Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041346

ARTICLE

An Intelligent Secure Adversarial Examples Detection Scheme in Heterogeneous Complex Environments

Weizheng Wang[1,3], Xiangqi Wang[2,*], Xianmin Pan[1], Xingxing Gong[3], Jian Liang[3], Pradip Kumar Sharma[4], Osama Alfarraj[5] and Wael Said[6]

[1]College of Information Science and Engineering, Hunan Women's University, Changsha, 410138, China

[2]School of Mathematics and Statistics, Hunan First Normal University, Changsha, 410138, China

[3]School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410114, China

[4]Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3FX, UK

[5]Department of Computer Science, Community College, King Saud University, Riyadh, 11437, Saudi Arabia

[6]Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, 44511, Egypt

*Corresponding Author: Xiangqi Wang. Email: xiangqi.wang@foxmail.com

Received: 19 April 2023 Accepted: 19 July 2023 Published: 08 October 2023

ABSTRACT

Image-denoising techniques are widely used to defend against Adversarial Examples (AEs). However, denoising alone cannot completely eliminate adversarial perturbations. The remaining perturbations tend to amplify as they propagate through deeper layers of the network, leading to misclassifications. Moreover, image denoising compromises the classification accuracy of original examples. To address these challenges in AE defense through image denoising, this paper proposes a novel AE detection technique. The proposed technique combines multiple traditional image-denoising algorithms and Convolutional Neural Network (CNN) network structures. The used detector model integrates the classification results of different models as the input to the detector and calculates the final output of the detector based on a machine-learning voting algorithm. By analyzing the discrepancy between predictions made by the model on original examples and denoised examples, AEs are detected effectively. This technique reduces computational overhead without modifying the model structure or parameters, effectively avoiding the error amplification caused by denoising. The proposed approach demonstrates excellent detection performance against mainstream AE attacks. Experimental results show outstanding detection performance in well-known AE attacks, including Fast Gradient Sign Method (FGSM), Basic Iteration Method (BIM), DeepFool, and Carlini & Wagner (C&W), achieving a 94% success rate in FGSM detection, while only reducing the accuracy of clean examples by 4%.

KEYWORDS

Deep neural networks; adversarial example; image denoising; adversarial example detection; machine learning; adversarial attack

1 Introduction

With the improvement in computer performance and data processing capacity, deep neural networks have demonstrated great advantages in intelligent environments such as image and speech recognition [1], autonomous driving [2], natural language processing [3,4], and network security detection [5]. However, recent studies have shown that deep neural networks are vulnerable to AEs [6–8]. AEs are intentionally crafted inputs that can deceive deep learning models, leading to incorrect and potentially harmful outputs. AEs are almost indistinguishable from real samples by the naked eye but can cause models to be misclassified with high confidence. Even if several models have different structures and training data, the same AEs can attack them [9]. As shown in Fig. 1, in a realistic scenario, AEs can perform targeted attacks on machine learning-based target systems, which can be applications with high-security requirements such as autonomous driving, face recognition app, and smart home. The presence of AEs puts the application of machine learning in security-sensitive fields at serious risk, and these threats can trigger machine learning-driven recognition systems to execute incorrect instructions or become paralyzed, even posing a significant risk to people's lives. Motivated by these pressing challenges, this paper aims to address the security issues associated with AEs and enhance the robustness of machine learning systems. The primary motivation behind this study stems from the urgent need to promote the widespread adoption and application of machine learning technologies while ensuring their reliability and security.

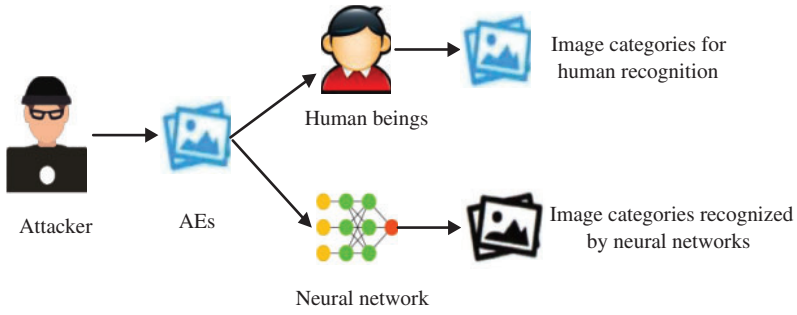

Figure 1: The attacker generates an AE that makes the system different from humans

In recent years, researchers attach great importance to the security issue of AEs, and great progress has been made in the research of adversarial example defense. For example, adversarial training [10] uses the attack algorithm to generate AEs before model training, and then it mixes the AEs with the original samples to train the model, constantly generating new AEs in each step of training to improve the recognition accuracy of the model to the AEs. Defensive distillation [11] reduces the sensitivity of a neural network model to input perturbations by generating a smooth classifier, making the classifier more adaptive to AEs. These defense techniques by modifying the model structure or modifying the training process can significantly enhance the robustness of neural network models. However, the training overhead is high (adversarial training requires a large number of AEs for training), the complexity is high (defensive distillation requires adding distillation temperature and modifying the objective function), and the defense against black-box attacks, which are widely used in real-world scenarios, is ineffective. In addition to enhancing the ability of the model to resist AEs, AE detection [12–15] is also the mainstream defense technology at present. In this technology, before the image is classified by the image recognition model, the image is screened by the detection system to detect whether it is an AE. AE detection does not require AEs for training, nor does it require modification of model structure and parameters, reducing training overhead and defense complexity, and making it

easier to deploy in realistic machine learning systems. However, the performance of the AE detection is closely related to the detector. In addition, this method only detects whether there are AEs, and can not improve the robustness of the model. Table 1 summarizes the advantages and disadvantages of the current mainstream AE defense technology.

Table 1: The advantages and disadvantages of defense techniques against AEs

Type	Advantages	Disadvantages
Adversarial training	Simple operation, remarkable defense effect	High training cost and poor generalization ability
Defensive distillation	High generalization ability and low training cost	High computational complexity, difficult to resist black box attacks
AEs detection	Low computational complexity, independent of model structure and adversarial sample algorithms	Highly correlated with the detector and did not improve the robustness of the model

Previous research has made notable strides in AE defense techniques, including adversarial training and defensive distillation, which modify model structures or training processes to enhance robustness. However, these approaches often suffer from high training overhead, increased complexity, and limited effectiveness against real-world black-box attacks. On the other hand, AE detection has gained prominence as a defense mechanism that screens inputs to identify AEs, without modifying the model or incurring significant training costs. However, the performance of AE detection heavily relies on the choice of the detector and does not improve the overall robustness of the model.

