An Adversarial Example Defense Algorithm for Intelligent Driving

Jiazhong Lu, Chenli Wang, Yuanyuan Huang, Kangyi Ding, and Xiaolei Liu

Abstract

In terms of intelligent driving, the adversarial example of an attack against traffic signs will cause the vehicle to make wrong judgments and decisions. However, the existing adversarial examples of defense algorithms generally have problems such as high training costs and poor defense effects and struggle to adapt to the environment of intelligent driving. In order to reduce the training cost while improving the accuracy of example classification, we propose a novel defense algorithm for adversarial examples combining micro-network structure and a generative adversarial network (GAN). The algorithm compresses the classification model and the Discriminator. And the Generator is designed to make the reconstructed sample generated closer to the real example distribution so as to solve the common problems of the existing adversarial example defense algorithm. Experiments on the collected traffic sign data set show that the proposed algorithm can achieve a better defense effect on the premise of lower training costs. The example classification accuracy can reach more than 97.9%, and the similarity between the reconstructed samples and the real examples reaches 96.26%. Moreover, the number of computations and parameters for training a single example is far lower than that of other commonly used defense methods, and the response speed is approximately doubled, which can greatly improve the safety of intelligent driving.

Introduction

At present, much progress and breakthroughs have been made in the field of intelligent driving. But intelligent driving needs to rely on artificial intelligence technology to process and analyze a large amount of data obtained by sensors for decision-making and planning. Therefore, the development of intelligent driving is also facing challenges and difficulties. The security of intelligent driving technology has always been a hot topic, one of which is adversarial example attacks. Adversarial examples mean that the attacker misleads the model to produce wrong classification results by adding perturbations to the data set samples, which can lead to extremely serious consequences. In real-world scenarios, intelligent driving vehicles need to recognize and understand traffic signs on the road, which are crucial for the vehicle's driving direction, speed limit, and road traffic safety. However, if the attacker makes a small modification to the traffic sign to make it an adversarial example, the self-driving car will make a wrong decision, which will cause serious traffic accidents and casualties.

So far, researchers have proposed various defense algorithms against adversarial examples in the field of intelligent driving, including training data sets, compressing data, adversarial example reconstruction, etc. However, these methods have some limitations, to varying degrees. For instance, the defense range is limited, the accuracy is low, and the training cost is high. Compared with other defense algorithms, defense algorithms based on GAN will "recover" the adversarial example into reconstructed samples that are almost consistent with the distribution of the original samples. This will prevent the attacker from successfully attacking the model, so defenses based on GAN can provide a comprehensive defense and will not cause a waste of data resources. Nonetheless, such defense methods require large amounts of training data and computing resources, and the training process may be very expensive. And there is a slight error between the reconstructed samples and the original samples, which will lead to a low accuracy of sample classification. Our proposed algorithm can ensure the quality and high classification accuracy of reconstructed samples while using a small training cost.

In recognizing traffic sign images, the traditional classification model is responsible for extracting image spatial features from the convolutional layer and classifying them through the fully connected layer. However, when using the fully connected layer for computation, the 3D data needs to be flattened into 1D data. During this process, the adjacent elements in the input image may lose their spatial adjacency due to the flattening operation. Therefore, convolutional layers can more correctly understand types of data, such as images. Additionally, the fully connected layer involves a large number of parameters and computations in the calculation process, leading to high training costs.

Based on the micro-network structure [1], we proposed a new defense algorithm using GAN, addressing the existing issues of high training costs and low accuracy in adversarial example

Digital Object Identifier: 10.1109/MNET.2024.3392582
Date of Current Version: 18 November 2024
Date of Publication: 23 April 2024

Jiazhong Lu, Chenli Wang, and Yuanyuan Huang are with the School of Cybersecurity (Xin Gu Industrial College), Chengdu University of Information Technology, Chengdu 610225, China, also with the Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610225, China, and also with the SUGON Industrial Control and Security Center, Chengdu 610225, China; Kangyi Ding and Xiaolei Liu (corresponding author) are with the Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan 621900, China.

defense. We first delve into the potential threats posed by adversarial example attacks on autonomous driving and road safety. Subsequently, we categorize, analyze, and summarize the existing adversarial example defense methods, along with their associated flaws, providing readers with a comprehensive understanding of the current state of the field. The algorithm section is aimed at addressing the shortcomings of current adversarial example defense methods in the context of autonomous driving and provides the algorithm's workflow and model structure. We then proceed to compare and analyze various parameters. Finally, we summarize the research outcomes and propose further research directions. Our main contributions are as follows:

(1) Inspired by the micro-network structure, we introduce the micro-network structure into the traffic sign adversarial example defense model. The experimental models all adopt the micro-network structure, which greatly reduces the training cost. On the premise of using the collected traffic sign image set to train the model, the computations and parameters needed for each individual sample in our defense algorithm are greatly reduced.

(2) The Generator is composed of 3 layers of convolutional layers and 3 layers of deconvolutional layers with a step, and the reconstructed sample of the generated traffic signs is closer to the real sample distribution.

