2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Adversarial Attacks and Defense for Deepfake Detection: A Comparative Research of Classical Classification Models

Xin Jin [1][0000−0003−2211−2006], Zhiyuan Li[1], Yuhao Xie [1], Bo Li [1], Cong Huang [2],
Xiaoyuan Xu [3], Ahmed Zahir [1], and Qian Jiang[1]

[1] School of Software, Yunnan University, Kunming, 650000, China
xinxin_jin@163.com, izhiyuanli@163.com,yuhaoxie0322@163.com,
1214801656@qq.com, zahir@ynu.edu.cn, jiangqian_1122@163.com
[2] University of Surrey Guildford, Surrey, UK
chong.huang@surrey.ac.uk
[3]School of Journalism, Yunnan University, Kunming 650091, Yunnan, China
xxy@ynu.edu.cn
(Corresponding author: Qian Jiang)

Abstract. With the widespread application of deep learning technologies such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), facial forgery techniques have matured rapidly, bringing innovative applications to multiple fields while also raising serious security concerns. To address this challenge, researchers have developed various deepfake detectors. However, these detectors have shown significant vulnerabilities when faced with adversarial attacks. This study aims to systematically evaluate the performance of deepfake detectors under adversarial attacks and test the effectiveness of various defense methods. Through large-scale experiments, we analyzed the performance of different types of detectors under various adversarial attacks and assessed the efficacy of existing defense strategies. The results indicate that while some defense methods perform well in specific scenarios, the overall robustness of detectors still needs improvement. This research not only deepens our understanding of adversarial robustness in deepfake detection but also provides important experimental evidence and theoretical guidance for developing more effective defense strategies.

Keywords: Internet of things, adversarial examples, object detection, computer vision, deep learning

1 Introduction

Benefiting from the significant advancements in deep learning technologies, particularly Convolutional Neural Networks (CNNs) and deep generative models such as Generative Adversarial Networks (GANs), Diffusion Models, DeepFake has emerged as a prominent topic of interest in recent years. With the continuous advancements in face generation technologies such as PGGAN [1] and StyleGAN3 [2], and face editing technologies like AttGAN [3] and StarGAN [4], the boundary between fake and real images is becoming increasingly indistinct, making it difficult for the human eyes to differentiate between them. These technologies have introduced numerous innovative and practical applications in various fields such as advertising and film production, as illustrated in Fig 1. However, while DeepFake technology offers limitless possibilities for enhancing various aspects of people's lives, it also poses potential threats to personal data privacy, social stability, and national security, as illustrated in Fig 1.

Fig. 1. The top half of the image shows the work of Surrealist painter Salvador Dalí, who was resurrected using DeepFake technology, and who was able to talk about his past and even interact with visitors at the Dalí Museum. The bottom half of the image shows American actor Jordan Peele playing Obama's speech using Deepfake technology.

With the widespread application of deepfake technology, effectively detecting these fake images has become an urgent issue. To address this challenge, researchers have developed various deepfake detectors that utilize deep learning models to identify subtle differences between fake and real images. Although these detectors perform excellently in many cases, they also face significant challenges, especially when confronting adversarial attacks. Adversarial attacks are a method of misleading deep learning models by adding carefully designed, minute perturbations to input data. For deepfake detectors, the threat of adversarial attacks is particularly severe, as these attacks can easily cause detectors to misclassify, allowing fake images to evade detection. This phenomenon has garnered widespread attention, prompting researchers to explore more robust defense mechanisms. To enhance the robustness of deepfake detectors, strategies for defending against adversarial attacks have gradually become a research focus. Although various defense methods have been proposed, their performance is inconsistent across different attack scenarios. Therefore, a systematic evaluation of the effectiveness of existing defense methods has become particularly necessary. This study aims to systematically evaluate the performance of existing deepfake detectors when faced with adversarial attacks through experimental analysis, and to test the effectiveness of various defense strategies. Through extensive experiments, we have conducted a detailed analysis of the performance of different detectors under various attacks and explored the advantages and disadvantages of these defense methods.

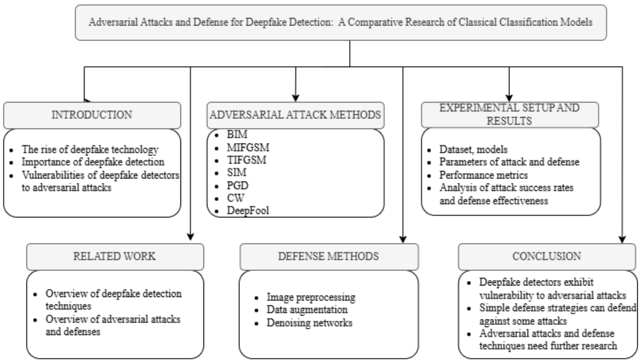
Adversarial Attacks and Defense for Deepfake Detection: A Comparative Research of Classical Classification Models
INTRODUCTION
The rise of deepfake technology
Importance of deepfake detection
Vulnerabilities of deepfake detectors to adversarial attacks
ADVERSARIAL ATTACK METHODS
BIM
MIFGSM
TIFGSM
SIM
PGD
CW
DeepFool
EXPERIMENTAL SETUP AND RESULTS
Dataset, models
Parameters of attack and defense
Performance metrics
Analysis of attack success rates and defense effectiveness
RELATED WORK
Overview of deepfake detection techniques
Overview of adversarial attacks and defenses
DEFENSE METHODS
Image preprocessing
Data augmentation
Denoising networks
CONCLUSION
Deepfake detectors exhibit vulnerability to adversarial attacks
Simple defense strategies can defend against some attacks
Adversarial attacks and defense techniques need further research

Fig. 2. Overview of Adversarial Attacks and Defense for Deepfake Detection.