This paper presents a new AE detection technology based on image denoising. Without modifying the model structure and affecting the accuracy of clean samples, the input samples are detected based on the difference between the prediction of adversarial examples and clean samples before and after image denoising. The training cost of this method is very small, and it can effectively detect the AEs generated by the current mainstream attacks. As well, the proposed detection technology can be applied to image data filtering. In applications with high-security requirements, the image dataset is first detected, and if an image is detected as an AE, it will be filtered or restored for secondary processing. Therefore, the research of this paper has important theoretical and practical significance.

The main contributions of this paper are as follows:

(1) The paper presents a novel adversarial example detection scheme based on the inconsistency of model prediction between original samples and AEs before and after image denoising. The detection technique aims to identify AEs without modifying the model structure or affecting the classification accuracy of clean samples.
(2) The detection scheme involves training a detector model by integrating the classification results of different models using a machine-learning voting algorithm. The detector is fine-tuned by comparing the classification results of original images and denoised images obtained from multiple image-denoising algorithms.
(3) The AE detection framework consists of classifying an input image using the detector, denoising the image, and reclassifying it using the detector. The deviation between the classification results before and after denoising is calculated using the L2 norm and compared with a detection threshold. If the deviation exceeds the threshold, the image is classified as an AE.

(4) The proposed method is evaluated using the MNIST dataset and various AE attack algorithms. The results demonstrate the effectiveness of the detection scheme, with success rates ranging from 88% to 94% for detecting AEs generated by different attacks. Compared to other defense techniques, the proposed method shows better performance while maintaining high prediction accuracy for clean samples.

The rest of this paper is organized as follows: Section 2 introduces the preliminary knowledge of this paper, including the basic concepts of image denoising and AEs. Section 3 describes in detail the detection framework and the specific implementation process proposed in this paper. Section 4 evaluates our proposed detection technique under different AE attacks by experiments. Finally, we summarize the current research results, give an outlook on the subsequent research of this paper, and propose possible solutions and research directions.

2 Related Work

In 2014, Szegedy et al. [6] first introduced the concept of adversarial examples and proposed the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method to construct AEs. The nonlinear and nonconvex objective optimization problem is approximated by minimizing the loss function to find the minimum additional term of the loss function that makes the model misclassify. This method is stable and effective, but it has high computational complexity. Goodfellow et al. [10] explained subsequently the basic principle of AEs and proposed one of the simplest and most effective FGSM. Adding perturbations in the gradient direction and linearizing the loss function causes the model to misclassify the generated images. This method is efficient in constructing AEs and thus has been widely used in the field of AE research, often as a benchmark attack for new defense frameworks. However, the perturbation size of FGSM-generated AEs is not well controlled and prone to label leakage. Kurakin et al. [16] proposed a BIM for optimization under FGSM. By applying the adversarial perturbation to small step sizes and clipping the results after each iteration step to ensure that the perturbation size is within the neighborhood of the original class, a high-quality AE can eventually be generated. Recently, Carlini et al. [17] proposed a C&W attack based on iterative optimization of L-BFGS. Under the L_0, L_2, and L_∞ distance metrics, the attack constructs high-quality AEs by using the objective function of optimization constraints, and can successfully defeat the mainstream defensive distillation technology. By now, adversarial attacks have been studied widely in many fields, such as object detection and tracking [18,19], reinforcement learning [20,21], face recognition [22,23], and healthcare data [24].

In recent years, the proliferation of research on AE attack algorithms has made the development and application of deep learning in security-sensitive fields suffer from a great threat [25], and these threats may trigger a complete breakdown of deep learning-driven recognition systems, and even pose a risk to life for applications with high security-level requirements such as autonomous driving. To counter the security risks posed by AEs, researchers have proposed a series of defense mechanisms for AEs.

The research on AEs defense is mainly divided into two aspects: 1) Active defense: Enhancing the robustness of neural network model through technical reinforcement; 2) Passive defense: Independent of the attack algorithm and the network model structure, it is only necessary to determine whether the image is an AE or not. Fig. 2 shows the different forms of AE defenses. For the study of active defense, Goodfellow et al. [10] were the first to propose a direct and effective adversarial training defense technique. The basic idea is to include AEs in the training process, train the model with clean samples as part of the training set, and continuously generate new AEs at each step of training, thus

enhancing the model's ability to resist the AEs. However, to ensure the recognition accuracy of AEs, a large number of AEs are needed to train the model, and the training process is tedious and costly. The defensive distillation proposed by Papernot et al. [11] reduces the sensitivity of the neural network model to input perturbations by generating smooth classifiers, making the classifier more adaptable to adversarial samples. This defensive technique is independent of the generation of AEs and has a high generalization capability. However, an attacker can easily bypass the distillation model by training an alternative model similar to the distillation model and then using the gradient of the alternative model to generate the AEs, so the defensive distillation can be easily broken by black-box attacks. Moreover, the technique requires changing the model structure and retraining the classifier, further increasing the overhead and complexity of the defense.