(3) The *MN-GAN* algorithm uses a micro-network structure to design a classifier and Discriminator. And we design a Generator model with a stride of 2 in the algorithm, which makes the sample classification accuracy rate reach 97.9%.

Related Work

The defense methods currently proposed by researchers mainly include three directions. This section briefly describes the research status of each defense direction.

- Defense Algorithms Based on Input: Currently, the most widely used is the data augmentation technique. Goodfellow et al. [2] first found through experiments that the trained model on a dataset with adversarial examples can reduce the probability of the model producing false negative results for adversarial examples. This marks the inception of adversarial training. This method enables the model to correctly identify adversarial examples and gain defense capabilities. Shafahi et al. [3] proposed Free Adversarial Training (FressAT). The idea is to directly use the gradient information to generate adversarial examples to reduce the computational cost of generating adversarial examples. Misclassification Aware Adversarial Training (MART) proposed by Wang et al. [4] and Triplet Loss Adversarial (TLA) proposed by Mao et al. [5] both use regularization to optimize robust error judgments to improve adversarial training. In addition to this, data compression is also one of the classic methods, which reduces the interference of adversarial noise by compressing the sample data. This type of defense method will reduce the model's recognition accuracy for benign samples. Moreover, the defense algorithm utilizing brute force for confrontation training relies too much on the attack method and cannot traverse all confrontation samples. Hence, unknown adversarial examples can still easily deceive the defense model. The defense range of this method is small, the training cost is high, and adapting to the network environment is hard.
- Defense Algorithms Based on Network Structure: This type of method refers to enhancing the robustness of the network by modifying the model's architecture to achieve the goal of defending against adversarial example attacks. Ross and Doshi-Velez [6] bring up a network according to the degree of change in the output caused by the penalty of input changes, so that small perturbations have no effect on the results. In addition to regularization methods, Papernot et al. [7] came up with an algorithm known as defensive distillation. This scheme uses soft labels to train the distillation model, which can help the probability vector discover additional knowledge. This prevents overfitting and helps generalize better to the training points. The experimental results of this method show that defense distillation can significantly improve the model's resilience to small perturbations. Ge et al. [8] proposed a dual-stream architecture to protect Deep Neural Networks (DNN) from adversarial examples by understanding the effect of perturbations on the models of high-resolution and low-resolution. They established a model with 10,000 user nodes and 1 server node to imitate the true environment for experiments. This method does not require adversarial examples and can detect adversarial examples. This type of defense method is easy to implement and has strong generalization ability and transferability. But it is easy to cause overfitting, and the defense effect depends on the size of the disturbance added during the attack. This method has a large overhead, and the model needs to be reconstructed and retrained, resulting in high training costs.
- Defense Algorithms Based on Additional Networks: This type of method refers to mitigating adversarial attacks by adding one or more additional networks to the original neural network. Meng and Chen [9] present the Magnet network, which consists of a detector and reformer. First, the detector is used to judge whether the target sample is kept away from the manifold boundary. If the target sample is far from the manifold boundary, it is deleted immediately. Otherwise, pass it to the reformer to move it towards the manifold of original samples to recover the malignant sample. The algorithm does not need to understand the generation process of adversarial examples and

In order to reduce the training cost while improving the accuracy of example classification, we propose a novel defense algorithm for adversarial examples combining micro-network.

Compared with other defense algorithms, defense algorithms based on GAN will "recover" the adversarial example into reconstructed samples that are almost consistent with the distribution of the original samples.

have strong generalization ability. However, adversarial examples with large perturbations can only be discarded. Although security is guaranteed, it will inevitably cause a waste of data resources. Besides, using a denoising network for defense is also one of the most commonly used methods in this category. Akhtar et al. [10] propose a defense framework for denoising networks. The framework adds extra layers to the model. In this framework, a method to calculate the adversarial perturbation is designed to train the pre-input layer to correct the perturbed image. This scheme does not require any perturbation to the target network and has strong generalization ability, but the training overhead is high.

The current adversarial example defense algorithm based on GANs prevents attackers from attacking by mapping the malignant sample to the distribution of the original sample. Samangouei et al. [15] proposed Defense-GAN, a universal defense algorithm effective for both black-box and white-box attacks. This algorithm does not amend the classifier structure or training process. By minimizing the reconstruction error, the input image is "projected" into the scope of the Generator, which greatly reduces the efficiency of the adversarial example attack. Jin et al. [11] also proposed APE-GAN. This method uses DNN to build a Generator and Discriminator to get rid of the limitations of traditional GAN gradient disappearance. Experiments show that the error rate of adversarial example input is significantly reduced, but the defense accuracy is not high enough. Moreover, the generated reconstructed samples and real samples have slight errors, and the training cost is expensive.

Although GAN-based adversarial defense has a higher detection rate and is simple to implement, it relies heavily on the performance of GAN. If the GAN is not properly trained, it will affect the identification precision of examples, thereby affecting the defense effect. Moreover, the training cost of this method is relatively high, requiring a large amount of data and computing resources. To this end, we propose an improved model for defense with GAN. This design of the model uses a micro-network structure, which greatly reduces the training cost while ensuring the defense effect. At the same time, we also introduce model compression into the classification model and construct a new Generator, hence, the reconstructed sample is closer to the real example distribution.

