class feature of the specified target signal x_t.

Therefore, compared with the label-based attack method in the decision layer using real labels and predictive value to calculate the loss function, the attack method in this section uses the feature vector of the example to calculate the loss function in the feature layer of the model. Compared with the FA method, we use class features to replace the characteristics of individual signals, and use the previous method of constructing dynamic iterative step to generate adversarial examples. We call the attack method proposed in this section Dynamic Iterative Feature Adversaries (DIFA). Compared with the DIM proposed in the previous section, DIFA only changes the loss function, and the feature loss L_f is used as the loss function to achieve feature confrontation.

C. Double Level Adversarial Attack

To further enhance the adversarial performance, we consider generating adversarial examples from feature level adversarial and decision level adversarial. Previously, we have obtained the feature loss L_f by using the class features of the signals. In the double level attack method, we first use the gradient of feature loss to the input signal to determine the direction of feature level adversarial perturbation. The perturbation direction can be expressed as

$$\operatorname{sign}\left(g_f\right) = \operatorname{sign}\left(\mu \cdot g_n + \frac{\nabla_{x_n{}^*} L_f}{\left\|\nabla_{x_n{}^*} L_f\right\|_1}\right), \tag{18}$$

where g_n represents the gradient accumulation of the nth iteration, μ represents the attenuation factor, $x_n{}^*$ represents the adversarial examples generated by the nth iteration, and $\nabla_{x_n{}^*} L_f$ represents the gradient of characteristic loss for these examples.

After determining the direction of adversarial perturbation, referring to the method of constructing dynamic iterative step size in DIM, we calculate the level of perturbation by using the gradient of feature loss at the input point. In each iteration, the overall perturbation level is divided into feature layer perturbation and decision layer perturbation. The perturbation size of the feature layer can be expressed as

$$\lambda \cdot \alpha_n = \lambda \cdot \frac{\left|2\nabla_{x_n{}^*} L_f - \nabla_{x_{n-1}{}^*} L_f\right|}{\left\|2\nabla_{x_n{}^*} L_f - \nabla_{x_{n-1}{}^*} L_f\right\|_1}, \tag{19}$$

where α_n is the global perturbation level for each iteration. λ is the perturbation factor for the feature layer, representing the proportion of perturbation used when the feature layer produces confrontation. It is a hyper-parameter with a value range of $[0, 1]$, which needs to be artificially set before generating adversarial examples.

After determining the perturbation direction and perturbation size under the infinite norm constraint, the feature level confrontation can be generated by

$$x_f{}^* = \operatorname{Clip}_{x,\varepsilon}\left\{x_n{}^* + \lambda \cdot \alpha_n \cdot \operatorname{sign}\left(g_f\right)\right\}. \tag{20}$$

After the feature level confrontation is obtained, the confrontation is used to replace the original input and spread to the decision level in the network model to obtain the prediction probability of the model for the confrontation. The real label of feature confrontation is taken as the real probability distribution, and input into the cross entropy loss function together with the prediction probability to calculate the decision loss, which can be expressed as

$$L_d = -\frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} l_{ij}\left(x_f{}^*\right)\log_2\left(p_{ij}\left(x_f{}^*\right)\right), \tag{21}$$

where $l_{ij}\left(x_f{}^*\right)$ is the real label of feature confrontation, and $p_{ij}\left(x_f{}^*\right)$ is the prediction probability distribution of feature confrontation by the recognition model.

On the basis of the gradient accumulation of the feature layer g_f, the gradient of the decision loss for the feature confrontation $x_f{}^*$ is continuously accumulated to determine the direction of decision level confrontation, which is also the

final gradient accumulation of this iteration. The perturbation direction can be expressed as

$$\operatorname{sign}(g_{n+1}) = \operatorname{sign}\left(g_f + \frac{\nabla_{x_f{}^*} L_d}{\left\|\nabla_{x_f{}^*} L_d\right\|_1}\right). \quad (22)$$

At the same time, the residual perturbation level of the feature layer is taken as the size of the decision layer confrontation, which is $(1-\lambda)\cdot\alpha_n$. Then, the production process of double level confrontation is completed by

$$x_{n+1}{}^* = \operatorname{Clip}_{x,\varepsilon}\left\{x_f{}^* + (1-\lambda)\cdot\alpha_n \cdot \operatorname{sign}(g_{n+1})\right\}. \quad (23)$$

In the whole process of generating double level confrontation, when $\lambda = 0$, the perturbation of feature layer confrontation is 0, and the double level confrontation degenerates into decision level confrontation. When $0 < \lambda < 1$, it means that after the feature layer uses a part of perturbation to generate initial feature confrontation, it continues to use residual perturbation to deal with feature confrontation results in the decision layer to enhance confrontation performance. When $\lambda = 1$, it means that the perturbation of the decision layer is 0, and the double level confrontation degenerates into the feature level confrontation.