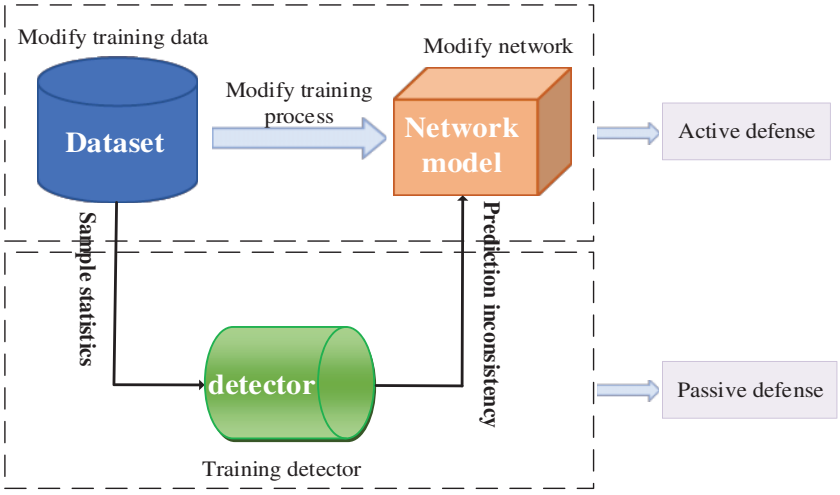

Figure 2: Different ways of defending against AEs

Compared with active defense, the research of passive defense is relatively much simpler. Passive defense distinguishes clean samples and AEs by detection. Gondara [12] used density ratio estimation as a measurement method of the model to detect AEs. Based on the fact that clean samples and AEs have different potential probability densities, AEs with high confidence are detected by estimation of the density ratio. This method can handle grayscale and color images well, but has high computational complexity and can only detect AEs far from the decision boundary. Meng et al. [13] proposed a defense framework MagNet that includes multiple detector networks and a reformer network. The detector network learns to distinguish clean samples from AEs by approximating multiple forms of clean samples, and the reformer network shifts AEs to multiple forms of clean samples. This defense technique has good detection in both black-box and gray-box attacks, but the training overhead is high, and modifying the original model reduces the classification accuracy of clean samples. Jia et al. [14] proposed a defense framework called ComDefend based on image compression reconstruction. ComDefend consists of two CNN modules including ComCNN and ResCNN, where the role of ComCNN is to store the main structural information of the original image and the role of ResCNN is to restore the original image with high resolution. ComDefend processes the image in image blocks instead of directly processing the whole image, which reduces the training time and computational overhead, but the clean samples after image compression will lose some of the prediction accuracies. Xu et al. [15] proposed an AE defense strategy based on feature compression. The method adds two external models to the Deep Neural Networks (DNN) classifier for reducing the color bit depth of

each pixel and smoothing pixels using a spatial filter, respectively. The samples are distinguished based on the model's inconsistency in the prediction of clean samples and AEs before and after feature compression. The method shows excellent detection performance under different attacks, but it degrades the classification accuracy of the model for clean samples on a relatively complex dataset like ImageNet.

Before the work of this paper, many scholars carried out in-depth research on AE defense based on image denoising [26–28]. The basic framework of its defense is shown in Fig. 3. For a given input image, it is first filtered by the detector. Then, the detected AEs are preprocessed by an image-denoising algorithm to reduce the small disturbance of AEs. Finally, the processed AEs are input into the model to be correctly classified, while the clean samples are directly input into the model for prediction and classification. In this way, the image recognition model can be protected from the attack of AEs.

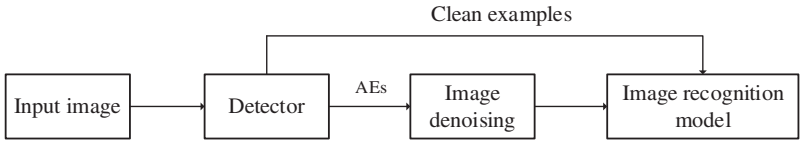

Figure 3: The framework of defending against AEs based on traditional image denoising

With the continuous development of AE defense technology, image denoising has been widely used as an effective defense method. Xie et al. [29] proposed a method to remove minor adversarial perturbations by performing a random transformation on the input image. Specifically, two random layers, a random resize layer and a random fill layer, are introduced before the original image enters the classification model, and the original image is transformed by the above two random layers and then passed to the classification model. This method can resist white-box attacks well without retraining or changing the model. However, this framework requires a large number of AEs trained in a complex way to have a good defense. Gu et al. [27] proposed a defense framework for deep compression networks using a smoothness penalty term regularization network similar to the compressive self-encoder to enhance the robustness of the network. The defense of the deep compression network is better under adversarial training, but the computational complexity is too high. Hu [28] proposed a D-D model based on deep residual learning denoising, which improves the accuracy of identifying AEs by superimposing two different defense models, reverses the law of denoising by using residual learning, and improves the training efficiency of the model by using Batch Normalization (BN) and Rectified Linear Units (ReLU) layers. Dong et al. [26] proposed a High-level representation Guided Denoiser (HGD) network to train a neural network-based denoiser to remove adversarial perturbations. In the training process, the loss function is added directly to the higher-level feature layers of the network, trained on a small subset of images, and then narrowly shifted to other classes. Compared to other denoising networks, HGD has good migration properties. In the conference and workshop on Neural Information Processing Systems (NIPS) adversarial defense competition, the HGD network won with a significant lead in detection performance. However, HGD training requires a large number of AEs, and the training overhead is high. Besides, the detection performance under white-box attacks is poor.

3 System Model and Definitions

3.1 Image Denoising

Digital images will be disturbed by different noises in the process of acquisition and transmission to varying degrees, which brings some trouble to the high-level image processing, so image denoising

has become an important part of image pre-processing. The goal of image denoising is to obtain a clean image from a noise-containing image by subtracting noise and to restore the original image information to the maximum extent while still retaining enough detailed information. Specifically, for an image $v(x)$ with noisy input, the additive noise can be expressed as:

$$v(x) = u(\mathrm{x}) + \eta(x), \ \mathrm{X} \in \Omega \tag{1}$$

where $u(x)$ represents the image without noise, $\eta(x)$ is additive noise, representing the impact of noise, and Ω is the set of pixels, i.e., the whole image. According to whether denoising is combined with a machine learning model, the image-denoising algorithm is mainly divided into traditional denoising algorithm and machine learning denoising algorithm.

The traditional denoising algorithm finds out the law from the noisy image, and then it carries on the corresponding denoising processing. The main algorithms are mean denoising [30], Non-Local Means (NLM) denoising [31], and Block-Matching and 3D (BM3D) filtering [32], which are usually able to handle images with specific noise types. For example, mean denoising can effectively deal with Gaussian noise, while wavelet denoising is mainly applicable to images with white noise. In real scenarios, due to the imperfection of digital devices, the original images are inevitably contaminated by various kinds of noise in the process of transmission and storage. Therefore, we can carry out comprehensive denoising according to the characteristics of different denoising algorithms, such as combining median denoising and wavelet denoising for filtering, which can effectively reduce the noise in images while ensuring the integrity of image edges, textures, and other detailed information. In practical applications, to improve the generality of the denoising algorithm, different denoising algorithms or combinations of many different denoising algorithms should be applied according to the characteristics of the original image and the type of noise. If we cannot find the law from the noisy image itself, we can use machine learning to denoise. We summarize the inherent attributes of the image by constantly learning the characteristics of noise, and we denoise through the statistical characteristics of the image. Machine learning denoising mainly includes convolution neural network denoising [33], self-encoder denoising [34], and generative adversarial network denoising [35]. Compared with the traditional image-denoising algorithm, the machine learning denoising algorithm can denoise well and retain the detailed information of the image edge, but the denoising speed still needs to be improved. Based on the target of image denoising, image denoising should meet four aspects at the same time:

(1) The noise contained in the image should be removed as much as possible;
(2) The integrity of important detail information (such as edge and texture) contained in the image should be ensured;
(3) Additional noise types should not be introduced in the process of noise removal;
(4) In the real scene, the denoising efficiency should be high enough.

