applied sciences

MDPI

Article

Adversarial Example Detection and Restoration Defensive Framework for Signal Intelligent Recognition Networks

Chao Han, Ruoxi Qin, Linyuan Wang, Weijia Cui, Dongyang Li and Bin Yan *

Information Engineering College, Information Engineering University, Zhengzhou 450001, China
* Correspondence: ybspace@hotmail.com

Abstract: Deep learning-based automatic modulation recognition networks are susceptible to adversarial attacks, posing significant performance vulnerabilities. In response, we introduce a defense framework enriched by tailored autoencoder (AE) techniques. Our design features a detection AE that harnesses reconstruction errors and convolutional neural networks to discern deep features, employing thresholds from reconstruction error and Kullback–Leibler divergence to identify adversarial samples and their origin mechanisms. Additionally, a restoration AE with a multi-layered structure effectively restores adversarial samples generated via optimization methods, ensuring accurate classification. Tested rigorously on the RML2016.10a dataset, our framework proves robust against adversarial threats, presenting a versatile defense solution compatible with various deep learning models.

Keywords: adversarial examples; sample detection; sample restoration; autoencoder

check for updates

Citation: Han, C.; Qin, R.; Wang, L.; Cui, W.; Li, D.; Yan, B. Adversarial Example Detection and Restoration Defensive Framework for Signal Intelligent Recognition Networks. *Appl. Sci.* **2023**, *13*, 11880. https://doi.org/10.3390/app132111880

Academic Editors: Wenjia Li, Lei Chen and Yun Lin

Received: 28 September 2023
Revised: 21 October 2023
Accepted: 27 October 2023
Published: 30 October 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 Internet of Things (IoT) has rapidly advanced, leading to intelligent connectivity and massive data generation via numerous sensors [1,2]. Modulation signal recognition, a pivotal IoT technology, ensures physical security and has seen deep learning-based improvements in sensitivity and accuracy [3,4]. The adoption of convolutional neural networks (CNNs) and deep neural networks (DNNs) in signal recognition, as noted in [5], reduces reliance on prior knowledge, mitigates manual feature extraction subjectivity, and delivers enhanced recognition and generalization.

Deep learning models, particularly those developed in recent years, have exhibited vulnerability to adversarial attacks. While Szegedy [6] notably spotlighted that specific imperceptible perturbations can dramatically increase a network's prediction error, leading to image misclassification, it is crucial to contextualize this within a broader history of adversarial threats. The challenges associated with machine learning model security have been highlighted earlier, indicating risks beyond the contemporary deep learning models [7]. Moreover, adversarial attacks are rooted in the wider concept of "adversarial learning", a notion that predates the extensive study of neural network vulnerabilities. For instance, adversarial classifications were explored in foundational works like those by Dalvi et al. [8]. Outside the confines of machine learning, similar threats were considered in domains such as biometrics, as seen in the works of Volchikhin et al. [9].

These adversarial concerns are part of a broader landscape of influences, each with its historical lineage. The subsequent research, applications, and risks—such as those in autonomous driving [10]—address these vulnerabilities specifically within the realm of deep learning systems and gradient descent-based algorithms. Hence, while our focus is on these contemporary issues, it is essential to acknowledge that the challenge of adversarial attacks and the quest for solutions span a broader and more longstanding continuum in the realm of machine learning and artificial intelligence.

Following this historical backdrop of adversarial threats in machine learning and artificial intelligence, contemporary research on adversarial attacks has significantly amplified,

leading to the evolution of defense mechanisms [11,12]. Adversarial training, emerging as a primary countermeasure, has harnessed the power of data augmentation to fortify model robustness [13]. Another approach, defense distillation, focuses on attenuating the sensitivity to adversarial perturbations by training sequential deep networks [14].

The rise of adversarial attacks in deep learning has ushered in innovative detection methods such as reconstruction error and Kullback–Leibler (KL) divergence. Techniques harnessing reconstruction errors, like those using Generative Adversarial Networks (GANs) [15], focus on comparing original inputs with their regenerated counterparts to discern adversarial examples. Similarly, KL divergence [16,17], which measures the difference between two probability distributions, has proven instrumental in detecting classifier inconsistencies. Notably, in computer vision, GAN-based methodologies are emerging as frontrunners for restoring adversarial examples [15].

The unique nature of electromagnetic signal data sets them apart from traditional image or speech data. Though techniques like reconstruction error and KL divergence have proven valuable for image classification, their direct application to signal modulation classification often falls short. This gap emphasizes a pressing need for a method tailored to electromagnetic signals, harmoniously combining reconstruction error, KL divergence, and adversarial restoration for heightened safety and classification resilience.

This study focuses exclusively on enhancing adversarial example detection and restoration in electromagnetic signal modulation. We developed a framework using autoencoders (AEs) to detect and fix adversarial examples. This framework identifies whether a signal is adversarial and offers a reliable evaluation standard. Our design comprises two AE structures: one for detecting and another for restoring signals. The detection method can identify different types of attacks, namely gradient-based and optimization-based. Optimization attacks are subtle yet harmful. Our system can repair these tampered samples, ensuring they are recognized correctly. Importantly, our method can identify the attack's origin, whether from gradient-based or optimization tactics. Knowing the attack type is crucial for accurate defenses. Our framework can assist various models. Tests on the RML2016.10a dataset show that our method improves the signal recognition against adversarial threats.

The contributions of this paper are as follows:

- Integrating the metrics of reconstruction error and KL divergence, we propose a refined signal adversarial example detection framework based on AE. This approach uniquely captures the deep features of input samples, marking the first-time identification of the mechanisms behind adversarial examples, discerning whether they stem from gradient-based methods or optimization techniques.
- We design an adversarial examples restoration method based on AE. Through a carefully conceived AE architecture, we effectively restored the identified adversarial examples generated through optimization techniques, thereby enhancing the model's ability to recognize and recover from adversarial perturbations.
- All proposed frameworks and techniques were rigorously validated on the RML2016.10a dataset. Achieving a comprehensive detection rate of up to 88% for five classical adversarial examples, the model's recognition rate improved by over 40% after the recovery of adversarial examples. These results compellingly attest to the effectiveness of the introduced framework.

The remaining sections of this paper are structured as follows: Section 2 reviews related work. Section 3 details our research methods. Section 4 presents our proposed framework. Section 5 validates the framework with results and discussions. Section 6 concludes with contributions, findings, and future directions.

2. Related Work

The electromagnetic domain faces notable challenges from adversarial attacks. Sedeghi et al. [18] first identified adversarial impacts on signal recognition. Subsequent studies highlighted the reduction in communication efficiency in IoT systems due to these attacks [19] and vulnerabilities in RF fingerprinting [20–22]. Practical implications were

explored by Flowers et al. [23] who proposed BER as an evaluation metric. Research also delved into the influence of gradient-based attacks on modulation recognition [24] and the vulnerabilities in CNN-based device identification [25].

From these observations, it is evident that the electromagnetic domain is severely impacted by adversarial attacks, affecting communication processes, modulation recognition, and reliable transmission. This jeopardizes regular communication security and privacy protection. Adversarial examples have thus evolved into a widely acknowledged threat within the electromagnetic domain. The threat posed by adversarial examples to applications in the electromagnetic field is depicted in Figure 1.

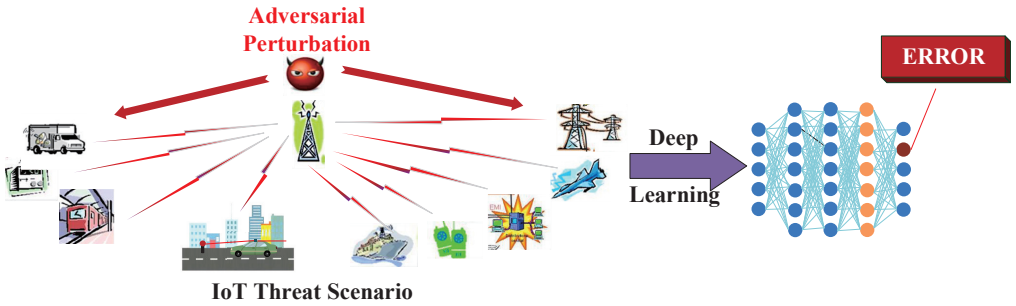


Figure 1. Threat scenarios of adversarial examples in the electromagnetic field.

To address the challenges posed by adversarial attacks, researchers in the signal domain have introduced various defense techniques. Examples include the adversarial training method proposed by Ren et al. [26], the multi-layer defense mechanisms by Tian et al. [27], and the stochastic smoothing approach by Kim et al. [28]. Notably, these techniques are largely aimed at enhancing model robustness. However, they often require retraining or modifications to the model's architecture, which could affect its performance and practicality.

Recent advancements in adversarial example detection have leveraged AE, with initial success in computer vision. Wójcik et al. [29] utilized intermediate layers of target networks to detect adversarial examples using AE. Tong et al. [30] integrated Gaussian, mean, and median filtering with AEs for image adversarial detection. Ye et al. [31] proposed the FADetector using feature knowledge. In the realm of RF signals, Silvija et al. [22] took a statistical approach toward adversarial detection, although with a constrained experimental scope.