2 Related Work

2.1 DeepFake

The term deepfake originates from a combination of deep learning and fake. This technology gained widespread attention in 2017 when a Reddit user named deepfakes posted videos where faces of famous actresses were superimposed onto those of adult performers, bringing public awareness to the potential impact of deepfake technology. Deepfake technology is inherently dual-use, with applications that can be either beneficial or harmful, depending on the intentions of the user. In recent years, while various face-swapping software and features have been introduced on entertainment media platforms, many were later forced to shut down due to information security concerns. Positive applications of deepfake technology include enhancing participant experience in video conferences [5] or creating special effects in film production, bringing beloved deceased actors back to the movie screen. However, the technology can also be maliciously exploited to create unauthorized pornographic content, manipulate recordings of political figures' statements or behaviors, or even to breach facial recognition systems for financial fraud and other criminal activities.

The malicious use of forged images can have severe negative impacts on individuals and society. To address this challenge, researchers have been dedicated to developing various detection methods to differentiate between real and forged images. These methods include traditional feature extraction-based algorithms and deep learning-based approaches. Traditional feature extraction algorithms analyze information such as image discontinuities [6], noise characteristics [7] and geometric transformations [8] to extract statistical features from the images. These features are then used with classifiers to determine whether an image has been forged. However, these methods often have limited effectiveness in detecting advanced forgery techniques. In contrast, deep learning-based methods have shown superior performance in forged image detection. These methods leverage the powerful learning capabilities of deep neural networks, training models to learn the differences between real and forged images and to classify them accordingly. Common deep learning models include Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs). CNNs automatically extract information such as texture, shape, and edges from raw images through hierarchical structures and convolution operations, learning to identify visual inconsistencies and anomalies in forged images. Deep learning-based methods have significantly improved the accuracy and robustness of detection tasks in identifying deepfake images.

Deepfake detection fundamentally involves a binary classification task, which entails distinguishing between "fake" and "real" images. This task typically employs Deep Neural Networks (DNNs), which learn to detect deeply falsified data through training on both "real" and "fake" data examples. Commonly used deep convolutional networks include VGG [9], ResNet [10], MesoNet [11], EfficientNet [12], Xception [13], DenseNet [14], MobileNet [15] and GramNet [16]. Although these models excel in this particular task, recent studies have revealed the potential vulnerability of DNNs to adversarial attacks.

2.2 Adversarial Attacks and Defense

Over the past decade, adversarial attacks have been extensively studied. Various neural network architectures, from ordinary image classifiers and deepfake detectors to video forgery detection systems and object detection models, have been proven vulnerable to adversarial vulnerabilities. These models can be easily deceived by carefully crafted adversarial examples, leading to incorrect predictions. Adversarial attacks are generally categorized into two main types: white-box attacks and black-box attacks.

In white-box attacks, the attacker has full access to all parameter information of the target model and can utilize detailed gradient information to finetune the perturbations. In contrast, black-box attacks lack direct access to the model's internal information and can be further divided into query-based attacks and transferability-based attacks. In query-based attacks, attackers create effective adversarial examples by submitting specific inputs to the target model and analyzing its responses. They infer the model's features and potential vulnerabilities by examining the correlation between input samples and their outputs. In transferability-based attacks, attackers leverage knowledge or adversarial examples obtained from other models, transferring this information to the target model to understand its behavior and implement attacks.

In response to adversarial attacks, researchers have proposed various defense strategies to enhance the robustness of deep learning models against adversarial examples. These defense methods can be broadly classified into two categories: input preprocessing defense methods and model internal enhancement defense methods.

The core idea of input preprocessing defense methods is to process the input data to eliminate or weaken adversarial perturbations. These methods include image smoothing techniques (such as median filtering, mean filtering, and Gaussian filtering) and denoising techniques (such as autoencoder-based denoising [18-20] and JPEG compression). These approaches can reduce the effectiveness of adversarial examples without significantly affecting normal samples. However, while preprocessing methods can mitigate attacks to some extent, they often show limitations when facing high-intensity perturbations. Model internal enhancement defense methods aim to improve the model's robustness through modifications in model architecture or training processes. Data augmentation defenses, such as AugMix [21], introduce diverse input samples during training to help the model learn more generalizable features, thereby enhancing robustness under adversarial perturbations. Adversarial Training is a widely used and effective defense strategy that incorporates adversarial examples directly into the training set, enabling the model to learn how to resist these attacks. Additionally, techniques like Gradient Masking increase the difficulty for attackers to generate adversarial examples by obscuring the model's gradient information. Defensive Distillation trains the target model to soften its outputs, making it challenging for attackers to generate effective adversarial perturbations using gradient information. Furthermore, recent years have seen the emergence of defenses based on high-level features, which focus on the robustness of high-level features in deep networks and can more effectively counter cross-model adversarial transfer attacks.

However, despite the fact that adversarial defense methods have improved model robustness to some extent, existing defense strategies still have significant limitations in the face of more complex and dynamic attack techniques. Defense mechanisms are often unable to comprehensively address different types of attacks, particularly showing unsatisfactory performance against black-box attacks.

3 Adversarial Attack Strategies

Adversarial attacks have evolved significantly, employing various sophisticated techniques to manipulate neural networks. This section examines several prominent attack strategies that have emerged in recent years. We will explore the Basic Iterative Method (BIM) [22], Momentum Iterative Fast Gradient Sign Method (MIFGSM) [23], Translation-Invariant Fast Gradient Sign Method (TIFGSM) [24], Scale-Invariant Method (SIM) [25], Projected Gradient Descent (PGD) [26], Optimization-based attack algorithms (C&W) [27], and DeepFool [28].