Only when these four requirements are met simultaneously can the best effect of noise removal be achieved. However, the current traditional image denoising is difficult to achieve a balance between removing noise and retaining image edge detail information. Machine learning denoising satisfies the requirements of (1)(2)(3) well, but the existing machine learning denoising methods are still in the experimental stage, and further improvement is needed for high training speed and recovery performance.

3.2 Adversarial Example

In 2014, Szegedy et al. [6] first proposed the concept of AEs. The AE introduces a slight disturbance to the input of the neural network model so that the disturbed input is incorrectly classified

by the neural network model with high confidence, but the human eye cannot distinguish the changes after image disturbance. Formally, supposing that there is a machine learning model M and the original sample C correctly classified by the model, $M(C) = y_{true}$, where y_{true} is the real label of C. Perturbing the original sample C to generate an adversarial example C', $M(C') \neq y_{true}$, which is incorrectly classified by the model. A typical example is shown in Fig. 4. In the image on the left, the neural network model thinks that the image is a "Panda" (57.7%), but with small perturbations, it is classified as "gibbon" by the model with 99.3% confidence after being transformed to the right when the human eye cannot see the difference at all.

Figure 4: Generating an AE with the FGSM attack algorithm

For a more intuitive representation, we use the neural network model in Fig. 4 as an example to show the change in output by perturbing the inputs. For any of the inputs, small changes will not affect the overall prediction of the classifier, but small changes to all dimensions of the inputs will result in large changes in the output of the classifier. As shown in Fig. 5, where $W^{(1)}$ and $W^{(2)}$ are weight matrices, and the original input value and weight value are randomly initialized. After disturbing the original input by size sign (0.5), the adversarial inputs X_1', X_2', and X_3', are all equal to 1.5. Then, after the transformation operation of the weight matrix $W^{(1)}$ in the first layer and the activation function ReLU, the adversarial outputs $a_1'^{(2)}$, $a_2'^{(2)}$, and $a_3'^{(2)}$ in the first layer are all equal to 1.5. Finally, after the transformation operation of the weight matrix $W^{(2)}$ in the second layer and the activation function sigmoid, we find that the probability of the output class changes from 0.2689 to 0.8176, which is enough to make the model misclassify with high confidence. With the increase of the depth of the neural network model, the probability change of the output class will be more obvious.

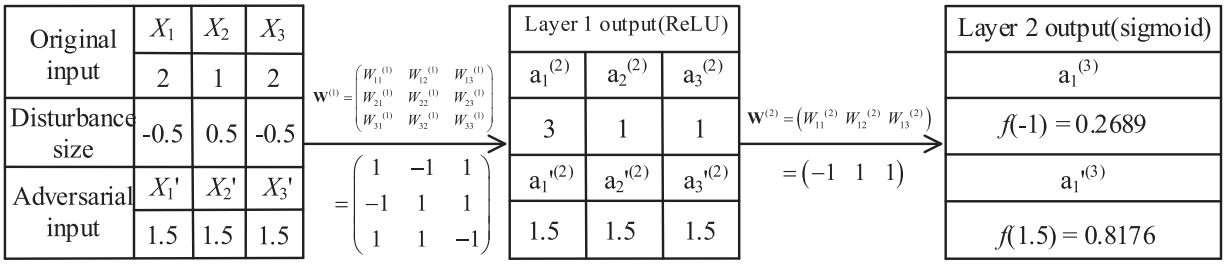

Figure 5: The change of output after adding perturbations to the inputs of the neural network [36]

AE attacks can be directed, in which the opponent's goal is to classify the output into a specific class. It can also be nondirectional, in which the opponent only needs to classify the output into any class other than the correct class. More formally, taking the directed attack as an example, given an input $x \in X$ and a classifier $(f(\cdot))$, the attacker's goal is to find an antagonistic input $x' \in X$ by adding

small perturbations to the input X, and x' is classified as target class t. Modeled as a mathematical problem, the target of the directional attack is:

$$f(x) \neq f(x') = t \wedge \| x, x' \|_p \leq \varepsilon \tag{2}$$

where $\| \cdot \|_p$ represents the distance norm of the original input x and the antagonistic input x', and p can be 0, 1, 2, and infinity. ε is used to limit the size of the disturbance and avoid the excessive disturbance added to the input being detected by human eyes.

4 Our Proposed AE Detection Scheme

The detection technique proposed in this paper is based on the inconsistency of model prediction between the original sample and the AE before and after the denoising process to detect the input samples. Namely, the prediction accuracy obtained by applying the AE to the model changes significantly before and after image denoising, and the prediction accuracy of the clean sample remains unchanged. Specifically, the image recognition model is first used to classify the original image, and the results of the classification (prediction accuracy) are recorded. Then different image-denoising algorithms are used to denoise the images, and the processed images are classified under the same model, and the classification results are recorded. Finally, the differences in classification results of images before and after denoising are compared. If the classification results before and after image denoising are the same, the sample is clean, otherwise, the sample is an AE. For the detected AEs, we can also perform a secondary intervention to determine whether to discard and restore the samples. For example, if a suspected AE is detected in autonomous driving, we can first pull over and then manually intervene to judge the next operation; if a suspected AE is detected in face recognition, we can stop the machine recognition to manually verify it. The detection technique proposed in this paper can effectively detect the AEs generated by the current mainstream attack algorithms without modifying the model structure and without affecting the classification accuracy of the original samples, and will avoid the error amplification effect brought by image denoising. The research process of this detection technique is divided into two main stages: training the detector model and detecting AE.

4.1 Training Detector Model

The detector model integrates the classification results of different models as the input to the detector and calculates the final output of the detector based on a machine learning voting algorithm. As shown in Fig. 6, the specific training process is as follows:

(1) The adversarial dataset is constructed using the mainstream AE attack algorithms (FGSM, BIM, DeepFool [37], C&W).
(2) Input the adversarial data set to the image recognition model 1 for classification, and record the result p1 of image classification by model 1.
(3) For each input image of the same data set, four different image-denoising algorithms (two traditional denoising algorithms and two machine learning denoising algorithms) are used for denoising. Then the other four models are given for classification, respectively, and the classification results corresponding to these models are recorded as p_2, p_3, p_4, and p_5.
(4) The results of classifying the same image by different models in step (1) and step (3) are compared correspondingly, and four different deviation values d_1, d_2, d_3, and d_4 are obtained. Then, based on the machine learning voting algorithm, the classification results of the detector model and the corresponding confidence score are outputted.