We call the attack method in this section Double Level Attack (DLA). Algorithm 1 summarizes the detailed algorithm steps of DLA.

Algorithm 1 Double Level Attack

Input: Original signal example x; ground-truth label l; loss function L of a classifier.

Input: Perturbation constraint ε; decay factor μ; feature layer perturbation factor λ.

Output: An adversarial example x^* with $\|x^* - x\|_\infty \le \varepsilon$.

$x_0{}^* = x$; $g_0 = 0$; $\alpha_0 = 0$;

for $n = 0$ to $N-1$ **do**

Input $x_n{}^*$ to the classifier, calculate the feature loss L_f according to (17), and determine the direction of the feature level perturbation according to (18);

Calculate the perturbation size of each iteration

$$\alpha_n = \frac{\left|2\nabla_{x_n{}^*} L_f - \nabla_{x_{n-1}{}^*} L_f\right|}{\left\|2\nabla_{x_n{}^*} L_f - \nabla_{x_{n-1}{}^*} L_f\right\|_1}$$

Update feature level confrontation

$$x_f{}^* = \operatorname{Clip}_{x,\varepsilon}\left\{x_n{}^* + \lambda\cdot\alpha_n\cdot\operatorname{sign}(g_f)\right\}$$

Calculate the decision loss L_d according to (21), and determine the direction of the double level perturbation according to (22);

Update double level confrontation

$$x_{n+1}{}^* = \operatorname{Clip}_{x,\varepsilon}\left\{x_f{}^* + (1-\lambda)\cdot\alpha_n\cdot\operatorname{sign}(g_{n+1})\right\}$$

end for

return $x^* = x_N{}^*$

In fact, without considering the time cost, in order to further enhance the attack performance, the adversarial examples generated by the attack algorithm can be re-used as the input of the model to generate new adversarial examples, so that these examples can be as close as possible to the iterative optimal point.

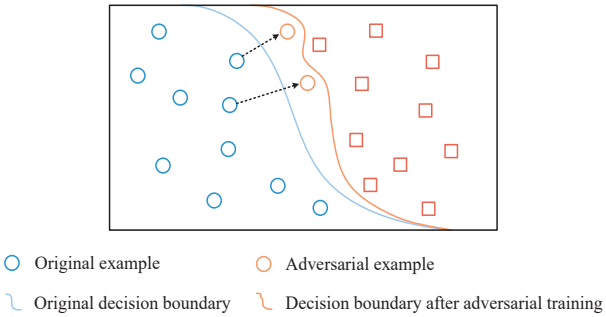


Fig. 6. Illustration of adversarial training.

IV. Adversarial Defense and Defense Transfer

Our main purpose is to improve the robustness of the communication jamming recognition model used in air traffic control. Therefore, after obtaining adversarial examples with strong attack performance, we can use them to continuously improve the decision boundary of the recognition model through adversarial training to enhance the defense performance of the model against adversarial attacks. In addition, we can use transfer learning to transfer the defense knowledge learned by the model to a new communication environment, so that the new model has the ability to defend against adversarial examples even it has not had adversarial training.

A. Adversarial Training

Among various defense methods to enhance the robustness of recognition model, adversarial training is an active defense method against adversarial examples. This method uses adversarial examples to train the network model, which effectively improves the robustness of the model by minimizing the loss of adversarial examples generated in each training step. Studies have shown that adversarial training is one of the most effective defense methods against adversarial attacks, which can well improve the defense performance of the recognition model [31], [32].

The learning framework of adversarial training can be summarized as a typical min-max optimization problem as

$$\min_{\theta} \max_{D(x,x^*)<\eta} L(\theta, x^*, l), \quad (24)$$

where $L(\theta, x^*, l)$ is an adversarial loss function, θ is the network weight, $D(x, x^*)$ represents a distance measure between the original input and the adversarial example. In adversarial training, the optimization problem of the model can be divided into two parts. As shown in Fig. 6., the internal optimization problem is to determine the perturbation of the maximum loss function through the attack algorithm, so that the example passes through the original decision boundary after the attack, and fool the model to classify it into another class. The external minimization problem is to minimize the loss function