In essence, while AE-based techniques show promise, direct applications from the image to signal domains are ineffective. There is an emerging need for AE structures tailored to signals and broader validation across modulated signal datasets.

3. Description of Research Methods

3.1. Automatic Modulation Recognition

Automatic modulation recognition is a method that employs machine learning techniques to classify and identify radio signals. In this task, deep learning can be used to autonomously learn the features of modulation signals and classify them. The advantage of deep learning is its ability to automatically extract features, eliminating the need for manually designed feature extraction algorithms. The input data to the network can be the raw signal data, and the deep neural network, acting as a combination of a feature extractor and classifier, can self-learn patterns in the data. Based on these patterns, it can classify the data, realizing an end-to-end modulation pattern-recognition process.

3.2. Adversarial Examples

Adversarial examples are samples that have been deliberately perturbed to induce misclassification in a model. These carefully designed samples exploit vulnerabilities inherent in the model to attack it, enticing the model to misclassify the sample into the wrong

class with high confidence, thereby compromising the model's security. The expression for generating adversarial examples is as follows:

$$\widetilde{x} = x + \delta, \|\delta\|_\infty \leq \varepsilon \tag{1}$$

where x represents the original sample, $\widetilde{x}$ denotes the adversarial example, δ is the perturbation added to the input sample, and ε represents the maximum allowable perturbation to the input, ensuring that the adversarial example does not deviate significantly from the original sample.

3.2.1. Threat Model

Adversarial attacks can primarily be categorized into two types based on their generation methods: gradient-based attacks and optimization-based attacks [32].

Gradient-Based Attacks: These attacks exploit the gradient information of the neural network's loss function to generate adversarial examples. Such methods typically compute the gradient of the loss with respect to the input data to craft the adversarial perturbations. The main idea is to adjust the original input in the direction that maximizes the model's loss, making the model more likely to misclassify the perturbed input. Examples of gradient-based attacks include FGSM, BIM, and PGD.

Optimization-Based Attacks: These attacks involve solving an optimization problem to find the smallest perturbation that can lead to misclassification. Such methods do not directly rely on the gradients of the neural network's loss function but focus on other optimization criteria to craft adversarial examples. The optimization process is generally more computationally intensive than gradient-based methods but can produce highly effective adversarial examples. The C&W attack and Deepfool algorithms are typical examples of optimization-based attacks.

In this work, we focus on the white-box attack scenario. In the white-box model, attackers have access to the complete structure and parameter information of the target model. They can directly generate adversarial examples on the target neural network, compute the true or approximated gradients of the real model, and adjust their attack methods based on the defense mechanisms and parameters they encounter.

3.2.2. Adversarial Attacks

In this section, we classify adversarial attacks into two main categories: gradient-based attacks and optimization-based attacks, providing a comprehensive overview of each method.

Gradient-Based Attacks

These attacks leverage the gradient information of the model's loss function with respect to the input data to craft adversarial examples.

FGSM: The Fast Gradient Sign Method (FGSM) is a non-targeted attack method that generates adversarial samples by constraining the L_∞ norm of the original samples [11]. It operates by moving in the gradient direction of the adversarial loss function $J(\theta, x, y)$ to maximize the loss. Specifically, the adversarial sample generation using FGSM can be expressed as:

$$\begin{cases} \tilde{x} = x + \rho \\ \rho = \varepsilon \cdot \text{sign}(\nabla_x J_\theta(x, l)) \end{cases} \tag{2}$$

where x denotes the original sample, l the target label, ρ the perturbation, and ε the maximum allowed perturbation. The constraint ensures that the magnitude of ρ in the L_∞ norm remains within ε.

BIM: Building on FGSM, the Basic Iterative Method (BIM) [33] enhances adversarial example generation by repeatedly adjusting the gradient. This iterative process crafts more potent adversarial examples with less discernible perturbations compared to FGSM.

PGD: Projected Gradient Descent (PGD) [12] is a more refined gradient-based method that aims to find the perturbations maximizing the model's loss function, given certain constraints. PGD optimizes the expected loss over potential perturbations in the input space, producing notably robust adversarial samples.

Optimization-Based Attacks

These attacks solve optimization problems to find the least perturbation required to misclassify a given sample.

C&W Attack: The C&W attack, proposed by Carlini and Wagner [34], crafts adversarial examples across different distance metrics. While it predominantly uses the L_2 norm, the method's optimization ensures that the perturbed input remains within a valid input space, leading to high-confidence adversarial examples.

Deepfool: Deepfool [35] is an optimization-based attack that operates under the L_2 norm. Its goal is to determine the minimal perturbation required to have an input sample misclassified by crossing the decision boundary to another class. In comparison to FGSM and BIM, Deepfool is particularly efficient in achieving high misclassification rates with minimal perturbations.

3.3. Reconstruction Error

Reconstruction error is a measure used to quantify the difference between the original version of data and their reconstructed version. For many models, especially AEs, the objective is to learn an encoding function and a decoding function such that the input data, after being encoded and subsequently decoded, can approximate their original form as closely as possible. Let us assume that we have an input data x and their corresponding reconstruction $\hat{x}$. The reconstruction error E is typically defined as the difference between them. For continuous data, the most common reconstruction error is the Mean Squared Error (MSE), which can be defined as:

$$E(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{3}$$

where N represents the dimensionality of the data. For discrete or categorical data, other types of error functions, like cross-entropy loss, can be employed. In any form, the reconstruction error offers us a means to quantify the discrepancy between the original data and their reconstructed version.

3.4. Kullback–Leibler Divergence

KL divergence is a measure used to quantify the relative entropy between two probability distributions. It provides us with a means to quantify the information lost when approximating one probability distribution (usually the true distribution) with another (usually the model distribution). Mathematically, for the discrete probability distributions P and Q, the KL divergence is defined as:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{4}$$

For continuous distributions, it can be expressed as:

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \tag{5}$$

where $p(x)$ and $q(x)$ are the probability density functions of distributions P and Q, respectively. It is important to note that KL divergence is not symmetric, meaning that $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$. Thus, it is not a true distance metric, but offers us a powerful tool for measuring the similarity or difference between two distributions.

3.5. Adversarial Example Restoration

Adversarial example restoration is a process whose goal is to recover the original, unperturbed data from the tampered adversarial example. Given a known adversarial example x_{adv}, our objective is to find a close, untampered data point. This can be represented as an optimization problem where we attempt to minimize some distance measure between x_{adv} and its corresponding original sample x. A common approach is to use the Euclidean distance as the metric and recover through the following optimization problem:

$$\min_{x'} \|x_{adv} - x'\|_2^2 \tag{6}$$

here, x' is the unperturbed data sample we are attempting to recover. The crux of the restoration lies in finding an efficient method or algorithm to solve the above optimization problem, yielding x', which should be very close to the original sample x and distinctly different from the adversarial example x_{adv}.

4. The Proposed Detection and Restoration Framework

Our designed framework focuses on detecting and restoring adversarial examples for the automatic modulation classification model. The framework workflow begins with detection using the reconstruction error, proceeding to KL divergence-based detection if the error is below a threshold. If the reconstruction error surpasses a predefined limit, the sample is flagged as an adversarial example; otherwise, it is deemed clean and sent for classification. KL divergence measures the difference between output probability distributions of the sample and its reconstructed version, determining adversarial examples produced via optimization. Those exceeding the KL divergence threshold are restored before classification, while others are directly classified. We proposed two novel AE designs: one for detection, leveraging a hybrid encoder for precise feature extraction, and another for restoration, employing an intricate decoder to revert adversarial examples to their original state. The detailed architecture and workflow are illustrated in Figure 2.

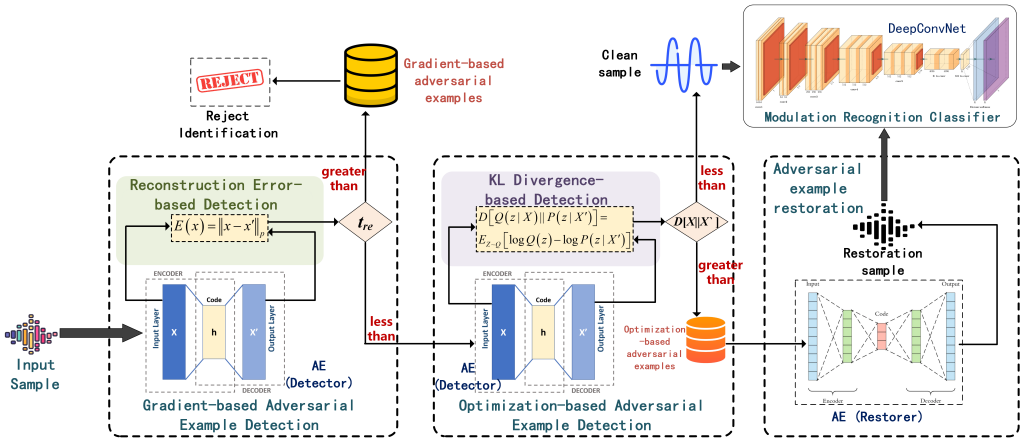


Figure 2. Flowchart of the proposed adversarial example detection and restoration framework for automatic modulation classification.