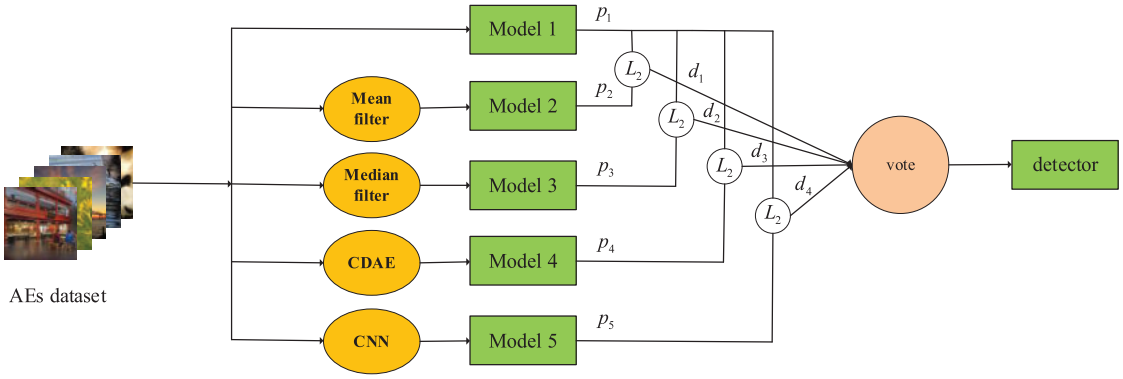

Figure 6: Training a detector model

The detector is obtained by integrating five image recognition models for fine-tuning. During the training process, we first keep the structure and parameters of the detector consistent with those of the image recognition model before the fully connected layer. Then the weights and biases of the fully connected layer of the detector are randomly initialized. At last, the final output of the detector is obtained by continuously iteratively updating the fully connected layer weights.

Algorithm 1: Integrate_detector (X,Detector)//Training detector model

Input: image X, Image recognition model Detector [7].
Output: the category of the vote: vote, the average probability of the class: prob.

```
SET vote, prob
for model,denoised_model In Detector do
    predict = run_model(X,model)
    predict_denoise = run_model(X,denoised_model)
    index= argmax(predict,predict_denoise)
    vote[index] += 1
    prob + = (predict+ predict_denoise)
end for
prob= prob/5
Return vote, prob
```

Algorithm 1 implements the specific process of training the detector model. The detector represents the detector model, which is a set of 5 objects, each object represents an image recognition model. The detector is trained by the input image $\boldsymbol{X}$ and the prediction accuracy of the image recognition models (prediction accuracy of one original model and prediction accuracy of four denoised models), and the result of the detector voting, "vote", and the corresponding class average confidence score, "prob", are obtained based on a machine learning voting algorithm.

4.2 AE Detection

4.2.1 AE Detection Framework

The main goal of the AE detection part is to verify the feasibility of detecting AEs based on image denoising and test the detection performance of the detector. As shown in Fig. 7, for an input image to be tested, it is first classified by the detector to obtain the predicted result p. Then the input

image is denoised by using the denoising algorithm, and it is classified by the detector to get the predicted result p_1. Based on the distance norm L_2, the deviation d before and after image denoising is calculated and then compared with the detection threshold K obtained in the training phase. If the deviation is less than the detection threshold, the image is a clean sample. Otherwise, the image is an AE, and it will be discarded or restored for secondary processing.

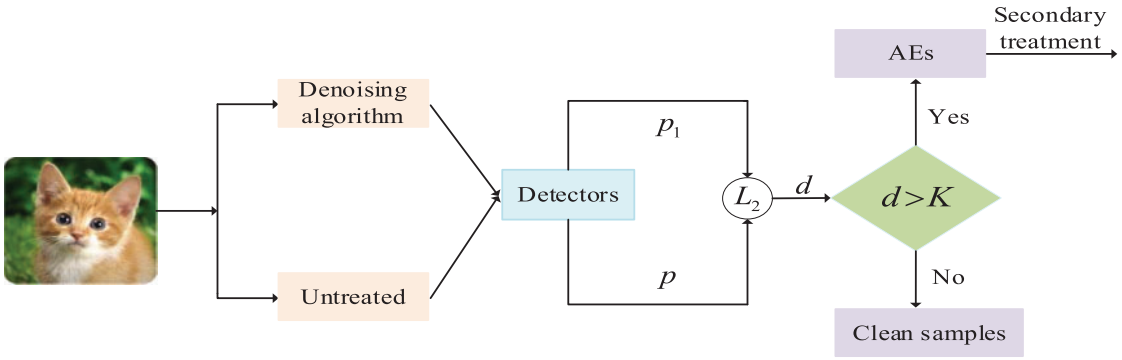

Figure 7: AE detection framework

Algorithm 2: Calculate the optimal detection threshold

Input: Image X, Denoised images $X^{denoise}$,Voting algorithm$VOTE$
Output: Optimal detection threshold K

SET $K = 1$
for $i \in [0, len(C)]$ **do**
  $K_i = \min(C_i)$
  $P_i = detecter(K_i)$ //Evaluate the classification performance of the detector on the dataset detection threshold varies under different AE
  if $P = P_i$ and $K = K_i$ **then**
    $P = P_i$
    $K = K_i$
  end if
end for
Return K

4.2.2 Detection Threshold Selection

The detection threshold is determined based on multiple update iterations of different models during the training phase. The selection of an appropriate distance threshold is specifically described by using the following steps (Algorithm 2, multiple iterations):

(1) Record the prediction accuracy of each model;
(2) Compare the absolute value of the difference between the prediction accuracy after denoising and the prediction accuracy without any processing;
(3) According to the value obtained in step (2), the distance value is calculated by the norm L2, and sorted;
(4) According to the majority voting algorithm, the threshold that satisfies the majority of the distance values is selected as the final detection threshold of the detection model.

As shown in Algorithm 2, based on the known input image $\boldsymbol{X}$ and the denoised image $\boldsymbol{X}^{\text{denoise}}$, we use the voting algorithm in machine learning to calculate the prediction difference threshold C between different models. C is defined specifically as:

$$C = \boldsymbol{VOTE}\left(\| f(X) - f\left(\boldsymbol{X}^{denoise}\right) \|_2\right) \tag{3}$$

where $VOTE$ is the voting algorithm. We use the minimum value of the predicted difference threshold C between every two models as the candidate threshold. Each candidate threshold is then evaluated in turn according to the accuracy of the original samples, and in this way, the optimal threshold is selected. It should be noted that the optimal detection threshold varies under different AE attacks. In general, the higher the strength of the attack, the smaller the optimal detection threshold.