3.1 BIM

The Basic Iterative Method (BIM), also known as Iterative Fast Gradient Sign Method (I-FGSM), is an extension of the Fast Gradient Sign Method (FGSM). BIM applies FGSM multiple times with a small step size, clipping pixel values after each iteration to ensure the resulting adversarial example remains within an ϵ-neighborhood of the original image. The attack can be formalized as:

$$x_0^{\text{adv}} = x, \quad x_{t+1}^{\text{adv}} = \text{Clip}_x^{\epsilon}\{x_t^{\text{adv}} + \alpha \cdot \text{sign}\left(\nabla_x J\left(x_t^{\text{adv}}, y_{\text{true}}\right)\right)\} \qquad (1)$$

where x_t^{adv} is the adversarial example at iteration t, α is the step size, and $Clip_x^{\epsilon}$ performs per-pixel clipping to ensure the L_∞ distance between x and x^{adv} is at most ϵ. BIM typically produces stronger adversarial examples than FGSM due to its iterative nature.

3.2 MIFGSM

The Momentum Iterative Fast Gradient Sign Method (MIFGSM) integrates momentum into the iterative process of BIM to stabilize update directions and escape from poor local maxima. MIFGSM maintains a velocity vector that accumulates the gradient of previous iterations, which is then used to update the adversarial example:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J\left(x_t^{\mathrm{adv}}, y_{\mathrm{true}}\right)}{\left\|\nabla_x J\left(x_t^{\mathrm{adv}}, y_{\mathrm{true}}\right)\right\|_1}, \tag{2}$$

$$x_{t+1}^{\mathrm{adv}} = \mathrm{Clip}_x^{\epsilon}\{x_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sign}(g_{t+1})\}, \tag{3}$$

where μ is the decay factor for the momentum term. MIFGSM has demonstrated improved success rates in both white-box and black-box attack scenarios compared to BIM.

3. 3 **TIFGSM**

The Translation-Invariant Fast Gradient Sign Method (TIFGSM) addresses the vulnerability of adversarial examples to simple image transformations like translation. TIFGSM generates adversarial perturbations that remain effective under translation by convolving the gradient with a kernel W:

$$x_{t+1}^{\mathrm{adv}} = \mathrm{Clip}_x^{\epsilon}\{x_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sign}\left(W * \nabla_x J\left(x_t^{\mathrm{adv}}, y_{\mathrm{true}}\right)\right)\}, \tag{4}$$

The kernel W is typically chosen as a uniform kernel. This approach creates more robust adversarial examples that can maintain their effectiveness even when the input image is slightly translated.

3.4 SIM

The Scale-Invariant Method (SIM) extends the concept of TIFGSM to handle scale transformations. SIM generates adversarial perturbations that are effective across different scales by incorporating a set of scaling operations s into the gradient calculation:

$$x_{t+1}^{\mathrm{adv}} = \mathrm{Clip}_x^{\epsilon}\{x_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sign}\left(\sum_{s \in S} \nabla_x J\left(s\left(x_t^{\mathrm{adv}}\right), y_{\mathrm{true}}\right)\right)\}, \tag{5}$$

where $s(x)$ represents the scaling operation applied to the input x. By considering multiple scales during the attack, SIM produces adversarial examples that are more robust to scaling transformations.

3.5 PGD

Projected Gradient Descent (PGD) is a powerful iterative attack that can be seen as a variant of BIM with uniform random initialization. PGD starts from a random point within the allowed perturbation range and then iteratively applies FGSM followed by projection onto the allowed set:

$$x_0^{\text{adv}} = x + \text{Uniform}(-\epsilon, \epsilon), \tag{6}$$

$$x_{t+1}^{\text{adv}} = \text{Proj}_x^{\epsilon}\{x_t^{\text{adv}} + \alpha \cdot \text{sign}\left(\nabla_x J\left(x_t^{\text{adv}}, y_{\text{true}}\right)\right)\}, \tag{7}$$

where $Proj_x^{\epsilon}$ projects the perturbed image back onto the ϵ-ball centered at x. PGD is often considered one of the strongest first-order attacks and is frequently used for adversarial training.

3.6 C&W

The Carlini and Wagner (C&W) attack is an optimization-based method that aims to find the smallest perturbation that can mislead the target model. It formulates the problem as an optimization task:

$$\min_{\delta} \left||\delta|\right|_p + c \cdot f(x + \delta), \tag{8}$$

subject to $x + \delta \in [0,1]^n$, where f is a carefully designed loss function that encourages misclassification, c is a constant balanced through binary search, and p is typically 2 (L_2 norm) or ∞ (L_∞norm). C&W attacks are known for producing highly effective adversarial examples with small perturbations.

3.7 DeepFool

DeepFool is an iterative attack that seeks the minimal perturbation to cross the decision boundary of the classifier. For binary classifiers, it approximates the decision boundary with a hyperplane and moves the input towards the closest point on this hyperplane. For multi-class classifiers, it iteratively moves towards the nearest decision boundary. The update step can be expressed as:

$$x_{t+1}^{\text{adv}} = x_t^{\text{adv}} + \frac{\left|f\left(x_t^{\text{adv}}\right)\right|}{\left|\left|\nabla f\left(x_t^{\text{adv}}\right)\right|\right|_2^2} \nabla f\left(x_t^{\text{adv}}\right), \tag{9}$$

where f is the decision function of the classifier. DeepFool often produces smaller perturbations compared to other methods while maintaining a high fooling rate.

4 Defense Strategies

Deep neural networks face the threat of adversarial examples, so more and more researchers are actively exploring the field of adversarial defense, considering improving the robustness of models as a top priority. This section will focus on image preprocessing methods, data enhancement [21] and Denoising Network [18-20].