4.1. Detector Design

The encoder designed in this paper for detection is a hybrid AE that combines both CNN and Long Short-Term Memory (LSTM) networks. The CNN component is primarily employed to process input data and extract their spatial features. Once the input data are processed by the convolutional encoder, the original input is fed into an LSTM layer. After both the convolutional encoder and LSTM layers have processed the data, their outputs are concatenated along the channel dimension. The concatenated data are then passed to the decoder, which consists of several transposed convolutional layers. These transposed convolutional layers can be viewed as the inverse operations of the convolutional layers

in the encoder. They gradually increase the spatial dimensions of the data and decrease the number of channels to restore the shape of the original input data. The AE structure designed for the detector is illustrated in Figure 3.

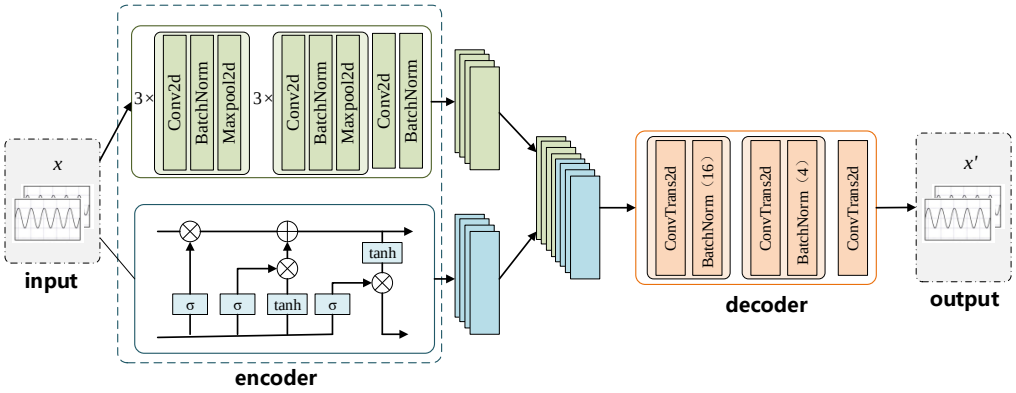


Figure 3. Designed detection AE.

As evident from Figure 3, the encoder segment comprises both a convolutional encoder and an LSTM layer. The convolutional encoder is responsible for extracting spatial features from the input data, while the LSTM layer handles the sequential nature of the input data. The outputs of these two components are then concatenated, forming a richer representation that encompasses both the spatial and sequential features of the input data. This design holds potential positive implications for adversarial example detection, particularly in terms of its capacity to learn complex patterns. The CNN and LSTM structures in the encoder part can learn features from both spatial and temporal dimensions, enabling the AE to capture more intricate patterns, thus enhancing its capability to detect adversarial examples.

4.2. Restorer Design

The designed AE for restoration also comprises two parts: an encoder and a decoder. The structure of the encoder part follows a standard CNN architecture. The decoder part is primarily responsible for reconstructing the output of the encoder back to the shape of the original input. The design of the decoder is closely related to that of the encoder, but it incorporates batch normalization and dropout layers at each step. Batch normalization accelerates training and enhances model generalization, while dropout prevents overfitting and increases model robustness. The AE structure tailored for restoration is illustrated in Figure 4.

The primary role of the decoder in the AE designed for restoration is to learn how to recover the original high-dimensional data from the hidden low-dimensional representation. Through this process, the decoder acquires the high-level features of the data and is capable of generating new data that resemble the input data. By utilizing convolutional transpose layers, the decoder can convert the low-resolution features from the encoder output into high-resolution outputs. In this process, the decoder learns how to recover fine-grained details from low-resolution features. With multiple levels of upsampling, convolution, batch normalization, and dropout operations, the decoder can capture features at different scales, enabling it to better restore the original data. Therefore, the design of this decoder allows the AE to effectively learn and generate high-quality data that closely match the input, facilitating the restoration of adversarial signal samples to their original forms.

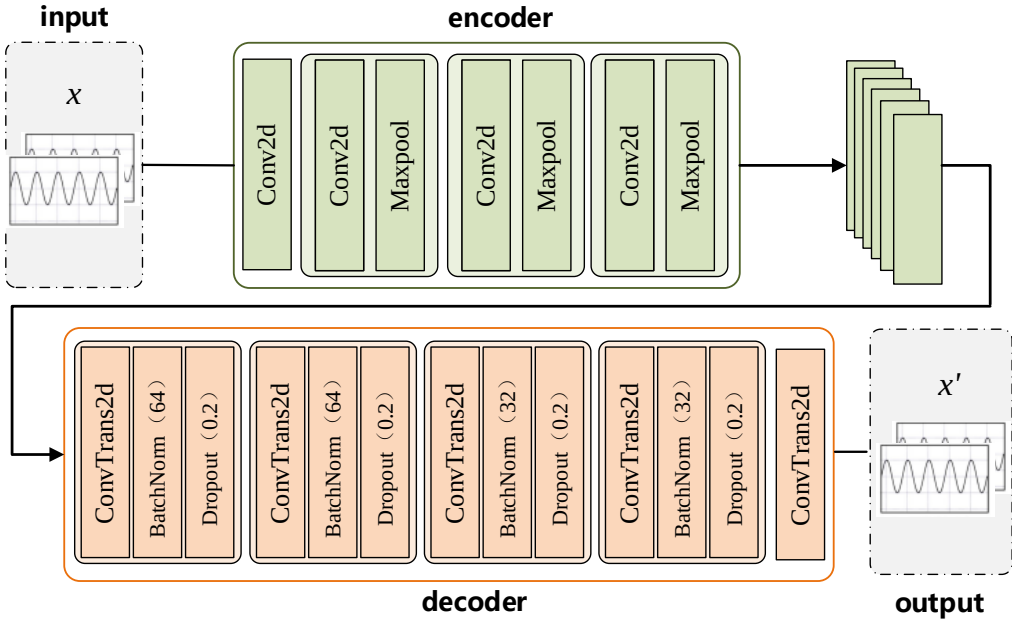


Figure 4. Designed restoration AE.

4.3. Detection Method Based on Reconstruction Error

Compared to supervised learning methods, the reconstruction-based detection approach eliminates the need for adversarial examples. This is because this approach does not require labeled anomalous data or prior knowledge of the features of anomalous data. Moreover, this method can capture implicit anomalies that have not appeared in the training data, which contributes to enhancing the generalization capability of the detector.

Our framework utilizes an AE-based approach for detecting adversarial examples, using only regular samples during modeling. Adversarial attacks introduce minor alterations to original data, challenging accurate reconstructions by machine learning models. To address this, we use an AE to reconstruct the input and compute the reconstruction error. If this error exceeds a threshold, the data are identified as an adversarial example. The AE, represented as $ae = d \circ e$, comprises an encoder, which compresses input data into a low-dimensional representation, and a decoder that tries to reconstruct the original data from this encoded format.

When dealing with signal datasets consisting of In-phase (I) and quadrature (Q), considering their characteristics, we propose a method that combines reconstruction errors with signal features to enhance the performance of the AE. Specifically, we focus not only on the reconstruction error but also introduce a signal feature measurement to better capture the subtle perturbations in I/Q signals. To enable the model to better capture the properties of I/Q signals, we introduce an enhanced loss function that optimizes the reconstruction error separately for the I component and Q component. Specifically, our loss function is defined as follows:

$$L(X_{\text{train}}) = \frac{1}{|X_{\text{train}}|} \sum_{x \in X_{\text{train}}} \left(\|x_I - ae(x)_I\|_2^2 + \|x_Q - ae(x)_Q\|_2^2 \right) \tag{7}$$

where x_I and x_Q represent the I and Q components of the input data, and $ae(x)_I$ and $ae(x)_Q$ represent the reconstructed outputs of the AE for the respective components. This decomposition allows us to calculate the reconstruction error separately for each component and then compute a weighted average of the errors from both components. This ensures that the model is optimized for both the I and Q components.

Given an input sample x, the reconstruction error, derived from the I and Q channels, is defined as:

$$E(x) = \frac{1}{N}\sum_{n=1}^{N}\left(\|x_{I,n} - ae(x)_{I,n}\|_2^2 + \|x_{Q,n} - ae(x)_{Q,n}\|_2^2\right) \tag{8}$$

where N is the sample count per channel. This function gauges discrepancies in both channels for each sample, ensuring comprehensive signal representation.

The AE learning comprises two stages:

- Encoder: Produces a compressed data representation.
- Decoder: Reconstructs the original data from this representation.

For adversarial examples, a high reconstruction error is anticipated, but it is minimal for regular test samples. A threshold t_{re} distinguishes normal samples. It is set as low as possible to detect subtle adversarial perturbations, but not so low as to misclassify regular samples. Typically, t_{re} is set based on the error distribution within a validation set. The method involves selecting a threshold. This threshold is determined by minimizing the proportion of samples whose reconstruction errors exceed it, relative to the total number of samples. This method can avoid overfitting and issues with model instability. Additionally, the threshold can be adjusted according to the specific needs of the application. The algorithmic process is outlined in Algorithm 1.

Algorithm 1: Reconstruction error-based adversarial detection.