Our Algorithm

In our previous work [12], [13], we employed public datasets such as MNIST and CIFAR-10 to evaluate the classification model based on the micro-network structure and to implement adversarial example defense. In our algorithm, the model will be tested using collected data, including speed limit signs, traffic light countdowns, lane markings, etc. Prior to testing, the collected data will be processed so that each datum is of size 32 × 32 pixels. The architecture of the model will be suitably modified, followed by the implementation of adversarial defenses against such datasets.

Based on the micro-network structure, we propose an algorithm based on GAN to prevent traffic signs from being attacked by adversarial examples, which is called *MN-GAN*. As shown in Figure 1, our defense algorithm consists mostly of three steps: building the dataset, training the GAN, and implementing defense. It primarily includes three network structures, namely the classification network, the Generator and the Discriminator. The network structure diagram is shown in Figure 2. The processing flow of *MN-GAN* is to first use the Fast Gradient Sign Method (FGSM) to attack the model to generate an adversarial example set corresponding to the original sample set. Then train the GAN using a dataset containing malignant samples and original samples. Finally, the adversarial examples are input into the trained Generator to generate reconstructed samples. Then, the reconstructed samples are sent into the classification network for classification. This

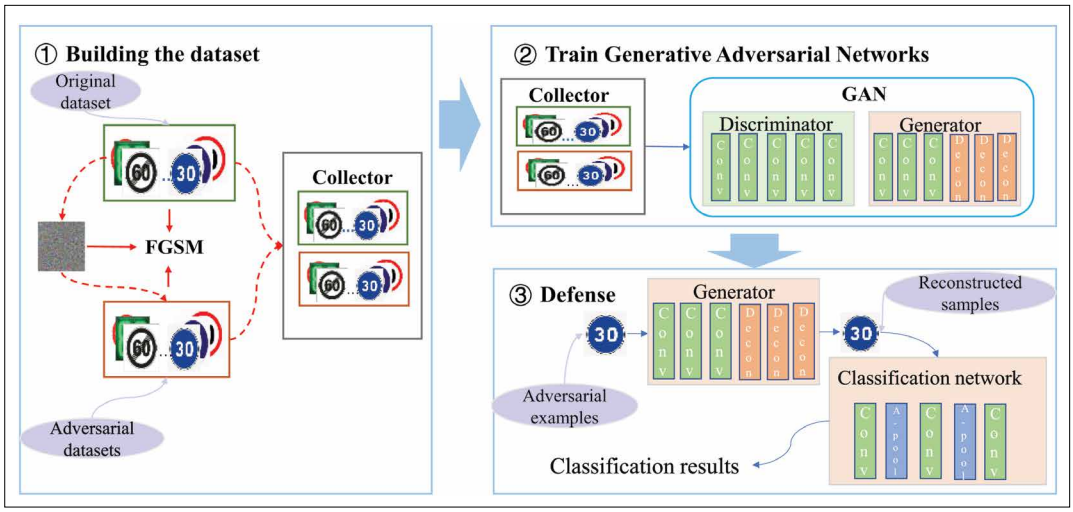

FIGURE 1. The processing flow of *MN-GAN*.