4.1 Image Preprocessing

Image preprocessing techniques play a crucial role in enhancing the robustness of deep learning models against adversarial attacks. By applying various smoothing methods to input images, we can reduce the effectiveness of perturbations introduced by attackers.

In this section, we discuss three commonly used smoothing techniques: Median Smoothing, Gaussian Smoothing, Average Smoothing and JPEG Compression.

Median Smoothing Median Smoothing is a non-linear filtering technique commonly used to remove noise from images. This method replaces each pixel's value with the median value of the pixels in its surrounding neighborhood. The median is determined by sorting the pixel values within the neighborhood and selecting the middle value. Median Smoothing is particularly effective in eliminating 'salt-and-pepper' noise, which manifests as random white and black pixels scattered throughout the image. This technique is beneficial for adversarial defense as it can diminish the impact of isolated pixel perturbations introduced by adversarial attacks without significantly blurring the image.

Gaussian Smoothing Gaussian Smoothing, also known as Gaussian Blur, is a linear filtering technique that reduces image noise and detail by averaging the pixels within a neighborhood weighted by a Gaussian function. The Gaussian function gives more weight to pixels closer to the central pixel, leading to a more natural and less harsh blurring effect compared to simple averaging. The degree of smoothing is controlled by the standard deviation (δ) of the Gaussian distribution. Gaussian Smoothing is effective in reducing high-frequency noise and perturbations, making it a useful preprocessing step in adversarial defense by blurring out subtle adversarial perturbations while maintaining the overall structure of the image.

Average Smoothing Average Smoothing, also known as Mean Filtering, is the simplest form of smoothing technique. It works by replacing each pixel's value with the average value of the pixels in its surrounding neighborhood. This technique effectively reduces random noise by averaging out pixel values, leading to a smoother image. However, it may also blur sharp edges and details, making it less suitable for preserving fine image structures. Despite this, Average Smoothing can still be useful in adversarial defense by reducing the impact of widespread perturbations across the image.

JPEG Compression JPEG Compression is a widely used image compression technique that can also serve as an effective adversarial defense method. This process involves converting the image into the frequency domain using the Discrete Cosine Transform (DCT), quantizing the DCT coefficients, and then encoding the quantized values. By reducing the precision of the DCT coefficients, JPEG Compression effectively removes high-frequency components, including many adversarial perturbations. The degree of compression can be adjusted by changing the quality factor, which controls the level of quantization. JPEG Compression is advantageous in adversarial defense as it can significantly reduce the influence of subtle adversarial noise while retaining the essential visual information of the image.

4.2 Data Enhancement

AugMix [21] AugMix is an innovative data augmentation technique designed to enhance the robustness of deep learning models and improve their generalization to uncertain data. In real-world applications, the distribution of training data often differs from that of test data, which can lead to significant degradation in model performance. AugMix addresses this challenge through a unique approach, opening new possibilities in the field of defense strategies. Unlike traditional data augmentation methods, AugMix employs a more complex and effective processing workflow. It begins by applying multiple parallel augmentation operations to the original image, such as geometric transformations, rotations, and color adjustments. These augmented images are then combined according to specific proportions, creating a new augmented version. Finally, this new version is mixed with the original image at a predetermined ratio to generate the final augmented sample. The core advantage of AugMix lies in its ability to generate samples that maintain high similarity to the original image, effectively avoiding excessive distortion. Furthermore, through the combination of various augmentation operations, it offers a richer range of possible variations. The randomness in mixing ratios further increases data diversity, exposing the model to a broader distribution of data.

By applying AugMix during the training process, models can learn more robust feature representations. This enables models to perform better when faced with unknown data distributions, thereby enhancing their adaptability and defensive capabilities in practical applications.

4.3 Denoising Network

DAE[18] The Denoising Autoencoder (DAE) is an unsupervised learning algorithm that has been adapted as a strategy for adversarial defense. Originally designed for noise reduction and feature extraction, DAEs have shown promise in mitigating the effects of adversarial perturbations on neural networks.

At its core, DAE is a type of neural network that learns to reconstruct clean data from corrupted inputs. This principle aligns well with the challenge of adversarial examples, where imperceptible perturbations can significantly alter a model's predictions. In the context of adversarial defense, the DAE is trained to remove these malicious perturbations, effectively denoising the adversarial examples.

The structure of DAE typically consists of two main components: an encoder and a decoder. The encoder compresses the input data into a lower dimensional latent representation, while the decoder reconstructs the output data from this latent space. During training, the DAE learns to minimize the reconstruction error between its output and the original, uncorrupted input.

HGD[19] Before discussing the High-Level Representation Guided Denoiser (HGD), let's first review the previously proposed Pixel Guided Denoiser (PGD). Let x represent the original clean image, x^* the adversarial example, and x' the denoised image.

PGD defines the loss function as $L = |x - x'|$, where $|\cdot|$ denotes the L1 norm. Since this loss function is defined at the image pixel level, it is named the pixel-guided denoiser. However, PGD has a fatal flaw. Denoising is relative, not absolute, and no matter how perfect the denoising process is, there will always be residual noise in the image. Moreover, there exists an error amplification effect in DNNs. That is, the residual adversarial noise is amplified layer by layer, resulting in sufficient noise in the final output to cause DNN misclassification.

To address this issue, liao [19] proposed the HGD method. Considering the error amplification effect in DNNs, HGD defines the loss function at the output layer. Let y represent the output of the original clean image through the DNN, y^* the output of the adversarial example, and $y^{'}$ the output of the denoised image. HGD defines the loss function as $L = |y - y'|$, representing the difference between the outputs of the denoised image and the original image. The goal of HGD is to minimize this loss function. The smaller the loss function, the smaller the difference between the output of the denoised image and the original image, indicating a closer approximation to the initial image and better denoising effect.