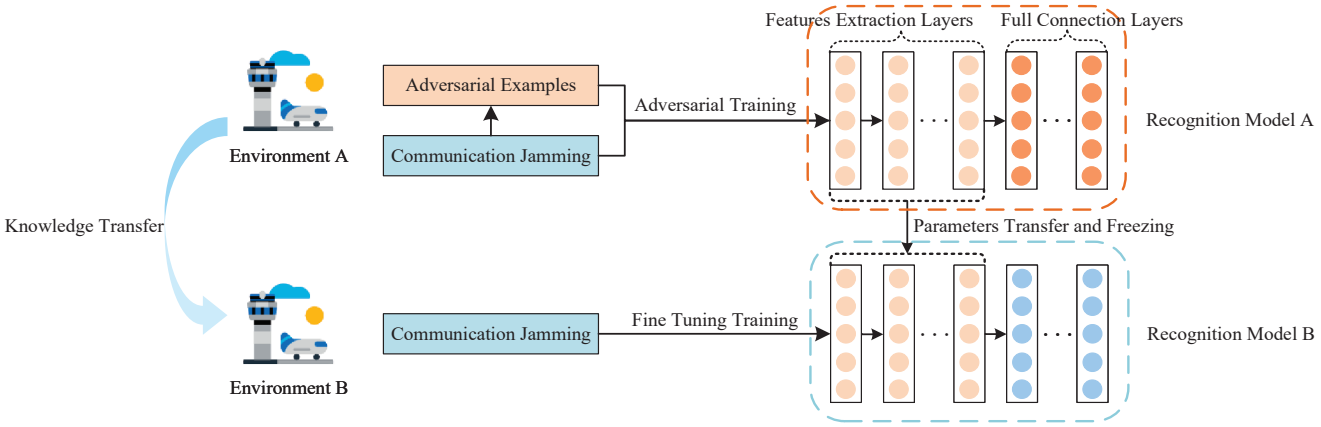


Fig. 7. Defense transfer model.

in the training process, and the decision boundary of the model is changed during this process, so that the model can classify the adversarial example into the correct class. The training process of min-max can generate a robust model with high resistance to most adversarial attacks, and can successfully classify the original examples and adversarial examples.

In general, the stronger the aggression of the adversarial examples is, the better the defense performance of the defense model obtained by adversarial training is. Therefore, we use the DLA algorithm proposed in this paper to generate adversarial examples and use them for adversarial training. The obtained model will have strong robustness to these adversarial examples.

B. Defense Transfer

In machine learning, it is often assumed that the current training data and future training data are in the same feature space and have the same distribution. But in many practical applications, this assumption may not hold. At this time, the knowledge of the trained model can be transferred to a new scene through transfer learning, so as to improve the efficiency and performance of learning. Pan et al. introduced some successful applications of transfer learning and pointed out that the similarity between the two fields may be negatively transferred because of the large difference in data distribution [33].

When the jamming recognition model is applied to the actual airport environment, there are often some differences between the jamming signals in different environments A and B. The result of directly using the jamming recognition model trained in environment A to identify the jamming in environment B is often not satisfactory. However, retraining a new recognition model in environment B not only consumes a lot of time, but also needs to re-use the model to generate adversarial examples for adversarial training to improve the defense performance of the model. In order to use the knowledge learned in environment A to fight adversarial attacks in environment B, we propose a transfer method of defense model, as shown in Fig. 7.

In Fig. 7, we assume that the jamming recognition model A has been obtained through adversarial training in environment A. Then, we share all the network parameters of the feature extraction layers before the full connection layers in model A with model B, and freeze these parameters in model B. Since the feature extraction layers of model A has been trained with knowledge of adversarial examples, model B also obtains defense capability through these shared network parameters without additional time to generate adversarial examples for adversarial training. After freezing the transferred parameters, model B uses fine-tuning training to learn the difference of communication jamming between environment B and environment A through the full connection layers.

V. Numerical Results and Discussion

In this section, we will test the untargeted attack performance of different attack methods by simulation. After generating adversarial examples, we will obtain a robust defense model by adversarial training, and use the transfer learning method to enhance the defense performance of jamming recognition models in an other environment.

A. Jamming Noise Ratio

JNR represents the ratio of jamming signal power to noise power, reflecting the strength relationship between jamming and noise. In order to study the effect of JNR on adversarial attack performance, we set the perturbation upper bound $\varepsilon = 0.0018$ to generate the adversarial examples with different JNRs. These adversarial examples are identified using the jamming recognition model, as shown in Fig. 8. In Fig. 8, we show the accuracy of the recognition model for adversarial examples generated under different JNRs. It can be seen that when JNR < -4 dB, the attack performance of DIM, DIFA and DLA proposed in this paper is slightly better than that of other methods. When JNR > -4 dB, the performance advantages