5 Experiment Results and Analysis

5.1 Experimental Setup

The dataset adopted in this experiment is the most classical MNIST dataset in the field of image recognition, where the MNIST dataset is provided by NIST, USA. This dataset contains 60000 training images and 10000 test images. The category labels of each image correspond to 0–9, and each sample is a grayscale handwritten digital image of fixed size 28×28 pixels with values between 0 and 1. We used AlexNet as the original model for training, and we used Denoising Convolutional Neural Network (DnCNN) and Denoising Autoencoder (DAE) as the models for convolutional neural network denoising and self-encoder denoising. As shown in Table 2, AlexNet is an 8-layer convolutional neural network model consisting of 5 convolutional layers, a maximum pooling layer, a fully connected layer, and a softmax layer. DnCNN is an 18-layer feedforward noise-reducing convolutional neural network. The middle 16 layers are normalized using batch normalization, which can speed up the model training and image-denoising efficiency. As shown in Fig. 8, Autoencoder consists of an encoder and a decoder, where the encoder has 2 convolutional layers and 2 maximum pooling layers, and the decoder also has 2 convolutional layers and 2 maximum pooling layers. Compared with convolutional neural network denoising, self-encoder denoising generates the corresponding input and output by coding and decoding, and then generates the corresponding denoised image by adding specific noise to the input image.

Table 2: Network architecture of the AlexNet and DnCNN

Type of layers	AlexNet	Type of layers	DnCNN
Convolutional layer + ReLU	$5 \times 5 \times 20$	Convolutional layer + ReLU	$3 \times 3 \times 64$
Convolutional layer + ReLU	$5 \times 5 \times 40$	Convolutional layer + BN + ReLU	$3 \times 3 \times 64$
Convolutional layer + ReLU	$3 \times 3 \times 80$	Convolutional layer + BN + ReLU	$3 \times 3 \times 64$
Convolutional layer + ReLU	$3 \times 3 \times 80$	.	
Convolutional layer + ReLU	$3 \times 3 \times 40$	.	
Maximum pooling layer	2×2	.	
Fully connected layer	10	Convolutional layer + BN + ReLU	$3 \times 3 \times 64$
Softmax	10	Convolutional layer	$3 \times 3 \times 1$

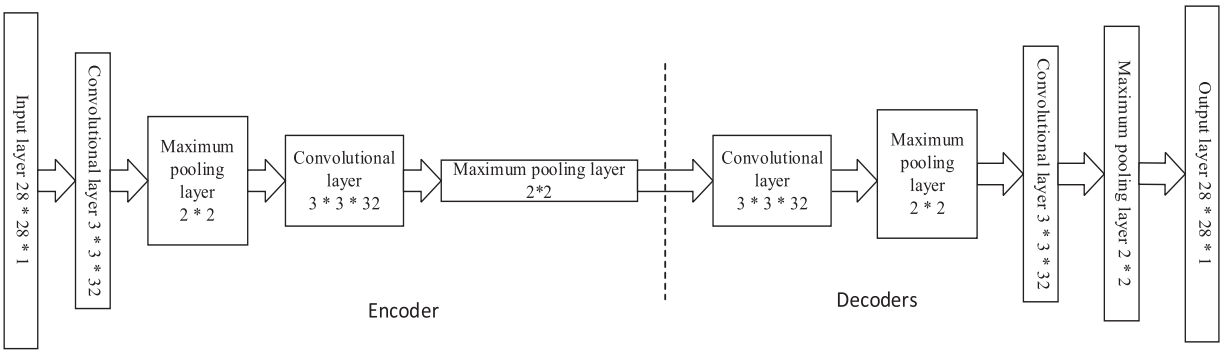

Figure 8: Network architecture of autoencoder

5.2 Image Denoising Effect

We used 2 traditional denoising methods and 2 machine-learning denoising methods for our experiments. Among them, the traditional denoising methods are median denoising and mean denoising, which are implemented using Python's Scipy library. As shown in Fig. 9, we test the effect of the 2 traditional denoising methods by adding pretzel noise and Gaussian noise, respectively. These two methods are fast and can remove the pepper noise and Gaussian noise relatively well, but they also blur the edge information of the image. We use the machine learning denoising of the denoising self-encoder and Feed-forward DnCNN, respectively. The encoder and decoder of DAE are implemented using Keras, and DnCNN is implemented using Tensorflow and OpenCV. the results of DAE and DnCNN denoising are shown in Figs. 10 and 11. Compared with the traditional denoising methods, the quality of the images obtained by machine learning denoising is significantly higher. Because the noise with certain regularity can be generated by feature extraction and autonomous learning of machine learning, the machine learning denoising method has a good effect on different types of noise and noise with different coefficients. However, during the experiments, it was found that machine learning denoising takes several minutes in the best case to denoise an image while the traditional denoising algorithm takes only a few seconds. As the architecture becomes increasingly complex, denoising algorithms may be time-consuming and resource intensive. Therefore, the efficiency of machine learning denoising needs to be further improved.

5.3 Effectiveness of Antagonistic Sample Detection

The AE detection is based on the inconsistency of model prediction between the original samples and AEs before and after the denoising process. After processing by denoising algorithms, the prediction accuracy of the classifier for the original samples is unchanged, while the prediction accuracy for the AEs varies greatly. Especially, the difference is more obvious for the AEs after combining various types of denoising algorithms. In this paper, AEs are generated by using FGSM, BIM, DeepFool, and C&W attacks in "cleverhans" [38]. "cleverhans" is an open-source Python library for adversarial attacks, defense, and benchmarking of machine learning models, which contains instructions and code implementations of the current mainstream AE attack algorithms and defense techniques. As shown in Table 3, we evaluated the detection effectiveness of our proposed defense method using AEs generated by different attack algorithms under optimal thresholds in the MNIST dataset.