Data: Training dataset X_{train}, Test sample x, Validation dataset $X_{\text{validation}}$
Result: Boolean indicating if x is adversarial
Initialize $ae = d \circ e$
while *not converged* **do**
  $X_{\text{reconstructed}} \leftarrow ae(X_{\text{train}})$
  $L = \frac{1}{|X_{\text{train}}|}\sum \|x - X_{\text{reconstructed}}\|_2$
  Update ae parameters using backpropagation
end
Compute reconstruction errors for $X_{\text{validation}}$ using ae
$t_{\text{re}} \leftarrow$ threshold based on error distribution of $X_{\text{validation}}$
$E(x) \leftarrow \|x - ae(x)\|_p$
if $E(x) > t_{re}$ **then**
  return True `// Sample x is adversarial`
end
else
  return False `// Sample x is not adversarial`
end

4.4. Detection Method Based on KL Divergence

Signal processing involves the challenge of distinguishing genuine signals from adversarial ones. Due to the inherent complexities, conventional methods often fall short. We propose an innovative solution, harnessing KL divergence and deep learning, to detect adversarial signals.

Given a genuine signal S, an ideal classifier outputs a probability distribution $P_C(S)$. Adversarial perturbation shifts this to $P_C(S_{adv})$, with the divergence quantified as:

$$D_{KL}(P_C(S)||P_C(S_{adv})) = \sum_i P_C(S)_i \log \frac{P_C(S)_i}{P_C(S_{adv})_i} \tag{9}$$

Using an AE, signals transition to a latent space representation z, followed by a classifier yielding $Q(z|S)$. The KL divergence between this and the genuine signal distribution $P(z|S)$ is our key metric:

$$D[Q(z|S)\|P(z|S)] = E_{Z\sim Q}[\log Q(Z) - \log P(z|S)] \tag{10}$$

We further refine the result by integrating the classifier parameters, θ_C, resulting in:

$$D[Q(z|S,\theta_C)\|P(z|S)] = E_{Z\sim Q}[\log Q(Z,\theta_C) - \log P(z|S)] \tag{11}$$

This methodology underscores the classifier's probabilistic boundary in the latent space, using KL divergence to spotlight adversarial perturbations. Using an appropriate prior distribution, typically Gaussian, enhances the method's robustness. This synergy between classifier probabilities and latent space representation empowers detection of adversarial signal examples, merging mathematical rigor with practical efficacy. The algorithmic process is outlined in Algorithm 2.

Algorithm 2: Detection method based on kl divergence with classifier integration.

Data: Training sample x_i, testing sample y_j, initial sample size N, random sample size M, classifier C

Result: Classification results of the samples

1. Assume the original data follows a Gaussian distribution $P(z)$.
2. Use $P(z|X)$ to approximate $P(z)$, where X is the output from the classifier $C(x)$.
3. Introduce another distribution $Q(z|X)$ to approximate $P(z|X)$.
4. Calculate KL divergence:

$$D[Q(z|X)\|P(z|X)] = E_{Z\sim Q}[\log Q(Z) - \log P(z|X)]$$

5. Set the optimal threshold t_{re} based on validation set.
6. **for** $j = 1$ **to** M **do**
   a. Compute $D[Q\|P]$ for y_j.
   b. **if** $D[Q\|P] < t_{re}$ **then**
      y_j is labeled as a normal sample.
   else
      y_j is labeled as an adversarial example.
   end
13. **end**

4.5. Adversarial Example Restorer Method

The detection method based on the reconstruction error has shown efficacy against gradient-based adversarial examples in signal processing. Gradient-based methods significantly alter a signal's structure, causing adversarial waveforms to diverge from clean samples. This divergence makes it challenging to restore the original structure.

Despite the minimal waveform differences between optimized CW, Deepfool adversarial examples, and their original counterparts, neural networks can discern in high-dimensional spaces what human eyes cannot. When adversarial examples traverse the AE, their latent variables adjust to resemble clean samples. In the low-dimensional latent space, perturbations dissipate, enabling the AE to reconstruct normal signals from adversarial inputs. This restoration allows convolutional neural networks to accurately classify samples. The restoration layers are represented as:

$$x_{h_{i+1}} = \mathrm{T}\left(W_{h_i} x_{h_i} + b_{h_i}\right) \tag{12}$$

where the matrix W_{h_i} and vector b_{h_i} represent the weights and biases of the hidden layer h_i, respectively. Their dimensions depend on the size of the input data, such that if the input data have dimensions m, then $W_{h_i} \in \mathcal{R}^{m \times r}$ and $b_{h_i} \in \mathcal{R}^m$. Conversely, if the input data dimensions are r, then $W_{h_i} \in \mathcal{R}^{r \times m}$ and $b_{h_i} \in \mathcal{R}^r$. Within this context, x denotes the hidden neurons, and the function $T(\bullet)$ acts as the restoration function. An important aspect to note is that the initial input x_{h_o} to the restorer corresponds to h_t. Building on these details, the computation for the output layer can be represented by the subsequent equation:

$$\tilde{\phi} = T(W_r x_h + b_r) \tag{13}$$

where $W_r \in \mathcal{R}^{r \times m}$ and $b_r \in \mathcal{R}^m$ are the weight matrix and bias of the final hidden layer, respectively.

The encoder of the AE maps the two-dimensional I/Q signal to a high-dimensional feature representation z of size $m \times n$ using multiple fully-connected layers:

$$z = f\left(W' f(Wx + b) + b'\right) \tag{14}$$

where x is the input signal, W, b, W', and b' are weight matrices and bias vectors for respective layers, and f is the nonlinear activation function.

The decoder, similar in structure, aims to reconstruct the input from the latent space z and feeds the output to a classifier.

$$\hat{x} = g\left(W''' g\left(W'' z + b''\right) + b'''\right) \tag{15}$$

where z is the output of the encoder, W'' and b'' represent the weight matrix and bias vector of a particular fully connected layer in the decoder, and W''' and b''' correspond to the subsequent fully-connected layer in the decoder. The function g is a nonlinear activation function. Ultimately, $\hat{x}$ is the output of the decoder, representing the reconstructed data.

An ideal decoder, in operation, should not introduce significant changes to normal samples; for adversarial examples, it should induce sufficient alterations to revert them to normal samples. However, these modifications should not exceed the divergence distribution range of normal samples. On this premise, our framework aims to enhance the classification accuracy of adversarial examples while retaining the accuracy for normal samples unchanged. To revert adversarial examples back to normal samples, we employ an AE as the decoder. Leveraging the latent variable characteristics of AE in generative models, we can amplify the divergence of the latent variables for adversarial examples, reconstructing them back to normal samples. The formula for divergence minimization can be expressed as:

$$\begin{aligned} &\min D_{KL}(Q_\phi(z \mid X) \| P_\theta(z)) \\ &\text{s.t. } D_{KI} \leq \eta \end{aligned} \tag{16}$$

where D_{KL} denotes the KL divergence, $Q_\phi(z \mid X)$ represents the posterior distribution of the observed data, $P_\theta(z)$ is the prior distribution, and η is a pre-defined threshold representing the upper limit of the divergence. Using this approach, and capitalizing on the latent layer properties of the AE, we can, in a controlled manner, augment the divergence of adversarial examples and effectively restore them to their original state as normal samples. Furthermore, we employ the stochastic gradient ascent optimization technique to ensure the divergence of the adversarial examples remains within a certain range, preventing it from exceeding the divergence of normal samples. The restoration methodology for samples is detailed as shown in Algorithm 3.

Algorithm 3: Adversarial example restoration.

procedure RESTORATION(x, AE)
  Initialize: η, *MaxIter*, lr
  $iter \leftarrow 0$
  $z \leftarrow \text{Encoder}(AE, x)$
  $x_{hat} \leftarrow \text{Decoder}(AE, z)$
  $D_{KL} \leftarrow$ Compute_KL_Divergence$(Q_\phi(z|x), P_\theta(z))$ **while** $D_{KL} > \eta$ and $iter < MaxIter$ **do**
    end
    $z \leftarrow z + lr \times \text{GradientAscent}(D_{KL}, z)$
  $x_{hat} \leftarrow \text{Decoder}(AE, z)$
  $D_{KL} \leftarrow$ Compute_KL_Divergence$(Q_\phi(z|x_{hat}), P_\theta(z))$
  $iter \leftarrow iter + 1$
  return x_{hat}
end procedure
procedure COMPUTE_KL_DIVERGENCE(Q, P)
  return KL divergence(Q, P)
end procedure

5. Results and Discussion

5.1. Datasets

This paper employs the RML2016.10a dataset [5] for algorithm validation. The RML2016.10a dataset is a renowned public dataset, frequently adopted for machine learning investigations within the wireless signal modulation classification domain. Introduced by the DeepSig group in 2016, this dataset comprises over a million samples of radio signals. These signals are I/Q-recorded, where "I/Q" stands for in-phase and quadrature. I and Q components represent the real and imaginary parts of the complex samples of a signal, respectively, which are essential for representing the amplitude and phase of a radio signal. For our study, the dataset was partitioned into training, testing, and validation sets with a distribution ratio of 0.8, 0.1, and 0.1.