HGD is further categorized into three types based on the position and type of the added loss function. First is the FGD (Feature Guided Denoiser). Let $l = -2$ be defined as the index of the topmost convolutional layer, after which the activations are fed into the linear classification layer. Therefore, compared to lower convolutional layers, it is more related to the classification objective. The loss function used in FGD is also known as perceptual loss or feature matching loss. The second type is LGD (Logit Guided Denoiser). Let $l = -1$ be defined as the index of the layer before the final softmax function, i.e., the logits. The loss function here is the difference between the logits of x and x' activations. The last type is CGD (Class Guided Denoiser). This method uses the classification loss of the target model as the denoising loss function, which is a supervised learning method requiring true labels.

In this study, we chose to use the LGD as our HGD implementation method. This approach directly optimizes the logits at the output layer, which can more effectively reduce the impact of adversarial noise on the final classification results.

TD[20] The TD method is a semi-supervised learning approach designed to effectively remove adversarial perturbations while preserving the original attributes of the input image. It comprises two main modules: a reconstruction module and a denoising module. These modules work together to ensure the method's effectiveness in removing adversarial noise while maintaining the original features of the input image. TD employs an encoder-decoder architecture for its denoiser. Specifically, the denoiser consists of an encoder G_{enc} and a decoder G_{dec}. Given an input image $x \in R^m$, the encoder G_{enc} encodes it into a latent representation, which is then decoded by the decoder G_{dec} into a reconstructed image. Theoretically, denoising learning can use any high-level representation to highlight the differences between adversarial and original samples, such as the topmost convolutional layer, logits layer, or final softmax layer. In this study, we chose the logits layer to assist denoising learning because the output of the logits layer is simpler and exhibits larger value differences. The reconstruction module ensures that the original features of the input image are not lost during the reconstruction process, while the denoising module is responsible for effectively removing adversarial perturbations. During training, each batch contains clean images and their corresponding adversarial images, with the clean images serving as labels. Simultaneously, random Gaussian noise is added to the latent representation to achieve good generalization. The TD method is a semi-supervised learning approach with good transferability. This means that denoised samples should also possess transferability, i.e., the denoising effect of the samples should be maintained when applied to other models. This characteristic makes TD a promising method for defending against adversarial attacks, especially in scenarios involving unknown models or black-box attacks.

5 Experiments

In this section, we provide a comprehensive overview of the experiments conducted to evaluate the robustness of DeepFake detectors against adversarial attacks and the effectiveness of various defense mechanisms.

5.1 Dataset

FaceForensics++ [29] is a popular dataset containing real videos from YouTube and their corresponding fake versions. This dataset includes various manipulation methods, such as DeepFake, Face2Face, FaceSwap, and NeuralTextures. We process the video frames to create a training set, a validation set, and a test set. The training set contains a total of 180,000 real and fake images. We also created a smaller test set to evaluate the attack algorithm. From this test set, we selected 1,000 images, consisting of 500 real and 500 fake images. The fake images were further divided into 4 subgroups, with each subgroup containing 125 images created using a different forgery technique. In total, our test set includes 500 authentic images and 500 synthesized images across 4 categories of deepfakes.

5.2 DeepFake Detectors

In this study, we have adopted several popular Deep Neural Network (DNN) based detectors to identify and analyze DeepFake images. These detectors include VGG16 [9], ResNet50 [10], MesoNet, MesoInception [11], EfficientNet [12], Xception [13], DenseNet121 [14], MobileNetV2 [15] and GramNet [16]. Each of these models brings unique architectural advantages and has demonstrated efficacy in the task of DeepFake detection.

5.3 Details of the Attack Implementation、

For the models, we use an input image size of 224×224×3. We have chosen the adversarial attack algorithms, such as BIM, MIFGSM, TIFGSM, SIM, PGD, C&W and DeepFool. For each attack method, in addition to varying the perturbation magnitude ϵ

(with a maximum value of 16/255), we use the optimal hyperparameters as specified in their respective original publications. These attacks are attacked with full knowledge of the model, i.e., white-box attacks.

5.4 Details of the Defense Implementation

For adversarial defense methods, we selected algorithms from three perspectives. From the image processing perspective, we chose Median Smoothing, Gaussian Smoothing, Average Smoothing, and JPEG compression. From the data augmentation perspective, we selected AugMix and from the denoising network perspective, we opted for DAE, HGD, and TD.

For various smoothing defenses of the image, the size of 3*3 convolution kernel is what we used. When using JPEG compression, we set the quality to 70, 80, and 90 to test the defense of JPEG compression at different quality.

When using AugMix for data augmentation to enhance the robustness of the deepfake detector, we strictly followed the parameters provided in the original paper, but the epoch was set to 20 to ensure the same parameters as when training the baseline model. The accuracy of the deepfake detector after data enhancement by AugMix is shown in Table 1. The base represents the accuracy of the benchmark model. Bolding in the table represents the optimal value.

When training the denoising network, the input image size is 224*224*3, the learning rate is set to 1e-3, and epoch are all set to 20, we use the optimal hyperparameters as specified in their respective original publications.

Table 1. Accuracy of Deepfake Detectors

Model\ACC	Base	AugMix [21]
VGG16 [9]	0.955	0.960
ResNet50 [10]	**0.939**	**0.939**
MesoNet [11]	0.774	**0.807**
MesoInception[11]	**0.875**	0.839
EfficientNetb4[12]	0.939	**0.962**
Xception [13]	0.945	**0.958**
DenseNet121 [14]	0.948	**0.960**
MobileNetV2 [15]	0.953	**0.961**
GramNet [16]	**0.953**	0.952

6 Analysis of Experimental Results

The experimental results on counter-attack algorithms and defense measures are shown in Table 2, where NA stands for no defense measures are performed, MS, GS and AS stand for Median Smoothing, Gaussian Smoothing, Average Smoothing. From the column in Table 2 NA, we can see that any of the adversarial attack algorithms, with knowledge of the model, are powerful enough to almost completely tamper with the model output.