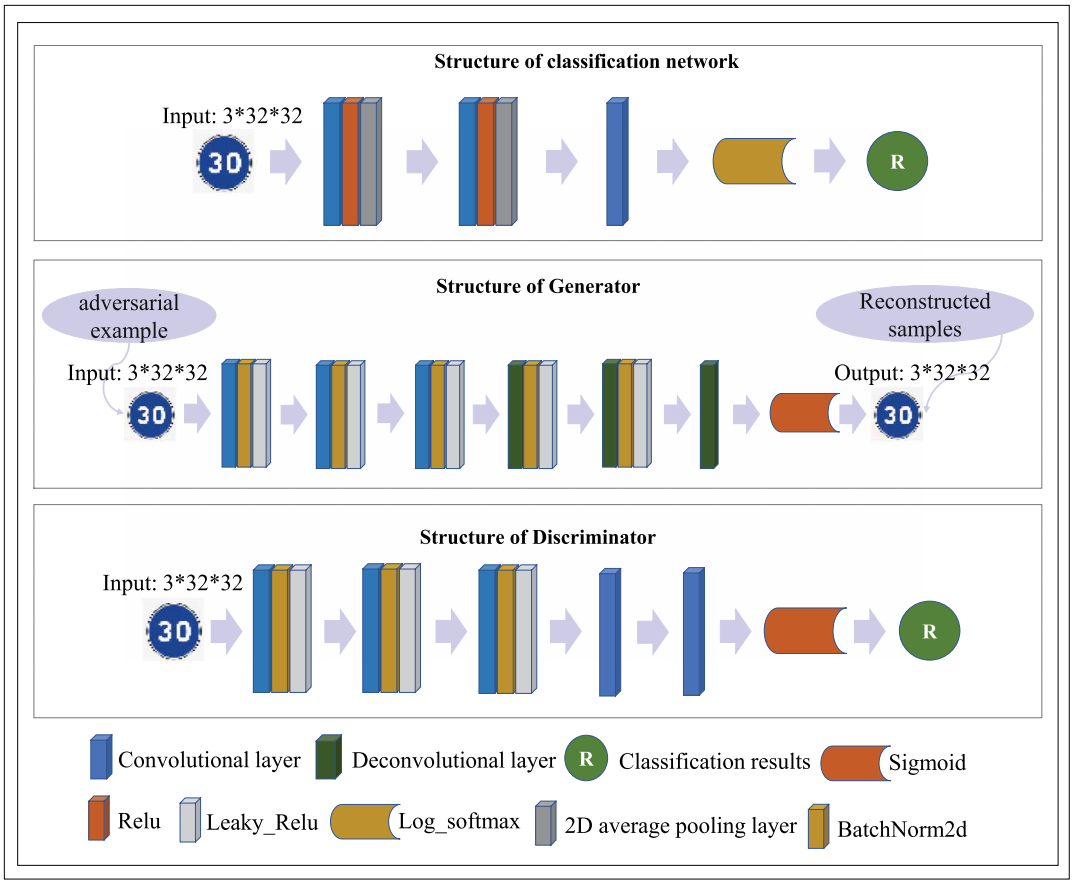

FIGURE 2. Three network architectures used in *MN-GAN*.

section will focus on the network module and loss function of the defense algorithm.

Classification Network

In the defense algorithm, the computational load and parameter usage of the classification during the training process are quite huge.

Common classification networks generally consist of convolutional layers and fully connected layers. The former extracts feature information, and the latter is used as classification. But this traditional classification network has certain limitations. Firstly, a fixed input image size is necessary, which requires operations such as cutting the image, and the image information will be lost virtually. Secondly, the model parameters of this traditional classification network are basically concentrated in the latter, and the prediction time is mainly occupied by the fully connected layer. It has an impact on real-time requirements and takes up a lot of memory. We use a micro-network structure to compress traditional classification networks. Eliminated the fully connected layer in the classification model to overcome the limitations of the classification network. It is composed of convolutional layers and 2D average pooling layers.

Generator

The role of the Generator is to try its best to generate "false samples" that closely match the distribution of the original samples. To ensure the effect of the "false examples" generated by the Generator, its model is composed of both convolutional layers and deconvolutional layers. The Generator structure in *MN-GAN* includes three convolutional layers and three deconvolutional layers. Experiments have shown that when the stride of the Generator's convolutional and deconvolutional layers is set to 2, the reconstructed samples produce the best results. Therefore, the Generator we propose first uses convolutional layers with a stride of 2 to obtain a lower-resolution feature map, and then uses deconvolutional layers with a stride of 2 to restore the original resolution. In 2011, Glorot proposed that the sigmoid activation function is closer to the biological neuron model [14]. The Generator used in our defense algorithm is ultimately trained using the sigmoid activation function. The final trained Generator produces a "false example" that is nearly identical to the structure of the original sample.

Discriminator

We proposed the Discriminator, which also uses the micro-network structure for model compression. As early as 2014, Lin et al. [1] proposed micro-network-enhanced local modeling based on the limitations of traditional convolutional layers using linear filters and nonlinear activation functions to scan inputs that are prone to overfitting. This method employs convolutional layers and two-dimensional average pooling instead of fully connected layers in a classification network. Doing so is less prone to overfitting. Therefore, the second half of our proposed Discriminator uses 1 × 1 convolutional layers instead of fully connected layers to compress the model. The

The MN-GAN algorithm uses a micro-network structure to design a classifier and Discriminator. And we design a Generator model with a stride of 2 in the algorithm, which makes the sample classification accuracy rate reach 97.9%.

network first utilizes four convolutional layers to output a 64 × 1 × 1 three-dimensional shape, then uses a 1 × 1 convolutional layer instead of a fully connected layer for model compression. The Discriminator we designed performs game training with the Generator.

Loss Function

For training the GAN, we utilized a dataset comprising both original and malignant examples. The Generator vigorously generates "false samples" to mimic the original structure, while the Discriminator will try its best to differentiate the original samples from the reconstructed samples. Therefore, GAN training involves the use of two loss functions.

The purpose of Generator is to ensure that samples generated by it are as authentic as possible. Therefore, the loss function of Generator is a weighted sum consisting of pixel-wise mean square error and antagonistic error.

Among them, the former mainly considers the image content loss, aiming to make the reconstructed samples constantly close to the original samples. The goal of adversarial error loss is to generate reconstructed images that closely match the distribution of the original samples. This is calculated based on the Discriminator's recognition probability of the reconstructed samples, and the prediction results of the reconstructed samples keep approaching the labels of the original samples. Therefore, the loss function of the Generator in this defense algorithm is:

$$l_{\text{mn}-g} = \varepsilon_1 l_{\text{mse}} + \varepsilon_2 l_{\text{adv}} \quad (1)$$

Here, ε_1 and ε_2 denote the weights of the mean squared error loss and adversarial error loss. l_{mse} represents the pixel-wise mean squared error loss, and l_{adv} represents the adversarial error.