5.2. Classifier Model

The deep learning classifier we utilized is the DeepConvNet, which is a high-performance model capable of classifying the 11 modulation signals in the RML2016.10a dataset. It serves as a standard benchmark model. The DeepConvNet architecture comprises four convolutional layers, each followed by batch normalization and ReLU activation functions. Additionally, two of these convolutional layers apply max pooling operations to downsample the inputs. The channel sizes across these layers increase from 2 to 256. The model is concluded with a fully connected part that flattens the tensor outputs from the convolutional layers, passing through a linear layer with dimensions ranging from $256 \times 1 \times 16$ to 512, followed by a ReLU activation layer and a dropout layer for regularization. The final linear layer reduces the size to the number of classes. During the forward pass, the input tensor passes through each of these layers sequentially.

5.3. Comparative Experiments on the Degree of Difference in Reconstruction Errors under Different Paradigms

In this experimental section, we will utilize adversarial examples as input to the detection AE and simultaneously obtain their reconstructed samples. Subsequently, we will calculate errors based on different norms to identify the most suitable norm for reconstructing error analysis. The optimal reconstruction error norm should differentiate between original and adversarial examples as much as possible. In other words, the recon-

struction error for original and adversarial examples should be maximized to the greatest extent possible.

In Table 1, we have documented the reconstruction error values based on L_0, L_1, L_2, and $|x|_\infty$ norms. The experimental results indicate that the reconstruction error based on the L_0 norm is nearly 1 for both the original samples and all adversarial examples, making it unsuitable as a sample detection threshold benchmark. The threshold selection based on the L_1 norm is also not a good reference point.

Table 1. Reconstruction errors based on different norms.

Norm	Original Sample	FGSM (ε = 0.15)	BIM (ε = 0.15)	PGD (ε = 0.15)	CW	Deepfool
L_0	0.9999	1.0	1.0	1.0	1.0	1.0
L_1	0.0387	0.0264	0.0327	0.0311	0.0387	0.0380
L_2	0.0021	0.0012	0.0016	0.0015	0.0022	0.0021
L_∞	0.0530	0.0370	0.0449	0.0430	0.0530	0.0520

The more suitable norms are the L_2 norm and $|x|_\infty$. Under these two norm benchmarks, there is a significant difference in the reconstruction error between the original samples and the three gradient-based adversarial examples, making them suitable as sample discrimination criteria. Under the L_2 norm, the reconstruction error for the original sample is 0.0021, differing by 33% from the 0.0012 error of the FGSM-based adversarial example. In the $|x|_\infty$ measurement, the two error values are 0.0530 and 0.0370, differing by 31%. Notably, regardless of the norm used for reconstruction error measurement, both types of optimization-based attacks, CW and Deepfool, have the same reconstruction error as the original sample. It can be observed that optimization-based attacks are very close to the original sample in explicit representation, making them undetectable by the reconstruction error-based detection method. Therefore, other methods are needed to detect adversarial examples generated by optimization-based attacks. Considering that $|x|_\infty$ reflects the maximum value in a vector, it ensures that the perturbation of each element of the signal does not exceed a specific upper limit. Thus, we use $|x|_\infty$ as a scale for measuring the reconstruction error. The detection rates under various reconstruction error thresholds based on $|x|_\infty$ are shown in Table 2.

Table 2. Detection rates based on $\|x\|_\infty$ under various reconstruction error thresholds.

Sample Type	0.0005	0.0006	0.0007	0.0008	0.0009	0.001	0.002	0.005	0.01
Original	0.7064	0.3845	0.2918	0.1773	0.1082	0.0727	0.0009	0.0	0.0
FGSM	0.9036	0.8036	0.6609	0.5573	0.4373	0.3445	0.0109	0.0	0.0
PGD	0.8500	0.7300	0.5836	0.4409	0.3155	0.2191	0.0018	0.0	0.0
CW	0.6873	0.4709	0.2873	0.1727	0.1055	0.0699	0.0018	0.0	0.0

Based on the analysis above, we conducted comparative experiments for the adversarial example threshold detection in Table 2. As shown in the table, both the FGSM and PGD adversarial examples were generated under a perturbation strength of 0.15. We tested thresholds from 0.0005 to 0.001 with a step size of 0.0001, and also conducted detection experiments with larger thresholds such as 0.002, 0.005, and 0.01. The experimental results show that when the threshold is set very low, such as 0.0005, although the detection rate of adversarial examples is high, the false positive rate of the detector is high as well, with about 70% of the original samples being identified as adversarial examples. As the threshold gradually increases, the false positive rate decreases, and the detection rates for the FGSM and PGD adversarial examples also decrease accordingly. However, at the threshold of 0.0006, the reconstruction error based on $|x|_\infty$ can effectively be used to detect gradient-based adversarial examples. At this point, the false positive rate for original sam-

ples is around 38%, but the detection rates for the FGSM and PGD adversarial examples can reach up to 80% and 73%, respectively, making this threshold setting optimal. We also tested C&W adversarial examples and found that their detection is consistent with the original samples, demonstrating the subtlety of adversarial examples generated by optimization methods. When the threshold continues to increase, the detector becomes largely ineffective. Hence, appropriate norm measurements and threshold selection are crucial elements in detecting adversarial examples.

5.4. Detection Experiments for Various Attacks Based on Reconstruction Error

From Figure 5, it can be observed that under the same perturbation strength, the detection rates of adversarial examples generated by BIM and PGD attacks are lower than those generated by FGSM attacks. For instance, when $\varepsilon = 0.15$, the detection rates of adversarial examples generated by BIM and PGD attacks using the reconstruction error-based detection method are 65.09% and 73.45%, respectively, which are lower than the detection rate for adversarial examples generated by FGSM attacks. Since FGSM is a one-step attack and is more direct and coarse in generating adversarial examples compared to the iterative attacks of BIM and PGD, it causes more significant disruption to the sample distribution, making it more prone to detection as adversarial examples.

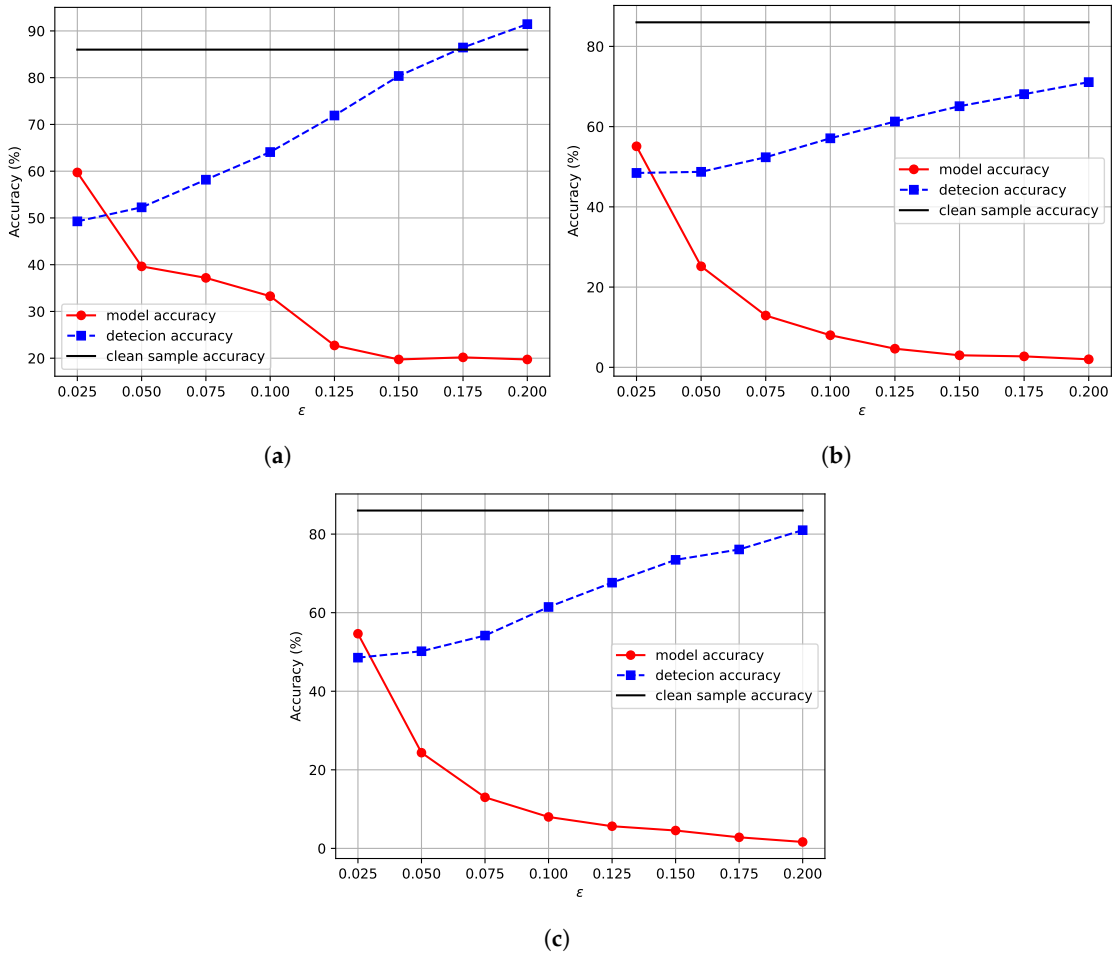


Figure 5. Detection accuracy against multiple adversarial attacks. (**a**) FGSM; (**b**) BIM; (**c**) PGD.