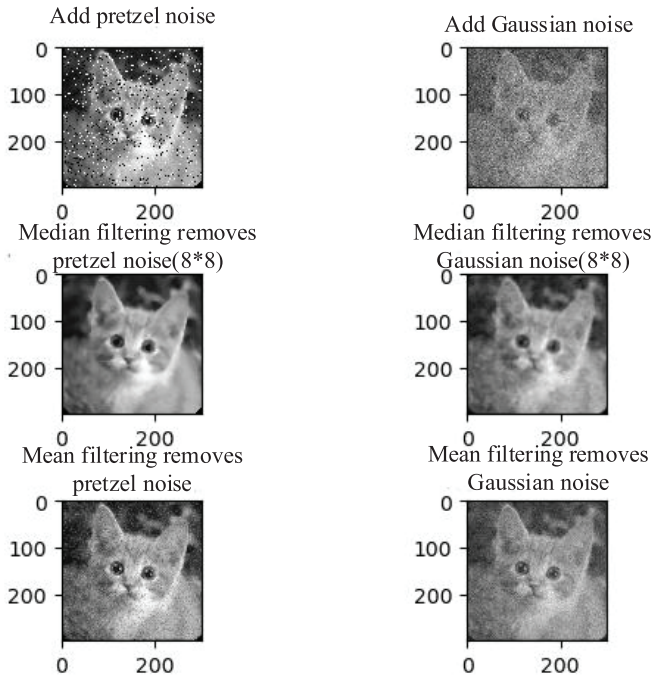

Figure 9: Effect of the traditional image-denoising algorithms

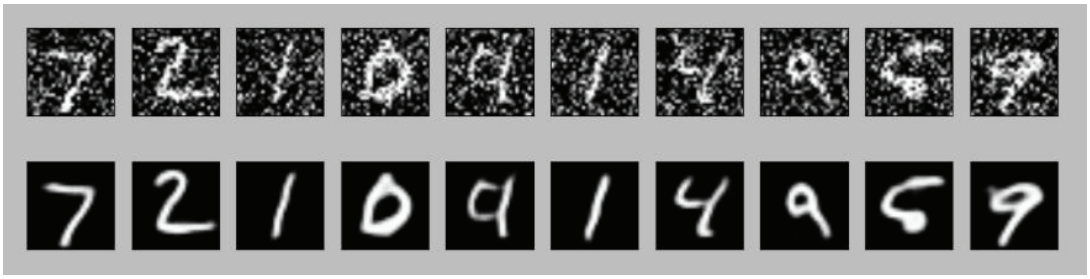

Figure 10: Effect of the autoencoder denoising algorithm

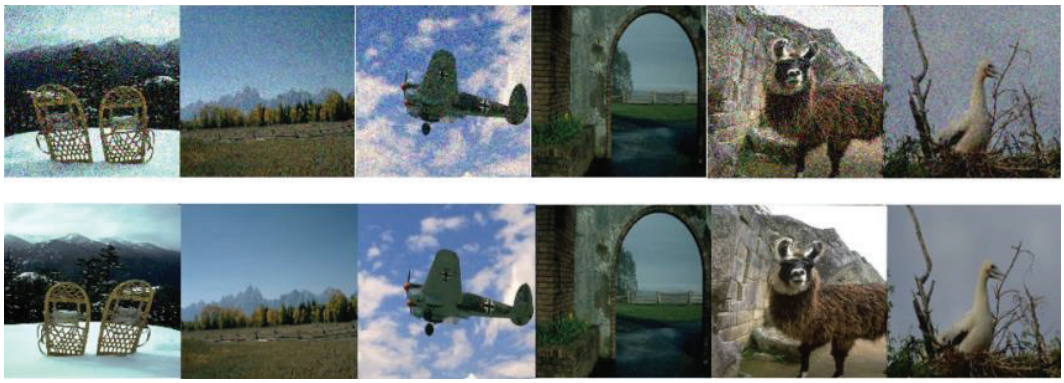

Figure 11: Effect of the DnCNN denoising algorithm

Table 3: The detection success rate of the detection model proposed in the paper

Attack	Threshold	Parameter	No defense	Defensive
FGSM	0.0100	L_∞	46%	94%
BIM	0.0100	L_∞	11%	89%
DeepFool	0.0044	L_∞	4.5%	88%
C&W	0.0080	L_2	0%	90%

Table 3 shows the success rate of the proposed method to detect the AEs in this paper. For the optimal detection threshold, we choose the size of 0.01 for the FGSM and BIM attacks. Due to the high strength of DeepFool and C&W attacks, we choose sizes of 0.0044 and 0.008, respectively. In the undefended case, the C&W attack shows the strongest aggressivity. The lower success rate of the FGSM attack is mainly attributed to the excessive perturbation of the attack with the label leakage effect. After using our proposed defense method, the success rate of model detection is over 88% in all cases. Especially, the success rate of detection of the AEs generated by the FGSM attack reaches 94%. This shows that our proposed defense method is effective and generalizes well under different attack algorithms.

To measure the performance of the proposed method, we compare the detection technique proposed in this paper with other related defense techniques. Table 4 shows the comparison results with other defense techniques on the MNIST image dataset. Compared with HGD and ComDefend, the method has less impact on the original samples and loses only 4% of the prediction accuracy of the original samples. The best detection performance is indicated under FGSM, BIM, and C&W attacks. One possible reason for the slightly degraded detection performance under the DeepFool attack is that the perturbation required to generate the AE for this attack is extremely small, and image denoising is sometimes difficult to effectively remove this small imperceptible perturbation, which ultimately leaves the label of the AE unchanged. In addition, the proposed denoising method mainly removes redundant information from the images and thus has an almost negligible impact on the original samples.

Table 4: Comparison of the defense mentioned with other defense techniques

Defensive methods	Original sample	FSGM	BIM	DeepFool	C&W
—	76%	3%	0%	1%	0%
HGD	54%	50%	36%	52%	51%
ComDefend	67%	56%	12%	**53%**	54%
Proposed method	**72%**	**59%**	**42%**	44%	**56%**

6 Conclusion

In this paper, our study presents an AE detection framework that combines multiple image-denoising algorithms and CNN network structures. Our key finding is that by analyzing the inconsistency in model predictions between original samples and AEs before and after the denoising process, we can effectively detect AEs without modifying the model structure or compromising the accuracy of original samples. The framework integrates traditional denoising algorithms such as mean

and median denoising, as well as machine learning denoising algorithms like convolutional neural networks and self-encoders. By utilizing these techniques, the method filters AEs and clean samples, significantly reducing the error amplification effect associated with image denoising and minimizing defense costs and overhead. Our experimental results demonstrate the excellent detection performance of the proposed method against mainstream AE attacks, including FGSM, BIM, DeepFool, and C&W. Notably, the method achieves a 94% detection success rate for FGSM, with only a 4% reduction in the accuracy of clean examples. These findings underscore the effectiveness and efficiency of our AE detection technique, showcasing its potential to enhance the security of intelligent environments susceptible to targeted attacks. In summary, this research contributes to the field of adversarial defense by providing a practical and robust AE detection framework. By leveraging multiple image-denoising algorithms and analyzing prediction inconsistencies, our method achieves high detection performance while preserving the accuracy of original samples. This study opens up new avenues for developing efficient and reliable defense mechanisms against adversarial examples in various machine-learning applications.