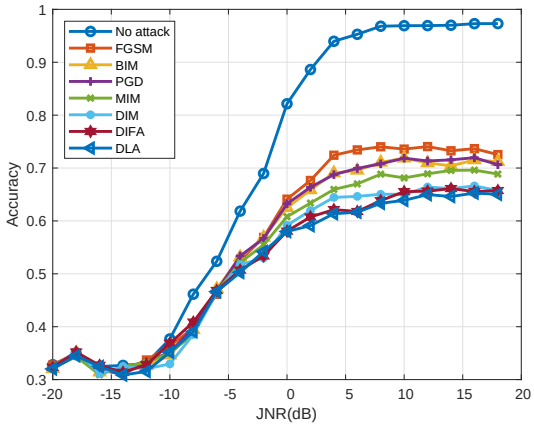


Fig. 8. Attack performance with different JNRs.

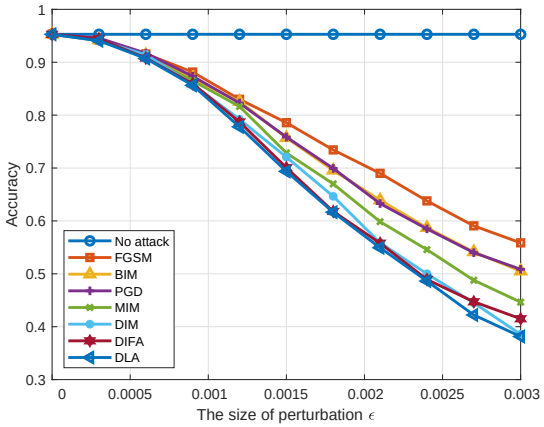
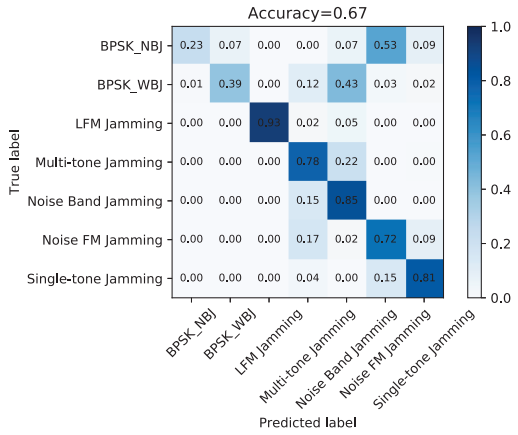


Fig. 9. Attack performance with different perturbations.

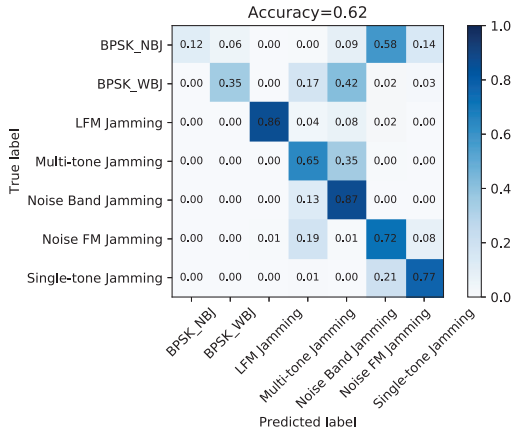
of these three methods gradually appear. With the increase of JNR, the performance gap between the attack methods proposed in this paper and the traditional methods is becoming larger. When the JNR is the same, the proposed methods make the accuracy of the model decrease more, indicating that their attack performance is better than the traditional methods. At the same time, when the accuracy of the model decreases as much, the JNR required by the proposed methods is larger, indicating that the proposed methods still have good performance under high JNR. Because DLA combines the improved ideas of DIM and DIFA, it has the best attack effect.

B. Adversarial Perturbation

The upper bound of adversarial perturbation is an important criterion to measure the concealment performance of adversarial examples. It refers to the maximum variation range of adversarial examples compared with the original examples, which determines whether the adversarial examples can successfully deceive the receiver and attack the recognition model. In order to study the influence of perturbations on the performance of adversarial attacks, we take the maximum constraint value of perturbation in the interval $[0, 0.003]$ and the length of interval is 0.0003. Using feature layer perturbation factor $\lambda = 0.2$ and JNR = 6 dB, we generate adversarial examples with different perturbation values and use the trained ResNet jamming recognition model to identify these examples. In Fig. 9, we show the impact of the size of the perturbation on the attack performance. When JNR = 6 dB and there is no attack, the accuracy of the jamming recognition model reaches 95.29%, which shows that the model has a good recognition effect. In addition, from Fig. 9, we can observe that the attack performance of DIM, DIFA and DLA proposed in this paper is significantly better than the traditional four attack methods. In our three methods, DLA has better attack performance than DIM and DIFA under any perturbation. Because the perturbation value will reduce the concealment of the adversarial example, we use the perturbation value $\varepsilon = 0.0018$ in the following experiments. Under this perturbation value, the attack performance of DLA is better than that of the four



(a) MIM



(b) DLA

Fig. 10. Confusion matrix after different types attack.

traditional attack methods, and is 5.36% higher than that of MIM.