In this experiment, the reconstruction error is used to compare the differences between original samples and reconstructed samples generated by the AE, as well as between original samples and adversarial examples. The AE is an unsupervised machine learning model that aims to preserve the information of the original input as much as possible during the compression and decompression process. Therefore, if the AE is properly trained, the

generated reconstructed samples should be very close to the original samples in terms of visual appearance and features, resulting in a low reconstruction error.

However, adversarial examples are usually designed to make subtle but crucial modifications to the original samples, deceiving the machine learning model without being noticeable to the human eye. Although adversarial examples may appear very similar to the original samples visually, these subtle changes can lead to larger errors when the AE attempts to reconstruct the adversarial examples. This is because these changes may cause the adversarial examples to not accurately map to the position of the original samples in the latent space. Therefore, the reconstruction error demonstrates the ability to differentiate between original and adversarial examples. By setting a reasonable reconstruction error threshold, we can to some extent distinguish between original and adversarial examples. In general, if the reconstruction error of a sample exceeds this threshold when compared to the original sample, we can consider that sample to be an adversarial one. This finding provides an effective method for detecting adversarial examples.

5.5. Detection Experiments for Various Attacks Based on KL Divergence

While human eyes cannot discern the differences between normal samples and adversarial examples, in the high-dimensional space, neural networks can distinguish them with a high probability. To restore adversarial examples closer to normal samples, thus enabling the classifier to classify correctly, adversarial examples undergo transformations when passing through the AE. Disturbances are eliminated in the low-dimensional latent space. The divergence of adversarial examples through the AE is then altered, achieving the reconstruction of regular signals.

Figure 6 showcases the differences in KL divergence values between adversarial examples generated by various attacks and the original samples. The term "frequency" in the figure refers to the number of samples falling within specific KL divergence intervals. From Figure 6, gradient-based attacks like FGSM, PGD, and BIM yield adversarial examples with a significant overlap in KL divergence value distribution with the original samples. Especially, PGD-generated adversarial examples concentrate in almost the same KL divergence value range as the original samples. By contrast, optimization-based attacks such as CW and Deepfool produce adversarial examples with discernible KL divergence value distribution differences from the original samples. Gradient method-based adversarial examples might show apparent waveform differences from the original signal since they are adjusted in the loss gradient direction of the model. Although humans can easily observe this difference, it might not be the primary feature that the AE emphasizes. Hence, the AE might face challenges in recovery and may only reconstruct waveforms resembling adversarial examples. Consequently, calculating KL divergence values for samples proves more apt for detecting adversarial examples generated by optimization-based attacks than those generated by gradient-based attacks.

The method based on KL divergence is utilized for adversarial example detection evaluation. To verify its efficacy, two notable adversarial attacks based on optimization techniques, namely CW and Deepfool, were chosen, and the adversarial examples generated by them were tested. As shown in Figure 7, the KL divergence thresholds were set in a range, with the values 0.00001, 0.0001, 0.0005, 0.002, 0.01, 0.05, and 0.2. Under these thresholds, the observed detection rates for adversarial examples from CW ranged from 6.09% to 91.81% and for Deepfool from 5.72% to 98.63%. The false-positive rates, representing genuine samples incorrectly classified as adversarial, for CW varied from 0.07% to 14.46%, and for Deepfool ranged between 0.04% and 14.90%.

Analyzing the data, as the threshold decreased, the detection rates of the adversarial examples generated by both CW and Deepfool witnessed a considerable decline. Specifically, at the highest threshold of 0.2, the detection rates for CW and Deepfool were 6.09% and 5.72%, respectively, indicating a low detection performance. However, at the lowest threshold of 0.00001, the rates reached 91.81% and 98.63%, respectively, showcasing a significant improvement in detection capabilities. Furthermore, it is evident that, although

the detection rate of Deepfool remained consistently higher than CW's across all thresholds, their overall trends were largely analogous. This highlights the effectiveness of KL divergence as a detection measure for both types of attack methods.

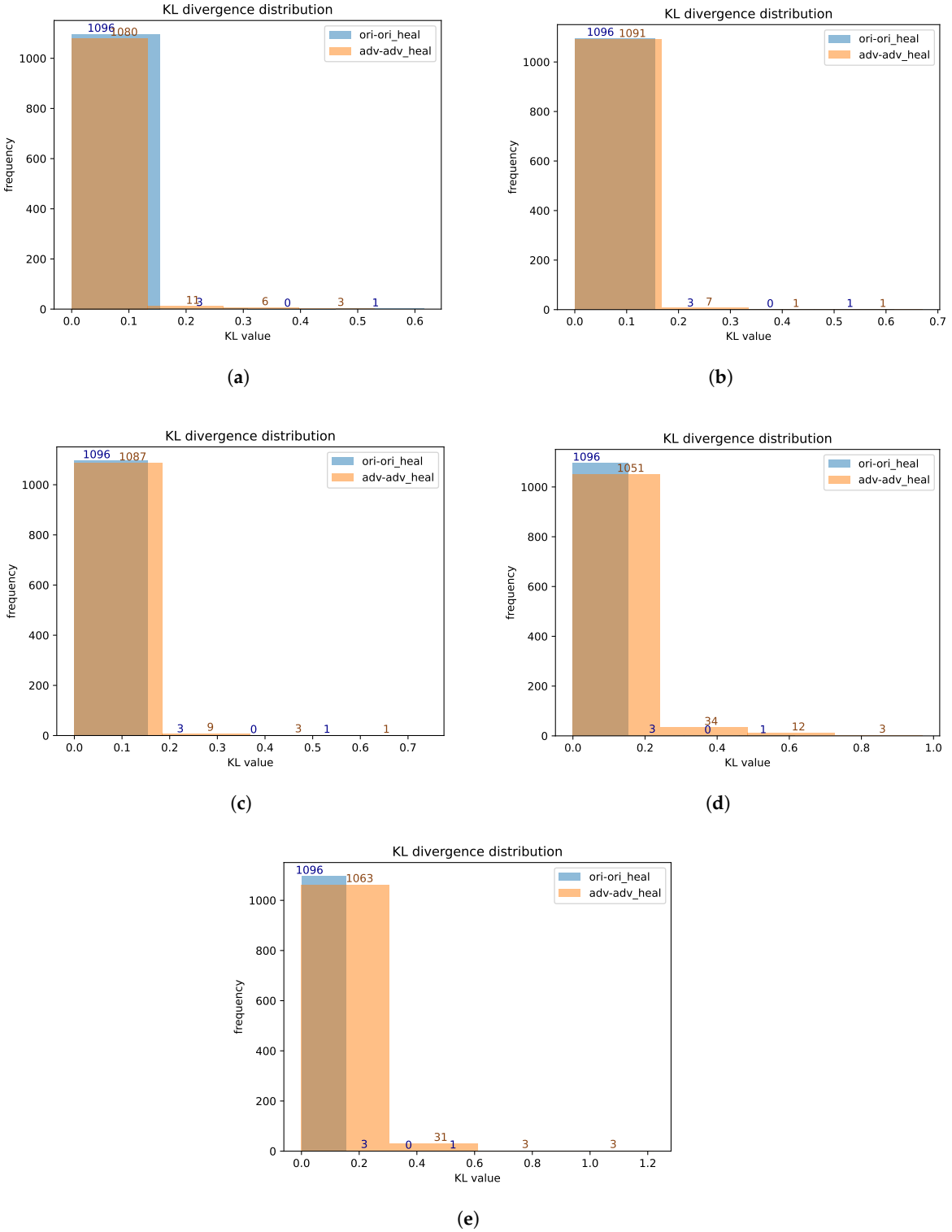


Figure 6. Multiple adversarial example detection based on Kullback–Leibler divergence. (**a**) FGSM; (**b**) PGD; (**c**) BIM; (**d**) CW; (**e**) Deepfool.

Moreover, the false-positive rates are equally important for understanding the trade-off between detection performance and misclassification of genuine samples. While the

detection rates are higher at lower thresholds, the false-positive rates also tend to increase, indicating a trade-off that needs consideration when setting the threshold.

In this experiment, KL divergence is employed to compare the differences between original samples and the reconstructed samples generated by the AE, as well as the differences between original samples and adversarial examples. The observed results reveal that the KL divergence between original samples and reconstructed samples is low, indicating that the AE effectively learns and replicates the probability distribution of the original samples. Consequently, the proximity in probability space between original and reconstructed samples is evident.