Using the Discriminator, we can distinguish the real samples from the reconstructed samples generated by the Generator. Consequently, the Discriminator is trained by setting the label of the original sample to 1 and the label of the reconstructed sample to 0. The loss function is defined as follows: where X and X^{adv} represent the original and adversarial examples severally.

$$l_{\text{mn}-d} = -\sum_{n=1}^{N}\left[logD_{\theta_D}(X) + logD_{\theta_D}\left(X^{\text{adv}}\right)\right] \quad (2)$$

Experiment

This chapter will illustrate the effectiveness, feasibility, and innovation of our proposed defense algorithm. Firstly, the dataset used in the experiment, experimental settings, and evaluation indicators are introduced. Subsequently, various evaluation metrics of the proposed algorithm are compared with those of other defense algorithms.

Dataset

The defense algorithm we propose will use two datasets: the original sample dataset and the adversarial example dataset. The original sample dataset comprises 10,000 color traffic signs collected after processing, and the pixels of the processed traffic signs are all set to 32 × 32. The dataset includes, but is not limited to, speed limit signs, lane signs, turning intersections and turning angle signs, intersection number signs, traffic light countdown, etc. For the experiment, the FGSM is employed to generate an adversarial example dataset. The image number, size, pixel, and channel of the adversarial example data set and the original sample are the same.

Experimental Setup

There are three primary sections to the experiment. First, the classification network was trained for a total of 100 epochs. The initial learning rate of the model is set to 0.01 and trained using the *SGD* optimizer. We also set the learning rate adjustment factor to 0.1, and the cross-entropy loss function is employed for gradient descent. Secondly, an attack model is trained to generate an adversarial example dataset. Subsequent experiments reveal that the attack effect is best when the disturbance value is set to 0.30 in the FGSM algorithm. Third, training the GAN enables the Generator to generate reconstructed samples that closely match the distribution of the original samples. The essence of training GAN is to simultaneously perform game training for the Generator and the Discriminator in each cycle. Experiments have demonstrated that the training of GAN is performed in 2 cycles, and the effect of performing 4 cycles on the Generator in each cycle is the best. Set the learning rate to 0.0002, and use $l_{\text{mn}-g}$ and $l_{\text{mn}-d}$ defined above as two loss functions of GAN. The weights for pixel-level mean square error loss and adversarial loss are set to 0.7 and 0.3, respectively. The Adam optimizer is employed.

Evaluation Index

In the experiment, we will employ computation amount, parameter amount, structural similarity, and the accuracy of reconstructed sample prediction as evaluation indicators.

1. Amount of Computation

The amount of computation corresponds to the time complexity, which primarily depends on the execution time of the training network. Floating-point operations (FLOPs) are commonly used as a unit to measure the amount of computation, representing the number of floating-point operations. GFLOPs are also frequently employed in papers when the computation is particularly heavy. GFLOPs denote one billion floating-point operations per second; thus, 1 GFLOP is equal to 10^9 FLOPs.

2. Number of Parameters

The number of parameters corresponds to the space complexity, which mainly indicates the amount of memory occupied by the network during training. It is typically assumed that a parameter is a float, which occupies 4 bytes. Therefore, bytes are generally used as the unit when evaluating parameter quantities. When

dealing with a large number of parameters during model training, KB, MB, and GB are commonly used as units of measurement.

3. Structural Similarity

In addition to using classifier detection and naked-eye observation to assess the effectiveness of reconstructed samples, the similarity between them and the original image will be measured using the Structural Similarity Index (SSIM).

SSIM measures the similarity between two images based on brightness, contrast, and structure. Specifically, mean, standard deviation, and covariance are used as metrics of brightness, contrast, and structural similarity, respectively. SSIM values range from 0 to 1, with a higher value indicating greater similarity. When two images are identical, the SSIM value is 1.

4. Classification Accuracy

We input the adversarial example set into the trained Generator to generate the reconstructed sample set. And then input this into the classification network to obtain the prediction accuracy.

Algorithm Analysis

In our experiment, the initial step involves determining the perturbation value that yields the best attack effect for the attack algorithm. Following that, we set the number of convolutional layers and the step size in the Generator. Finally, we utilize the verified parameters as the values of our experiments and proceed with the experiments. We then compare the experimental effects of *MN-GAN* with commonly used defense methods.

In our experiment, we mainly compare the defense accuracy of each defense method, the similarity between the reconstructed sample and the original sample, and the training cost, which includes the number of parameters and computations used by a single sample in the defense strategy. This section is divided into four main parts: parameter setting, reconstructed sample quality, defense accuracy, and training cost.

Parameter Setting

The initial stage of training is to prepare the dataset. In addition to using the original dataset, it is also necessary to generate an adversarial example set. In our experiment, the FGSM algorithm combined with the classification network is employed to produce an adversarial example set. The perturbation value of the attack algorithm is set to 0.10, 0.15, 0.20, 0.25, 0.30, and 0.35, respectively. An attack test is conducted on 10,000 data points in the original dataset. Figure 3 shows the classification accuracy, precision, recall, specificity, and balanced F-score of the input samples for each perturbation value. The experimental data shows that the FGSM algorithm has the best attack effect on the original dataset when the perturbation value is 0.30.