6.1 From the perspective of image preprocessing

Simple image smoothing techniques, such as Median Smoothing, Gaussian Smoothing, and Average Smoothing, demonstrate varying degrees of defensive capability against certain adversarial attacks, but their overall effectiveness remains limited. These techniques are generally more effective in addressing non-structured noise perturbations; however, their defensive capabilities are relatively limited against the structured perturbations introduced by many advanced adversarial attacks. Experimental results indicate that these smoothing techniques exhibit better defensive performance against DeepFool attacks. This may be attributed to the fact that the DeepFool algorithm aims to push images towards the decision boundary with minimal perturbations, which are more easily eliminated or weakened by smoothing processes. However, for attack algorithms like PGD or BIM that introduce larger magnitude and more structured perturbations, simple smoothing techniques often struggle to provide effective defense. This disparity can be explained by the nature of adversarial perturbations. Advanced adversarial attacks typically generate structured perturbations that correspond to the critical features relied upon by neural networks. These perturbations are meticulously designed to maximize their impact on the model's decision-making process while maintaining visual similarity. In contrast, simple smoothing techniques primarily target random noise and struggle to effectively remove these highly coupled, structure-based perturbations.

In conclusion, while image smoothing techniques can provide a certain level of defense in some scenarios, their effectiveness is highly dependent on the characteristics of the attack.

JPEG70, JPEG80, and JPEG90 represent JPEG processing at different compression rates, with higher numbers indicating lower compression rates and greater preservation of image details. As JPEG quality increases, the loss of image details due to compression decreases, and more of the adversarial perturbations are also retained. Consequently, JPEG90 typically offers less effective defense against adversarial attacks compared to JPEG70, but JPEG90 better maintains the visual quality and details of the original image. JPEG compression defense primarily works by reducing high-frequency components in the image. This method is particularly effective against attacks that mainly rely on high-frequency perturbations, as high-frequency components often contain many adversarial disturbances. However, for attacks that have widely distributed perturbations and do not solely depend on high frequencies, JPEG compression's defensive effectiveness is relatively lower. This is because such attacks may introduce significant perturbations in low and mid-frequency ranges as well, which are not easily eliminated by JPEG compression.

It's worth noting that JPEG compression as a defense mechanism presents a trade-off: higher compression rates may provide better defensive effects but also lead to greater loss of image details, potentially affecting the model’s normal recognition capabilities. Conversely, lower compression rates preserve more image details and potential adversarial perturbations but may still offer a degree of defense in certain situations while minimizing the impact on the original image quality.

Table 2. Adversarial Attacks and Defense Results

Models	Methods\ACC	NA	MS	GS	AS	JPEG70	JPEG80	JPEG90	AugMix [21]	DAE [18]	HGD [19]	TD [20]
VGG16[9]	BIM [22]	0.087	0.087	0.111	0.087	0.349	0.092	0.221	0.168	0.443	0.500	0.568
	MIFGSM[23]	0.091	0.092	0.111	0.087	0.050	0.088	0.090	0.116	0.087	0.894	0.870
	TIFGSM [24]	0.091	0.091	0.092	0.087	0.044	0.087	0.091	0.085	0.087	0.501	0.508
	SIM [25]	0.001	0.008	0.048	0.003	0.102	0.002	0.001	0.021	0.000	0.702	0.757
	PGD [26]	0.000	0.180	0.118	0.034	0.486	0.129	0.000	0.819	0.715	0.998	0.999
	C&W [27]	0.000	0.021	0.069	0.136	0.473	0.264	0.076	0.239	0.501	0.500	0.502
	DeepFool[28]	0.023	0.663	0.549	0.521	0.510	0.465	0.216	0.925	0.503	0.500	0.502
ResNet50 [10]	BIM [22]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.024	0.025	0.979	0.955
	MIFGSM[23]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.497	0.508
	TIFGSM [24]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.239	0.164
	SIM [25]	0.000	0.004	0.000	0.000	0.011	0.003	0.000	0.001	0.000	0.482	0.489
	PGD [26]	0.000	0.150	0.062	0.070	0.007	0.000	0.000	0.183	0.265	0.999	1.000
	C&W [27]	0.000	0.023	0.047	0.067	0.160	0.077	0.028	0.416	0.498	0.532	0.527
	DeepFool[28]	0.187	0.803	0.576	0.553	0.473	0.280	0.092	0.906	0.517	0.608	0.528
MeseNet [11]	BIM [22]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.007	0.001	0.722	0.647
	MIFGSM[23]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.008	0.000	0.565	0.644
	TIFGSM [24]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.007	0.000	0.230	0.455
	SIM [25]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.007	0.433	0.390
	PGD [26]	0.000	0.000	0.000	0.028	0.000	0.000	0.000	0.045	0.378	0.896	0.867
	C&W [27]	0.134	0.213	0.223	0.238	0.349	0.281	0.246	0.373	0.427	0.621	0.536
	DeepFool[28]	0.130	0.580	0.441	0.555	0.470	0.336	0.150	0.761	0.524	0.629	0.558
MesoInception [11]	BIM [22]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.376	0.935	0.991
	MIFGSM[23]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.699	0.582
	TIFGSM [24]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.007	0.000	0.082	0.922
	SIM [25]	0.000	0.000	0.010	0.002	0.000	0.000	0.000	0.007	0.000	0.682	0.548
	PGD [26]	0.000	0.002	0.000	0.139	0.000	0.000	0.000	0.043	0.459	0.974	1.000
	C&W [27]	0.010	0.072	0.194	0.417	0.429	0.326	0.182	0.285	0.505	0.515	0.505
	DeepFool[28]	0.091	0.690	0.544	0.527	0.560	0.498	0.254	0.786	0.507	0.508	0.503
Efficient-Netb4[12]	BIM [22]	0.000	0.022	0.000	0.016	0.000	0.000	0.000	0.136	0.195	1.000	1.000
	MIFGSM[23]	0.000	0.000	0.000	0.016	0.000	0.000	0.000	0.134	0.084	0.677	0.499
	TIFGSM [24]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.035	0.000	0.551	0.612
	SIM [25]	0.017	0.000	0.000	0.000	0.018	0.017	0.017	0.065	0.024	0.784	0.500
	PGD [26]	0.000	0.333	0.079	0.436	0.000	0.000	0.000	0.621	0.605	1.000	0.999
	C&W [27]	0.000	0.041	0.082	0.135	0.186	0.073	0.031	0.269	0.502	0.500	0.461
	DeepFool[28]	0.040	0.707	0.616	0.585	0.426	0.240	0.080	0.910	0.512	0.500	0.484
Xception [13]	BIM [22]	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.021	0.206	0.978	0.474
	MIFGSM[23]	0.001	0.001	0.001	0.001	0.010	0.010	0.010	0.021	0.050	0.683	0.001
	TIFGSM [24]	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.015	0.045	0.640	0.001
	SIM [25]	0.000	0.000	0.000	0.000	0.002	0.002	0.000	0.021	0.049	0.572	0.001