Acknowledgement: The authors would like to thank the Researchers Supporting Project of King Saud University for supporting this work.

Funding Statement: This work was supported in part by the Natural Science Foundation of Hunan Province under Grant Nos. 2023JJ30316 and 2022JJ2029, in part by a project supported by Scientific Research Fund of Hunan Provincial Education Department under Grant No. 22A0686, and in part by the National Natural Science Foundation of China under Grant No. 62172058. This work was also funded by the Researchers Supporting Project (No. RSP2023R102) King Saud University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: W. Wang, X. Pan, J. Liang; data collection: X. Gong, J. Liang; analysis and interpretation of results: X. Wang, P. K. Sharma; draft manuscript preparation: O. Alfarraj, W. Said. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data underlying this article will be shared on reasonable request to the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Yi and M. Mak, "Improving speech emotion recognition with adversarial data augmentation network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 172–184, 2022.

[2] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian *et al.,* "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Emerging Topics in Computing*, vol. 23, no. 2, pp. 722–739, 2022.

[3] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta *et al.,* "Globally normalized transition-based neural networks," in *Proc. of 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 2442–2452, 2016.

[4] J. Lan, R. Zhang, Z. Yan, J. Wang, Y. Chen *et al.,* "Adversarial attacks and defenses in speaker recognition systems: A survey," *Journal of Systems Architecture*, vol. 127, pp. 102526, 2022.

[5] T. Saha, N. Aaraj and N. K. Jha, "Machine learning assisted security analysis of 5G-network-connected systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 4, pp. 2006–2024, 2022.
[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan *et al.,* "Intriguing properties of neural networks," in *2nd Int. Conf. on Learning Representations*, Banff, Canada, pp. 1–10, 2014.
[7] J. Zhang, S. Peng, Y. Gao, Z. Zhang and Q. Hong, "APMSA: Adversarial perturbation against model stealing attacks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1667–1679, 2023.
[8] J. Zhang and J. Hou, "Unpaired image-to-image translation network for semantic-based face adversarial examples generation," in *Proc. of Great Lakes Symp. on VLSI*, pp. 449–454, 2021.
[9] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik *et al.,* "Practical black-box attacks against machine learning," in *Proc. of ACM Asia Conf. on Computer and Communications Security*, Dubai, United Arab Emirates, pp. 506–519, 2017.
[10] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.
[11] N. Papernot, P. D. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. of IEEE Symp. on Security and Privacy*, San Jose, USA, pp. 582–597, 2016.
[12] L. Gondara, "Detecting adversarial examples using density ratio estimates," arXiv preprint arXiv:1705.02224, 2017.
[13] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. of ACM SIGSAC Conf. on Computer and Communications Security*, Dallas, USA, pp. 135–147, 2017.
[14] X. Jia, X. Wei, X. Cao and H. Foroosh, "ComDefend: An efficient image compression model to defend adversarial examples," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, USA, pp. 6084–6092, 2019.
[15] W. Xu, D. Evans and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proc. of Network and Distributed System Security Symp.*, San Diego, USA, 2018.
[16] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial examples in the physical world," in *Proc. of Int. Conf. on Learning Representations*, Toulon, France, pp. 446–454, 2017.
[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. of IEEE Symp. on Security and Privacy*, San Jose, USA, pp. 39–57, 2017.
[18] P. Chen, B. Kung and J. Chen, "Class-aware robust adversarial training for object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 10420–10429, 2021.
[19] A. Zolfi, M. Kravchik, Y. Elovici and A. Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 15232–15241, 2021.
[20] A. Rakhsha, G. Radanovic, R. Devidze, X. Zhu and A. Singla, "Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning," in *Proc. of the 37th Int. Conf. on Machine Learning*, pp. 7974–7984, 2020.
[21] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. I. Al-Fuqaha *et al.,* "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 90–109, 2022.
[22] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. Gao *et al.,* "Improving transferability of adversarial patches on face recognition with generative models," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 11845–11854, 2021.
[23] S. Jia, C. Ma, T. Yao, B. Yin, S. Ding *et al.,* "Exploring frequency adversarial attacks for face forgery detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, USA, pp. 4093–4102, 2022.
[24] A. Almalawi, A. I. Khan, F. Alsolami, Y. B. Abushark and A. S. Alfakeeh, "Managing security of healthcare data for a modern healthcare system," *Sensors*, vol. 23, no. 7, pp. 3612, 2023.

[25] J. Liu, Y. Wang, Q. Han and J. Gao, "A sensitive image encryption algorithm based on a higher-dimensional chaotic map and steganography," *International Journal of Bifurcation and Chaos*, vol. 32, no. 1, pp. 2250004: 1–2250004: 22, 2022.
[26] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu *et al.,* "Boosting adversarial attacks with momentum," in *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 9185–9193, 2018.
[27] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," arXiv preprint arXiv:1412.5068, 2014.
[28] Y. M. Hu, "Research on D-D model defending adversarial examples based on deep residual learning denoising," M.S. Theses, Lanzhou University, China, 2018.
[29] C. Xie, J. Wang, Z. Zhang, Z. Ren and A. L. Yuille, "Mitigating adversarial effects through randomization," in *Proc. of the 6th Int. Conf. on Learning Representations*, Vancouver, Canada, pp. 1–16, 2018.
[30] R. C. Gonzalez and R. E. Woods, "Image restoration and reconstruction," in *Digital Image Processing*, 4th ed., New York, USA: Prentice Hall, Chapter 5, Section 5.3, pp. 327–349, 2018.
[31] A. Buades, B. Coll and J. Morel, "Non-local means denoising," *Image Process. Line*, vol. 1, pp. 208–212, 2011.
[32] K. Dabov, A. Foi, V. Katkovnik and K. O. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
[33] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
[34] Q. Chen and W. M. Pan, "Design and implementation of image denoising based on autoencoder," *Journal of Xinjiang Normal University (Natural Science Edition)*, vol. 37, pp. 80–85, 2018.
[35] J. Chen, J. Chen, H. Chao and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 3155–3164, 2018.
[36] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2578–2593, 2020.
[37] S. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 2574–2582, 2016.
[38] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman *et al.,* "Technical report on the cleverhans v2.1.0 adversarial examples library," arXiv preprint arXiv:1610.00768, 2016.