Convolution layer parameters such as layer number and step size should be considered when designing the Generator. In our experiments, we designed the model to employ 2-layer, 3-layer, and 4-layer convolutional layers and deconvolutional layers. Figure 4 shows the comparison of each index using different generators to generate reconstructed samples. It can be intuitively concluded that the image generated after training the Generator with 3 layers of convolutional layers and 3 layers of deconvolutional layers yields the best results. Compared to the Generator with 2 layers, the image similarity is improved by 5.63%, with a slightly higher training cost than the 2-layer network. However, the sample quality generated

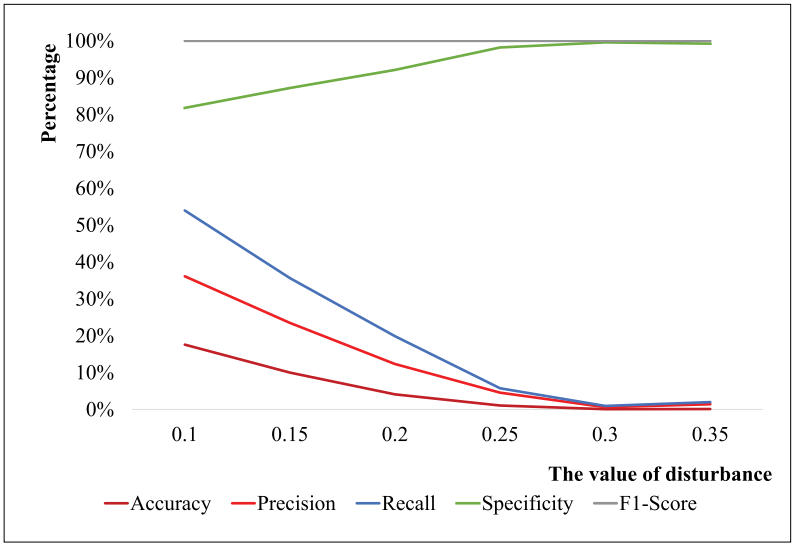

FIGURE 3. Comparison of various indicators for adversarial example classification.

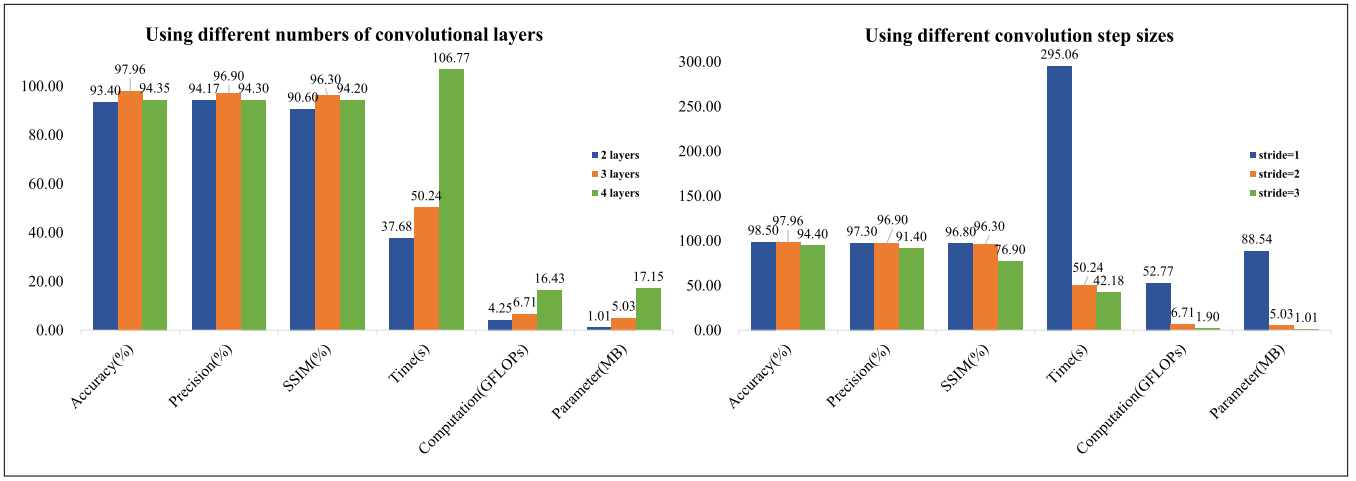

FIGURE 4. Comparison of metrics for reconstructed samples generated by different generator networks.

by a 4-layer Generator is inferior to that of a 3-layer network.

Therefore, based on the comparison of the above experimental data, the *MN-GAN* we proposed uses a 3-layer convolutional layer and a 3-layer deconvolutional layer to form a Generator. Subsequently, we conducted experiments by setting the step size parameters to 1, 2, and 3, respectively. It is observed that the accuracy, precision, and SSIM value of reconstructed samples are not much different when the step size is 1 and 2. But when the step size is set to 1, the training time is nearly 6 times that of the step size 2 training time. Although there is little difference in training time and cost between stride 2 and 3, the image quality generated with a stride of 2 is significantly better than that with a stride of 3. Considering these factors comprehensively, we chose a step size of 2 for both the convolutional layer and the deconvolutional layer in the Generator in our experiment.