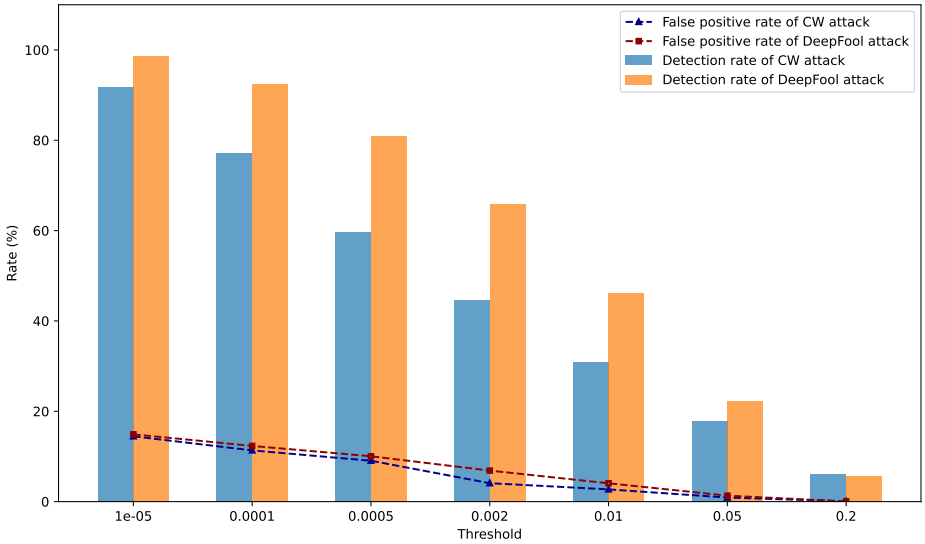


Figure 7. Detection rates for CW and Deepfool adversarial examples under different KL divergence thresholds.

However, for adversarial examples, the KL divergence is notably higher. This discrepancy may arise due to the intentional manipulation of features in adversarial examples during their design to deceive machine learning models. These adjustments are made to maintain a striking similarity to the original samples in human perception while inducing significant discrepancies in machine learning models. Hence, the gap between the probability distributions of adversarial examples and the original samples might widen.

Given this behavior of KL divergence, it reveals the distinctive representations of original and adversarial examples in the probability space, providing a robust foundation for detecting adversarial examples. By establishing an appropriate KL divergence threshold, we can reasonably distinguish between original and adversarial examples. Generally, if a sample's KL divergence from the original sample surpasses this threshold, the sample could be classified as an adversarial one.

5.6. Comprehensive Detection Capability

In our study, we evaluated five mainstream adversarial example attack methods and provided associated detection parameters for each type of adversarial example. These parameters involve the detection methods used and their corresponding thresholds, as shown in Table 3. First, for the FGSM, PGD, and BIM attacks, we employed the reconstruction error as the primary detection tool. These three attacks yielded different detection rates under the same threshold of 0.0006. The FGSM attack had a detection rate of 91.45%, which is the highest among the three. In comparison, PGD and BIM had detection rates of 81% and 71%, respectively, indicating certain challenges in detecting these two attacks under the given parameters. For the CW and Deepfool attacks, we chose to use KL divergence as the detection tool. At a threshold of 0.0002, the CW and Deepfool attacks had detection rates of

99.09% and 99.90%, respectively, which are very encouraging. Particularly, the Deepfool attack was almost entirely detected, suggesting that the detection framework exhibits very high stability against this type of attack at high thresholds.

Considering all five attack methods, our detection framework achieved an impressive average detection rate of 88.488%. This result not only reveals the efficiency of the framework but also demonstrates its broad adaptability. First, achieving a comprehensive detection rate close to 90% implies that in most cases, adversarial examples will be effectively identified and blocked, thereby significantly enhancing the model's security. In practical applications, such a high detection rate can substantially reduce the success rate of malicious attacks, ensuring that critical tasks maintain normal operation when faced with adversarial attacks. Second, the framework exhibited a high detection capability against five different attack methods, further highlighting its wide adaptability. In real-world scenarios, attackers might attempt various attack strategies, so a comprehensive, multi-strategy detection framework is of paramount importance. This not only saves time and resources required to design separate detection strategies for different attacks but also provides a unified defensive front, reducing the chances for attackers to find vulnerabilities.

Table 3. Assessment of adversarial example detection capabilities.

Attack Types	Detection Rate	Parameters	Detection Method	Threshold
FGSM	91.45%	$\varepsilon = 0.2$	reconstruction error	0.0006
PGD	81%	$\varepsilon = 0.2$	reconstruction error	0.0006
BIM	71%	$\varepsilon = 0.2$	reconstruction error	0.0006
CW	99.09%	confidence = 0	KL divergence	0.0002
Deepfool	99.90%	overshoot = 0.005	KL divergence	0.0002
overall detection rate			88.488%	

5.7. Sample Restoration Experiment Based on Autoencoder

A good reconstructor should be capable of reconstructing adversarial examples into their corresponding normal samples while retaining the essential features of normal samples. This improvement aims to enhance the recognition accuracy of adversarial examples without compromising the classifier's accuracy in identifying normal samples. As illustrated in Figure 8, the recognition rate of original samples was 86.36%, while the reconstructed recognition rate was 85.55%. It is evident that the reconstructed samples generated by the reconstructor from the original samples preserved their fundamental characteristics effectively. However, due to the nature of gradient-based attacks such as FGSM, PGD, and BIM, they severely disrupted the data's feature distribution during adversarial example generation. Even if the reconstructor could restore them within the normal sample's divergence distribution range, the classifier struggled to perform well. Under the settings of $C = 0$ and $C = 0.5$ for CW attacks, the recognition accuracy of adversarial examples generated by CW attacks was 16.55% and 15.55%, respectively. After the reconstruction, the classifier's recognition accuracy improved by 43.54% and 26.27%, respectively. For adversarial examples generated by Deepfool attacks, the recognition accuracy increased from 3.82% to 36.27% after the reconstruction by the reconstructor. It is evident that the reconstructor has a more pronounced effect on the reconstruction of adversarial examples generated by optimization-based attacks such as CW and Deepfool.

Building upon this, further investigation was conducted into the reconstructor's capability to handle adversarial examples generated by CW and Deepfool attacks under different parameters. Figure 9a illustrates the recognition rates of adversarial examples generated by CW attacks under various attack parameters, denoted as C, and the corresponding recognition rates of the reconstructed samples. By analyzing the data in Figure 9a, it is evident that as the parameter C increased, the recognition rate of the reconstructed samples gradually decreased. It decreased from 60.09% at $C = 0$ to 29.36% at $C = 1$, resulting

in a decrease of 30.73%. Similarly, Figure 9b presents the recognition rates of adversarial examples generated by Deepfool attacks under different attack parameters, known as overshoot, and the corresponding recognition rates of the reconstructed samples. Consistently, the data from Figure 9b indicate that as the overshoot parameter increased, the recognition rate of the reconstructed samples progressively decreased. The value decreased from 36.27% with an overshoot of 0.00001 to 27.73% when the overshoot reached 0.1, marking a reduction of 8.54%. This restoration in reconstruction efficacy is attributed to the higher attack intensity associated with larger attack parameters in CW and Deepfool attacks, leading to more severe disruption of the data distribution during adversarial example generation, thus causing a decline in the reconstructor's restoration effectiveness.

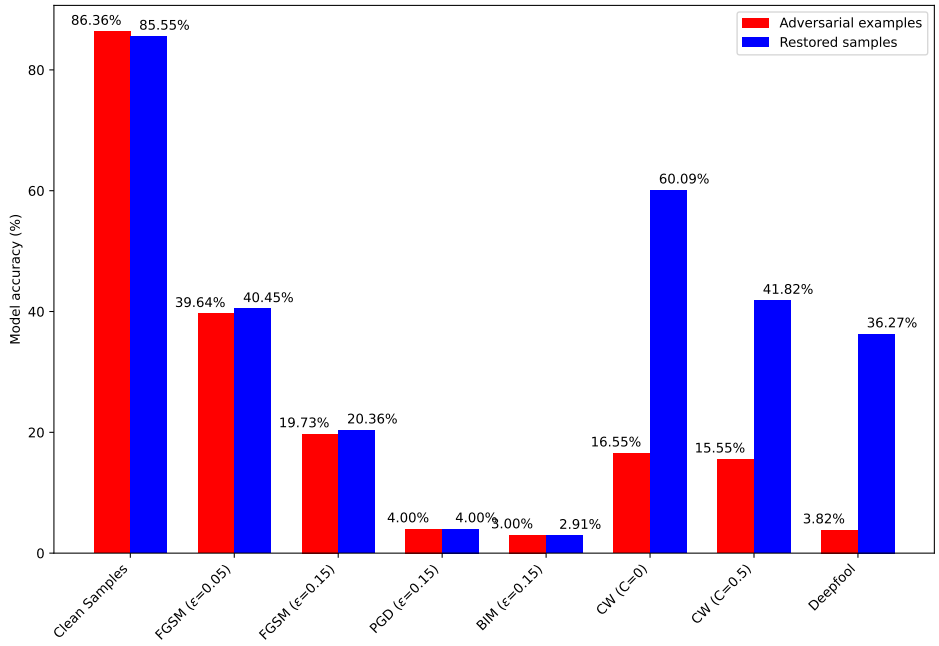
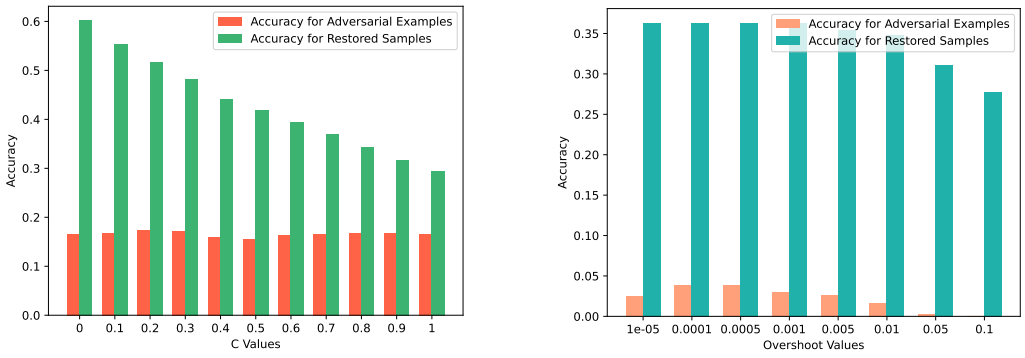


Figure 8. Comparison of accuracy for various adversarial examples and their restoration sample.