When the simulation conditions are the same, we record the time required to generate the adversarial examples corresponding to the test set by FGSM, BIM, PGD, MIM, DIM, DIFA, and DLA, and the average time cost of generating an

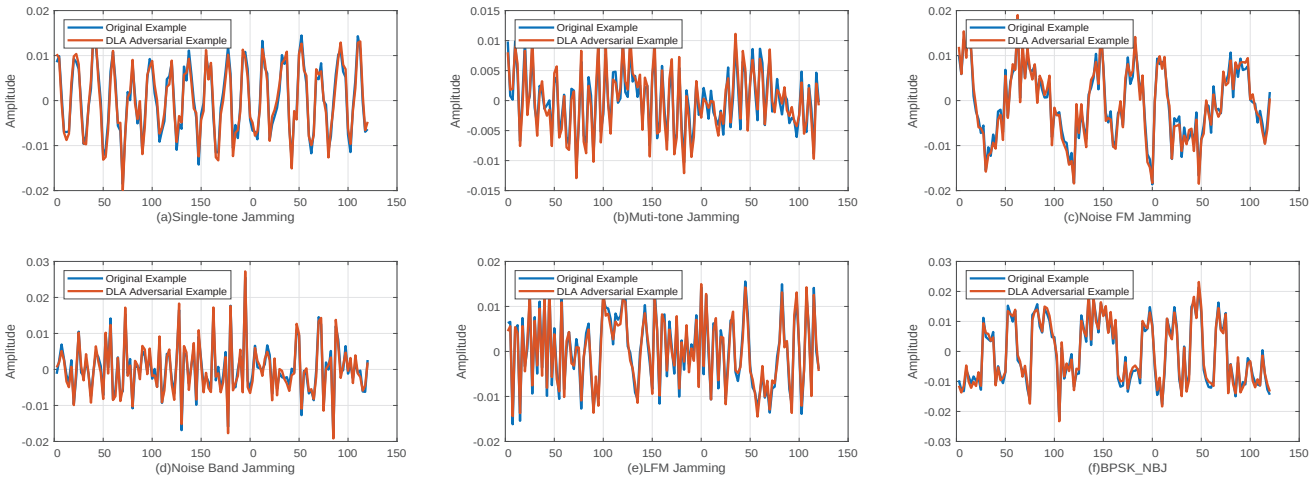


Fig. 11. Time domain waveform of different kinds of communication jamming before and after DLA.

example is 0.170 ms, 0.997 ms, 0.960 ms, 1.024 ms, 1.012 ms, 0.968 ms, and 1.826 ms, respectively. Therefore, the time cost of using BIM, PGD, MIM, DIM and DIFA to generate adversarial examples is basically the same, indicating that the time complexity of these algorithms is not much different. However, Fig. 8 and Fig. 9 show that the attack performance of the proposed DIM and DIFA is significantly better than the traditional methods. If the priority of attack success rate is higher than the time cost, DLA can be used to further improve the attack effect.

Using $\varepsilon = 0.0018$ and JNR = 6 dB, we use confusion matrix to compare the attack effect of MIM and DLA. The simulation results are shown in Fig. 10. Compared with the confusion matrix under the same perturbation and JNR in Fig. 3, the two attack methods make the recognition confidence matrix of the model more confused. From Fig. 10, we can see that compared with MIM, the accuracy of the recognition model for other classes of adversarial examples generated by DLA has decreased, except for Noise Band Jamming. This shows that the DLA is suitable for most communication jamming and has stronger attack performance.

C. Waveform Similarity

The waveform similarity can intuitively show the waveform similarity between the adversarial example and the original jamming signal, which is a further test of the performance of the attack algorithm. We can generate the time-domain waveform of the original jamming signal and the adversarial example through

$$S(t) = I\cos(2\pi ft) + Q\sin(2\pi ft), \tag{25}$$

where I is the in-phase component, Q is the orthogonal component, and f is the carrier frequency.