Reconstructed Samples

The examples produced by the trained Generator are reconstructed samples. The comparison of SSIM values between *MN-GAN* and other commonly used defense methods is shown in Figure 5. The SSIM value between the reconstructed samples generated by *MN-GAN* and the real samples reaches 0.9626, which is closer to the real sample distribution than other defense methods based on GAN. The SSIM value of Magnet is slightly higher than that of our algorithm. However, the later training cost comparison data of various defense methods shown shows that the computation amount used in Magnet training is about 7 times that of our defense algorithm, and the number of parameters is about 16 times that of our algorithm.

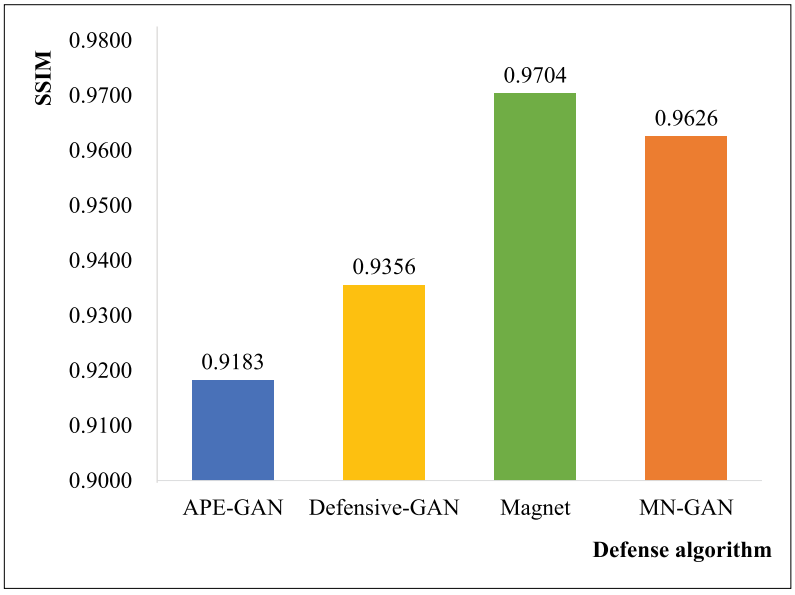

FIGURE 5. SSIM value comparison between *MN-GAN* and other defense algorithms.

Defense Effect

Our proposed defense algorithm is equivalent to denoising malignant samples. Figure 6 compares the recognition accuracy of original samples and reconstructed samples of several defense methods. The classification accuracy of our algorithm on the original samples reaches 99.59%, which is the best among all methods for the classification of original samples. The recognition accuracy of reconstructed samples has increased to 97.96%, which is only slightly inferior to Magnet's recognition accuracy of reconstructed samples among several defense methods.

The defense algorithm we proposed not only improves the quality of reconstructed samples and the recognition accuracy of reconstructed samples, but also enormously cuts down on the training cost. In the experiment, the amount of computation and parameters are used as indicators for comparison. In the defense model we proposed, the computation amount for a single sample entering the classification network training cycle is about 0.03 GFLOPs, and the parameter size is about 20.08 KBytes. The computation amount of entering the Discriminator training once is about 0.51 GFLOPs, and the parameter amount is about 1.66 MBytes. Our proposed defense algorithm based on the micro-network structure has the lowest number of parameters, and the computation amount is only slightly inferior to the defense distillation. Compared with Defense-GAN and Magnet, the amount of computation and parameters are reduced by about 15 times, respectively. Generally speaking, the training cost of the defense algorithm we propose is much lower than that of other defense methods, and the requirements for CPU and video memory are lower.

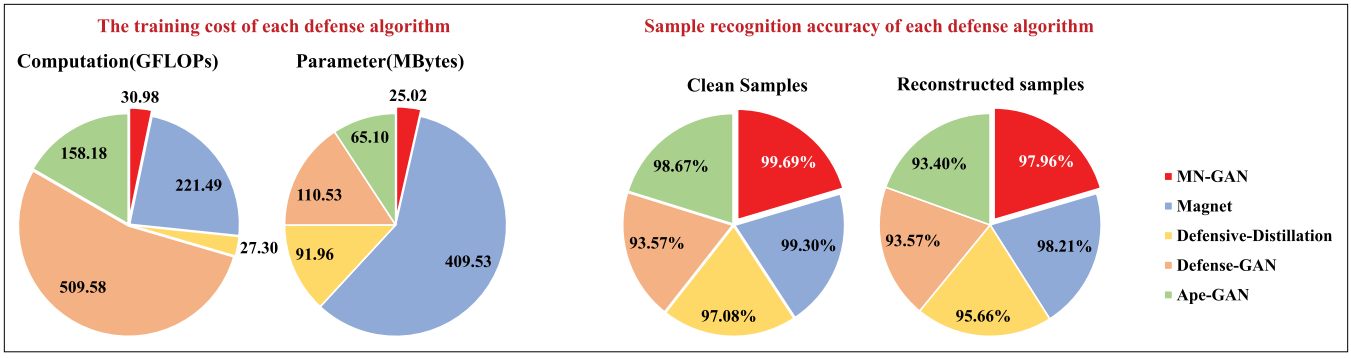

FIGURE 6. Comparison of defense effects of each algorithm.