(**a**) (**b**)

Figure 9. Comparison of classifier accuracy before and after restoration for different optimization attacks. (**a**) Comparison of classifier accuracy before and after restoration under different C parameters for CW attack; (**b**) comparison of classifier accuracy under different overshoot parameters for Deepfool attacks.

The experimental results indicate a significant improvement in the classifier's recognition rate when dealing with adversarial examples that have been reconstructed by the AE. This outcome could stem from two potential reasons. First, the AE might have successfully learned the true underlying distribution of the original data and effectively removed the

artificially introduced perturbations from the adversarial examples. Consequently, the reconstructed adversarial examples are closer to the original samples in terms of their features, allowing the classifier to make more accurate identifications. Second, the presence of the AE could be exerting a regularization effect on the classifier, reducing its sensitivity to perturbations and enhancing its ability to identify key features. Overall, this observation suggests that the AE can serve as an effective tool to assist the classifier in detecting adversarial examples, thereby improving the model's recognition accuracy.

6. Conclusions

In this research, we introduced an integrated adversarial example detection and restoration framework for signal intelligent recognition using AE. Addressing the pervasive threats of adversarial attacks on deep learning, our framework effectively detects gradient-based and optimization-based attacks. Through the reconstruction error and KL divergence methodologies, we achieved a comprehensive detection rate of 88.48%. Beyond detection, our design excels in restoration, particularly for optimization-based attacks. Using AE, adversarial examples are reverted to their original state, leading to a recognition accuracy boost of over 30% against CW and Deepfool attacks compared to an undefended model. Validation on public datasets confirmed our method's robustness, with FGSM and PGD attacks detected at rates of 91% and 81%, and CW and Deepfool attacks detected at a remarkable 99%. In essence, our approach provides a unified and efficient defense mechanism, enhancing deep learning classifiers' resilience against adversarial threats.

Looking ahead, we plan to delve deeper into understanding the distinct behaviors of adversarial examples and original samples within model decisions. This will offer critical guidance for further enhancing the accuracy of our detection framework. Concurrently, we are committed to developing new and more efficient detection methods to bolster the defensive capabilities of intelligent systems when faced with unknown attacks. Moreover, we will undertake a profound exploration of signal data features, probing the impact of adversarial perturbations on both shallow and deep signal characteristics. The goal is to discern adversarial examples from signals based on a comprehensive understanding of the data, without relying on model representations.

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

Funding: This research was funded by National Defense Key Laboratory Fund grant number 6142413210003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article.

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

References

1. Lin, Y.; Zha, H.; Tu, Y.; Zhang, S.; Yan, W.; Xu, C. GLR-SEI: Green and Low Resource Specific Emitter Identification Based on Complex Networks and Fisher Pruning. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, 1–2. [CrossRef]
2. Zha, H.; Wang, H.; Feng, Z.; Xiang, Z.; Yan, W.; He, Y.; Lin, Y. LT-SEI: Long-Tailed Specific Emitter Identification Based on Decoupled Representation Learning in Low-Resource Scenarios. *IEEE Trans. Intell. Transp. Syst.* **2023**. [CrossRef]
3. Lin, Y.; Tu, Y.; Dou, Z.; Chen, L.; Mao, S. Contour Stella Image and Deep Learning for Signal Recognition in the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 34–46. [CrossRef]
4. Ya, T.; Yun, L.; Haoran, Z.; Zhang, J.; Yu, W.; Guan, G.; Shiwen, M. Large-scale real-world radio signal recognition with deep learning. *Chin. J. Aeronaut.* **2022**, *35*, 35–48.

5. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks: 17th International Conference, EANN 2016, Aberdeen, UK, 2–5 September 2016*; Proceedings 17; Springer: Berlin/Heidelberg, Germany, 2016; pp. 213–226.
6. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
7. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J.D. The security of machine learning. *Mach. Learn.* **2010**, *81*, 121–148. [CrossRef]
8. Dalvi, N.; Domingos, P.; Mausam.; Sanghai, S.; Verma, D. Adversarial classification. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 99–108.
9. Volchikhin, V.; Urnev, I.; Malygin, A.; Ivanov, A. Information-telecommunication system with multibiometric protection of user's personal data. In Proceedings of the Progress in Electromagnetics Research Symposium, Moscow, Russia, 19–23 August 2012; pp. 71–74.
10. Gu, T.; Dolan-Gavitt, B.; Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv* **2017**, arXiv:1708.06733
11. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
12. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
13. Guo, C.; Rana, M.; Cisse, M.; Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv* **2017**, arXiv:1711.00117.
14. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 582–597.
15. Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-Gan: Protecting classifiers against adversarial attacks using generative models. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
16. Ponti, M.; Kittler, J.; Riva, M.; de Campos, T.; Zor, C. A decision cognizant Kullback–Leibler divergence. *Pattern Recognit.* **2017**, *61*, 470–478. [CrossRef]
17. Youssef, A.; Delpha, C.; Diallo, D. An optimal fault detection threshold for early detection using Kullback–Leibler divergence for unknown distribution data. *Signal Process.* **2016**, *120*, 266–279. [CrossRef]
18. Sadeghi, M.; Larsson, E.G. Adversarial attacks on deep-learning based radio signal classification. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 213–216. [CrossRef]
19. Sagduyu, Y.E.; Shi, Y.; Erpek, T. IoT network security from the perspective of adversarial deep learning. In Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA , 10–13 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–9.
20. Bair, S.; DelVecchio, M.; Flowers, B.; Michaels, A.J.; Headley, W.C. On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, Miami, FL, USA, 15–17 May 2019; pp. 25–30.
21. Kokalj-Filipovic, S.; Miller, R.; Morman, J. Targeted adversarial examples against RF deep classifiers. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, Miami, FL, USA, 15–17 May 2019; pp. 6–11.
22. Kokalj-Filipovic, S.; Miller, R.; Vanhoy, G. Adversarial examples in RF deep learning: Detection and physical robustness. In Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 11–14 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
23. Flowers, B.; Buehrer, R.M.; Headley, W.C. Evaluating adversarial evasion attacks in the context of wireless communications. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 1102–1113. [CrossRef]
24. Lin, Y.; Zhao, H.; Ma, X.; Tu, Y.; Wang, M. Adversarial attacks in modulation recognition with convolutional neural networks. *IEEE Trans. Reliab.* **2020**, *70*, 389–401. [CrossRef]
25. Bao, Z.; Lin, Y.; Zhang, S.; Li, Z.; Mao, S. Threat of adversarial attacks on DL-based IoT device identification. *IEEE Internet Things J.* **2021**, *9*, 9012–9024. [CrossRef]
26. Ren, K.; Zheng, T.; Qin, Z.; Liu, X. Adversarial attacks and defenses in deep learning. *Engineering* **2020**, *6*, 346–360. [CrossRef]
27. Tian, Q.; Zhang, S.; Mao, S.; Lin, Y. Adversarial attacks and defenses for digital communication signals identification. *Digit. Commun. Netw.* **2022**. [CrossRef]
28. Kim, B.; Sagduyu, Y.E.; Davaslioglu, K.; Erpek, T.; Ulukus, S. Channel-aware adversarial attacks against deep learning-based wireless signal classifiers. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 3868–3880. [CrossRef]
29. Wójcik, B.; Morawiecki, P.; Śmieja, M.; Krzyżek, T.; Spurek, P.; Tabor, J. Adversarial Examples Detection and Analysis with Layer-wise Autoencoders. In Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 1–3 November 2021; pp. 1322–1326. [CrossRef]
30. Li, T.; Luo, W.; Shen, L.; Zhang, P.; Ju, X.; Yu, T.; Yang, W. Adversarial sample detection framework based on autoencoder. In Proceedings of the 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Bangkok, Thailand, 30 October–1 November 2020; pp. 241–245. [CrossRef]
31. Ye, H.; Liu, X. Feature autoencoder for detecting adversarial examples. *Int. J. Intell. Syst.* **2022**, *37*, 7459–7477. [CrossRef]

32. Xiao, Y.; Pun, C.M. Improving adversarial attacks on deep neural networks via constricted gradient-based perturbations. *Inf. Sci.* **2021**, *571*, 104–132. [CrossRef]
33. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. *arXiv* **2016**, arXiv:1611.01236.
34. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 39–57.
35. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.

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.