For $\varepsilon = 0.0018$ and JNR = 6 dB, we use DLA to generate adversarial examples. We select the original and adversarial examples of single-tone jamming, multi-tone jamming, noise FM jamming, noise band jamming, LFM jamming and BPSK_NBJ, and the time domain waveform is shown in Fig. 11. In Fig. 11, we show the time-domain waveforms of different communication jamming signals before and after being attacked by DLA. It can be seen that there is a strong waveform similarity between the adversarial example and the original jamming signal. Although the waveform before and after attack seems to be similar, the recognition model may divide them into different classes. For example, the waveform of BPSK_NBJ before and after attack is similar, but we can see from Fig. 10 (b) that the model classifies the jamming as noise FM jamming with 0.58 confidence.

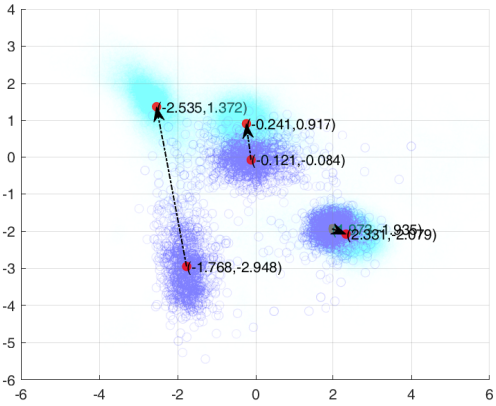


Fig. 12. Characteristic scatter plot of multi-tone jamming before and after DLA.

In order to analyze the internal characteristics of communication jamming before and after the attack, we choose the multi-tone jamming when JNR = 6 dB as the research object. The original example and the adversarial example of the jamming are input into the recognition model respectively and their feature mapping is output. We can obtain the feature scatter diagram as shown in Fig. 12. In Fig. 12, we show the aggregation of the first three feature points in the internal features of multi-tone jamming before and after DLA attack when JNR = 6 dB, and use the arrow to mark the change of the corresponding feature center. It can be seen that

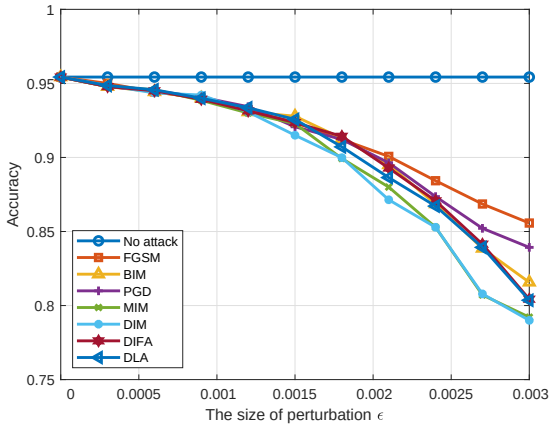


Fig. 13. Defense performance with different perturbations

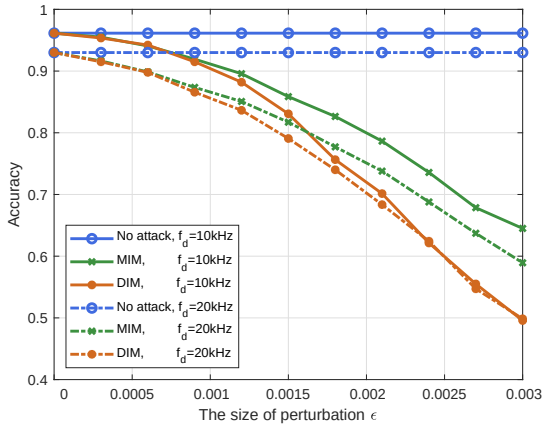


Fig. 14. Defense performance with different frequency shifts before transfer

after the attack, since the DLA algorithm generates feature confrontation within the communication jamming, the feature aggregation area obviously moves, which will lead to the wrong identification of the model.

D. Adversarial Defense

In the implementation of adversarial training, we first use the original training set examples and the proposed DLA algorithm to generate the adversarial examples of communication jamming for $\varepsilon = 0.0018$ and JNR = 6 dB. Then, we add these adversarial examples to the original training set examples as a whole training set, which is used to retrain ResNet to obtain jamming recognition model. In order to observe the robustness of defense model to adversarial examples, we study the influence of adversarial examples generated by different perturbations on the recognition accuracy of defense model. The simulation results are shown in Fig. 13. In Fig. 13, we show the recognition accuracy of the defense model obtained by adversarial training for different adversarial examples. It can be seen that, compared with Fig. 9, the recognition accuracy of the defense model for adversarial examples has significantly increased. For example, when $\varepsilon = 0.0018$, the recognition accuracy of the model for MIM increased from 0.67 to 0.90, an increase of 23%. Because the defense model in this section is trained by adversarial examples generated by DLA algorithm, the model has strong robustness to DLA attack algorithm. Through observation, we can find that this model also has a strong defense effect against DIFA, while the defense performance of DIM is the worst. Since in this paper, the proposed DIM, DIFA and DLA are label-based, feature-based, and double level attacks, respectively, and we use DLA for adversarial training, we speculate that the defense model has better defense effect for feature-based attacks than for label-based attacks.