	PGD [26]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.235	0.296	1.000	0.997
	C&W [27]	0.000	0.026	0.050	0.077	0.161	0.045	0.014	0.369	0.487	0.517	0.347
	DeepFool[28]	0.145	0.786	0.624	0.579	0.461	0.256	0.062	0.935	0.504	0.550	0.739
Dense-Net121[14]	BIM [22]	0.007	0.007	0.006	0.006	0.006	0.006	0.006	0.022	0.013	0.989	0.989
	MIFGSM[23]	0.007	0.007	0.006	0.006	0.007	0.007	0.007	0.006	0.006	0.583	0.945
	TIFGSM [24]	0.007	0.007	0.006	0.006	0.007	0.007	0.007	0.004	0.006	0.627	0.854
	SIM [25]	0.000	0.000	0.000	0.000	0.007	0.000	0.000	0.015	0.000	0.571	0.552
	PGD [26]	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.239	0.060	0.999	1.000
	C&W [27]	0.000	0.030	0.055	0.070	0.160	0.053	0.010	0.462	0.499	0.542	0.552
	DeepFool[28]	0.168	0.762	0.627	0.609	0.492	0.270	0.076	0.938	0.505	0.597	0.535
Mo-bileNetV2[15]	BIM [22]	0.007	0.007	0.011	0.019	0.008	0.023	0.009	0.148	0.152	0.500	0.930
	MIFGSM[23]	0.007	0.023	0.010	0.007	0.036	0.006	0.006	0.132	0.058	0.501	0.779
	TIFGSM [24]	0.007	0.007	0.007	0.007	0.006	0.006	0.006	0.029	0.007	0.529	0.512
	SIM [25]	0.009	0.011	0.018	0.026	0.016	0.014	0.025	0.020	0.007	0.502	0.894
	PGD [26]	0.000	0.086	0.090	0.114	0.333	0.318	0.004	0.539	0.658	0.999	0.997
	C&W [27]	0.095	0.110	0.137	0.196	0.307	0.160	0.114	0.326	0.513	0.486	0.502
	DeepFool[28]	0.145	0.744	0.697	0.616	0.548	0.395	0.113	0.915	0.531	0.502	0.500
GramNet [16]	BIM [22]	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.017	0.142	0.992	0.982
	MIFGSM[23]	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.029	0.006	0.977	0.981
	TIFGSM [24]	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.016	0.006	0.519	0.564
	SIM [25]	0.023	0.001	0.001	0.000	0.056	0.042	0.029	0.028	0.000	0.948	0.990
	PGD [26]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.162	0.310	0.986	0.980
	C&W [27]	0.000	0.019	0.042	0.062	0.107	0.036	0.020	0.248	0.511	0.585	0.549
	DeepFool[28]	0.082	0.795	0.620	0.582	0.434	0.221	0.038	0.915	0.515	0.540	0.537

6. 2 From the perspective of data enhancement

Data augmentation, by increasing the diversity of training data, enhances the robustness of models, making it a relatively effective defense strategy. As shown in Table 1, models generally exhibit significant accuracy improvements after AugMix data augmentation. Moreover, as shown in Table 2, AugMix demonstrates a certain degree of defensive effectiveness. This indicates that data augmentation not only boosts model performance on clean data but also strengthens its resilience against adversarial attacks. However, while data augmentation provides better defense against various adversarial attacks compared to simple smoothing methods, it still falls short of the effectiveness seen with specifically designed denoising defense networks. Data augmentation heavily relies on increasing the diversity of the training data, which might limit its effectiveness against certain specially crafted attacks, particularly when adversarial examples differ significantly from real data.

6.3 From the perspective of denoising network

The DAE builds upon the traditional autoencoder by adding noise to the input and then reconstructing the clean input from these noisy corrupted examples. As shown in Table 2, the DAE method demonstrates a certain level of defensive capability, indicating its ability to resist adversarial attacks to some extent. This lays the groundwork for subsequent defensive methods.