E. Defense Transfer

From the above simulation results, it can be seen that among the four traditional attack methods of FGSM, BIM, PGD and MIM, MIM has the strongest attack. Among the three attack

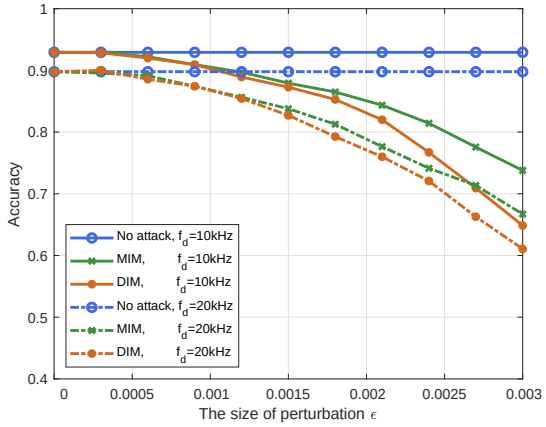


Fig. 15. Defense performance with different frequency shifts after transfer

methods proposed in this paper, DIM has the greatest threat to the defense model obtained by using DLA for adversarial training. Therefore, we choose MIM and DIM to test the influence of transfer learning on the accuracy of jamming recognition defense model.

Considering that the communication jamming received by the aircraft in different environments may have different Doppler shifts, we set the frequency shifts in the new environment to 10kHz and 20kHz respectively. In order to highlight the transfer effect of the defense model, we first use the communication jamming training recognition model with different frequency shifts, and use MIM and DIM to attack it. The results are shown in Fig. 14. From Fig. 14, we can see that with the increase of frequency shift, the recognition accuracy of the recognition model for the original examples and the adversarial examples is reduced, and the proposed DIM has stronger attack performance than MIM.

We transfer the parameters of the defense model to the recognition network in the new environment and freeze them, and then use the communication jamming with different frequency shifts to train the new recognition model. After training, the recognition accuracy of the new recognition model is shown in Fig. 15. It can be seen that the transfer

TABLE II
RECOGNITION ACCURACY OF THE MODEL BEFORE AND AFTER TRANSFER WHEN $\varepsilon = 0.0018$

Recognition Accuracy (%)		Attack Methods		
		No Attack	MIM	DIM
Before Transfer	$f_d = 10kHz$	96.14	82.64	75.64
	$f_d = 20kHz$	93.00	77.71	74.00
After Transfer	$f_d = 10kHz$	92.93	86.50	85.29
	$f_d = 20kHz$	89.79	81.27	79.27

of defense model improves the ability of models against adversarial attacks in new environments.

In order to analyze the performance changes of the recognition model more clearly, we take the recognition accuracy of the model when $\varepsilon = 0.0018$, as shown in Table II. In Table II, by comparing the recognition accuracy of the model under MIM and DIM attacks, we can find that the attack performance of DIM is significantly better than MIM. Although the recognition accuracy of the model for clean examples decreased after migration, the recognition accuracy of the model for adversarial examples increased significantly, showing defensive performance. When the Doppler frequency shift $f_d = 10kHz$, the transfer of defense knowledge increases the recognition accuracy of the model for MIM and DIM adversarial examples by 3.86% and 9.65%, respectively. When $f_d = 20kHz$, these two values are 3.56% and 5.27%, respectively. Therefore, transferring the knowledge of the existing defense model to the model in the new environment can effectively improve the defense performance of the new model.

VI. CONCLUSION

In this paper, we have studied the security problem of communication jamming intelligent recognition model in air traffic control. Based on the existing attack methods, we have proposed a dynamic iterative step to adjust the level of adversarial perturbation, and obtained class features to generate feature confrontation to design a new double level attack method by using the adversarial loss of feature layer and decision layer. In addition, we have used adversarial training to obtain a robust defense model, and used the transfer learning method to transfer the knowledge learned from the model to the jamming recognition models in other communication environments. Simulation results have shown that the proposed method has good attack and defense effect in the field of intelligent recognition of communication jamming, which is helpful to improve the safety and reliability of air traffic communication.

REFERENCES

[1] M. Liu, Z. Zhang, N. Zhao and Y. Chen, "Adversarial attacks on deep neural networks based modulation recognition," in *INFOCOM WKSHPS - IEEE Conf. Comput. Commun. Workshops*, 2022, pp. 1-6.