The results in Table 2 show that both HGD and TD exhibit relatively consistent and superior performance across different models and attack methods. This suggests that these methods might capture some essential features of adversarial examples and utilize these features to distinguish between normal and adversarial examples. Specifically, HGD leverages high-level features to guide the denoising process, indicating that adversarial perturbations may share certain common patterns in high-level feature spaces. Focusing on high-level semantic features, rather than pixel-level changes, could be the key to designing effective defenses. High-level features often correspond to more abstract semantic information, which may exhibit greater stability when facing different attacks. Therefore, HGD can achieve robust defensive performance across various attacks.

On the other hand, the design goal of the TD method is to achieve transferable denoising, and this transferability may be the reason for its outstanding performance across different scenarios. Transferable defense methods are particularly important for practical applications, as they can better address unknown or novel attacks. Compared to defenses designed for specific attacks, transferable methods can provide broader and more enduring defensive capabilities.

7 Conclusion

In this study, we systematically explored the effectiveness of adversarial attacks and defense methods in the task of deepfake detection. Through experimental analysis, we evaluated the performance of deepfake detectors when faced with various adversarial attacks and tested the efficacy of different defense strategies. The results indicate that deepfake detectors exhibit significant vulnerability to adversarial attacks, while applying specific defense strategies notably enhances their robustness. Notably, certain advanced defense methods demonstrated strong consistency and effectiveness in addressing different types of attacks, suggesting their ability to effectively capture and utilize key features of adversarial perturbations. This research not only deepens our understanding of adversarial robustness in deepfake detection but also provides strong support for developing more robust and efficient defense methods in the future.

Acknowledgments. This study is supported by the Basic Research Project of Yunnan Province (Nos. 202401AT070470, 202201AU070033).

References

1. T. Karras, T. Aila, S. Laine, et al. "Progressive growing of gans for improved quality, stability, and variation," 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018.

2. T. Karras, M. Aittala, S. Laine, et al. “Alias-free generative adversarial networks,” in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021.
3. Z. He, W. Zuo, M. Kan, et al. “Attgan: Facial attribute editing by only changing what you want,” IEEE Trans. Image Process, 2019, 28(11): 5464–5478.
4. Y. Choi, M. Choi, M. Kim, et al. “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018.
5. T. Wang, A. Mallya, and M. Liu, “One-shot free-view neural talking-head synthesis for video conferencing,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pp. 10 039–10 049.
6. A. C. Popescu and H. Farid, “Exposing digital forgeries by detecting traces of resampling,” IEEE Trans. Signal Process., vol. 53, no. 2-2, pp. 758–767, 2005.
7. D. Cozzolino and L. Verdoliva, “Noiseprint: A cnn-based camera model fingerprint,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 144–159, 2020.
8. M. K. Johnson and H. Farid, “Exposing digital forgeries through chromatic aberration,” in Proceedings of the 8th workshop on Multimedia & Security, MM&Sec 2006, Geneva, Switzerland, September 26-27, 2006, S. Voloshynovskiy, J. Dittmann, and J. J. Fridrich, Eds. ACM, 2006, pp. 48–55.
9. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015.
10. K. He, X. Zhang, S. Ren, et al. “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 770–778.
11. D. Afchar, V. Nozick, J. Yamagishi, et al. “Mesonet: a compact facial video forgery detection network,” in 2018 IEEE International Workshop onInformation Forensics and Security, WIFS 2018, Hong Kong, China, December 11-13, 2018, pp. 1–7.
12. M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol.97. PMLR, 2019, pp. 6105–6114.
13. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 1800–1807.
14. G. Huang, Z. Liu, L. van der Maaten,et al. “Densely connected convolutional networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 2261–2269.
15. M. Sandler, A. G. Howard, M. Zhu, et al. “Mobilenetv2: Inverted residuals and linear bottlenecks,” in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 4510– 4520.
16. Z. Liu, X. Qi, and P. H. S. Torr, “Global texture enhancement for fake face detection in the wild,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 2020, pp. 8057–8066.
17. LA. Ilyas, L. Engstrom, A. Athalye, et al. “Black-box adversarial attacks with limited queries and information,” in Proceedings of the 35th International Conference on Machine

Learning, ICML 2018, Stockholmsmassan, ̈Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 2142–2151.

18. S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015.
19. F. Liao, M. Liang, Y. Dong, et al. "Defense against adversarial attacks using high-level representation guided denoiser," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 1778–1787.
20. S. Gao, S. Yao, and R. Li, "Transferable adversarial defense by fusing reconstruction learning and denoising learning," in 2021 IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2021, Vancouver, BC, Canada, May 10-13, 2021. IEEE, 2021, pp. 1–6.
21. D. Hendrycks, N. Mu, E. D. Cubuk,et al. "Augmix: A simple data processing method to improve robustness and uncertainty," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.
22. A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017.
23. Y. Dong, F. Liao, T. Pang,et al."Boosting adversarial attacks with momentum," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 9185–9193.
24. Y. Dong, T. Pang, H. Su, et al. "Evading defenses to transferable adversarial examples by translation-invariant attacks," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, pp. 4312–4321.
25. J. Lin, C. Song, K. He, et al. "Nesterov accelerated gradient and scale invariance for adversarial attacks," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.
26. A. Madry, A. Makelov, L. Schmidt, et al."Towards deep learning models resistant to adversarial attacks," in 6th International Conference on Learning Representations, April 30 - May 3, 2018.
27. N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in 2017 IEEE Symposium on Security and Privacy, 2017, pp. 39–57.
28. S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
29. A. Rossler, D. Cozzolino, L. Verdoliva,et al."Faceforensics++: Learning to detect manipulated facial images," in IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.
30. S. Mercan, M. Cebe, R. S. Aygun, et al. "Blockchain-based video forensics and integrity verification framework for wireless internet-of-things devices," Secur. Priv., vol. 4, no. 2, 2021.