[2] Y. Wu, G. Ji, T. Wang, L. Qian, B. Lin and X. Shen, "Non-orthogonal multiple access assisted secure computation offloading via cooperative jamming", *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7751-7768, Jul. 2022.

[3] N. Zhao et al., "Secure transmission via joint precoding optimization for downlink MISO NOMA", *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7603-7615, Aug. 2019.

[4] H.-M. Wang and X. Zhang, "UAV secure downlink NOMA transmissions: A secure users oriented perspective", *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5732-5746, Sep. 2020.

[5] J. Johnston, Y. Li, M. Lops and X. Wang, "ADMM-Net for communication interference removal in stepped-frequency radar," *IEEE Trans. Signal Process.*, vol. 69, pp. 2818-2832, 2021.

[6] M. Liu, C. Liu, Y. Chen, Z. Yan and N. Zhao, "Radio frequency fingerprint collaborative intelligent blind identification for green radios," *IEEE Trans. Green Commun. Networking*, 2022, doi: 10.1109/TGCN.2022.3185045.

[7] H. Li, J. Luo and C. Liu, "Selfish bandit-based cognitive anti-jamming strategy for aeronautic swarm network in presence of multiple jammer," *IEEE Access*, vol. 7, pp. 30234-30243, 2019.

[8] L. Jia, Y. Xu, Y. Sun, S. Feng, L. Yu and A. Anpalagan, "A game-theoretic learning approach for anti-jamming dynamic spectrum access in dense wireless networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1646-1656, Feb. 2019.

[9] P. Eliardsson and P. Stenumgaard, "Artificial intelligence for automatic classification of unintentional electromagnetic interference in air traffic control communications," in *EMC Europe - Int. Symp. Electromagn. Compat.*, 2019, pp. 896-901.

[10] A. Nguyen, J. Yosinski and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427-436.

[11] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1-10.

[12] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 213-216, Feb. 2019.

[13] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 9, pp. 2805-2824, Sept. 2019.

[14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39-57.

[15] D. J. Miller, Z. Xiang and G. Kesidis, "Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks," in *Proc. IEEE*, vol. 108, no. 3, Mar. 2020, pp. 402-433.

[16] K. Wang, F. Li, C. -M. Chen, M. M. Hassan, J. Long and N. Kumar, "Interpreting adversarial examples and robustness for deep learning-based auto-driving systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9755-9764, Jul. 2022.

[17] A. Haydari, M. Zhang and C. -N. Chuah, "Adversarial attacks and defense in deep reinforcement learning (DRL)-based traffic signal controllers," *IEEE Open J. Intell. Transp. Syst. IEEE*, vol. 2, pp. 402-416, 2021.

[18] Y. Lin, H. Zhao, X. Ma, Y. Tu and M. Wang, "Adversarial attacks in modulation recognition with convolutional neural networks," *IEEE Trans. Reliab.*, vol. 70, no. 1, pp. 389-401, Mar. 2021.

[19] M. Liu, H. Zhang, Z. Liu and N. Zhao, "Attacking spectrum sensing with adversarial deep learning in cognitive radio-enabled internet of things," *IEEE Trans. Reliab.*, 2022, doi: 10.1109/TR.2022.3179491.

[20] I. Goodfellow, J. Shlens, and C. Szegedyn, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, Mar. 2015, pp. 189-199

[21] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Represent.*, May 2016, pp. 128-141.

[22] A. Madry, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, vol. 1, May 2018, pp. 1-23.

[23] Y. Dong and F. Liao, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mar. 2018, pp. 9185-9903.

[24] Y. Wang, M. Liu, J. Yang and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074-4077, Apr. 2019.

[25] B. Tang, Y. Tu, Z. Zhang and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713-15722, 2018.

[26] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cognit. Commun. Networking*, vol. 4, no. 3, pp. 433-445, Sept. 2018.
[27] T. J. O'Shea, T. Roy and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 168-179, Feb. 2018.
[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770-778.
[29] L. Huang, Y. Zhang, W. Pan, J. Chen, L. P. Qian and Y. Wu, "Visualizing deep learning-based radio modulation classifier," *IEEE Trans. Cognit. Commun. Networking*, vol. 7, no. 1, pp. 47-58, Mar. 2021.
[30] S. Sabour, Y. Cao, F. Faghri and D. J. Fleet, "Adversarial manipulation of deep representations," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[31] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 501-509.
[32] X. Xu, J. Zhang, Y. Li, Y. Wang, Y. Yang and H. T. Shen, "Adversarial attack against urban scene segmentation for autonomous vehicles," *IEEE Trans. Ind. Inf.*, vol. 17, no. 6, pp. 4117-4126, Jun. 2021.
[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.