Conclusion and Future Work

We propose a new defense algorithm (*MN-GAN*) based on micro-network structure and GAN to defend against traffic sign adversarial examples in intelligent driving. By training GAN, the Generator can map the manifold distribution of adversarial examples to the real samples to eliminate the adversarial disturbance so as to reach the aim of adversarial defense. Both the classifier and the Discriminator of the defense algorithm use the micro-network structure for model compression, and a model combining a convolutional layer with a step size and a deconvolutional layer is designed for the Generator in GAN. For the sake of verifying the experimental effect, experiments were carried out in combination with the FGSM attack algorithm in the collected data sets of various traffic signs and compared with several other defense methods. Experiments show that our defense algorithm has the best "recovery" effect on traffic sign adversarial examples and has high accuracy in its classification. Although compared with Magnet, our algorithm is slightly inferior in accuracy, but a lot of computation and parameters used in training are greatly reduced, and the response speed is improved. Our experiments focus on defending against FGSM adversarial instances of traffic signs, and the generalization is low. Therefore, future work should focus on designing and researching a traffic sign adversarial example defense framework that can continuously improve defense accuracy and defense model generalization while reducing training costs. Readers may consider designing compressed models for secondary defense and employing data augmentation techniques to enhance defense accuracy and generalization. This will have an important meaning for improving the safety of intelligent driving.

Acknowledgment

This work was supported in part the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province under Grant SKLACSS-202110; in part by the National Natural Science Foundation of China under Grant 62102049 and Grant 62102379; in part by the National Key Research and Development Program of China under Grant 2017YFB0802300; in part by the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0557; in part by the National Key Research and Development Plan of China, Key Project of Cyberspace Security Governance under Grant 2022YFB3103103; in part by the Key Research and Development Project of Sichuan Province under Grant 2022YFS0571, Grant 2021YFSY0012, Grant 2021YFG0332, and Grant 2020YFG0307.

References

[1] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Montreal, QC, Canada, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.

[3] A. Shafahi et al., "Adversarial training for free!" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[4] Y. Wang et al., "Improving adversarial robustness requires revisiting misclassified examples," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[5] C. Mao et al., "Metric learning for adversarial robustness," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[6] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 1660–1669.

[7] N. Papernot et al., "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[8] H. Ge et al., "Two-stream architecture as a defense against adversarial example," *J. Electron. Sci. Technol.*, vol. 20, no. 1, pp. 81–89, 2022.

[9] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," presented at the ACM SIGSAC Conf. Comput. Commun. Secur., Dallas, TX, USA, Oct. 2017.

[10] N. Akhtar, J. Liu, and A. Mian, "Defense against universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3389–3398.

[11] G. Jin et al., "APE-GAN: Adversarial perturbation elimination with GAN," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3842–3846.

[12] X. Liu et al., "TLTD: A testing framework for learning-based IoT traffic detection systems," *Sensors*, vol. 18, no. 8, p. 2630, Aug. 2018.

[13] X. Liu et al., "Weighted-sampling audio adversarial example attack," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 4908–4915, Apr. 2020.

[14] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[15] P. Samangouei, M. Najibi, and L. S. Davis, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition (CVPR)*, May 2018, pp. 2692–2700.

Biographies

Jiazhong Lu (ljz@cuit.edu.cn) currently works at the Sichuan Provincial Key Laboratory of Advanced Cryptography and System Security, School of Network Security, Chengdu University of Information Technology, as an Associate Professor and a Master's Tutor. His research fields are network security and machine learning.

Chenli Wang (3210810011@stu.cuit.edu.cn) is currently pursuing the master's degree with the School of Cyberspace Security, Chengdu University of Information Technology. Her research fields are adversarial sample defense and network security.

Yuanyuan Huang (hy@cuit.edu.cn) received the B.Sc., M.Sc., and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2004, 2007, and 2013, respectively. He was a Visiting Scholar with the University of Washington, Seattle, USA, from 2009 to 2011. He had been a Post-Doctoral Researcher with the University of Electronic Science and Technology of China. He is currently an Associate Professor with the Chengdu University of Information Technology, Chengdu. His main research interests include image/video processing, big data, and artificial intelligence.

Kangyi Ding (kangyiding@gmail.com) received the B.S., M.S., and Ph.D. degrees from the University of Electronic and Technology of China (UESTC). He is an Assistant Research Fellow with the Institute of Computer Application, China Academy of Engineering Physics. His research interests include AI security, adversarial sample, and adversarial transferability.

Xiaolei Liu (luxaole@gmail.com) received the M.S. and Ph.D. degrees in software engineering from the University of Electronic and Technology of China (UESTC). He is an Associate Research Fellow with the Institute of Computer Application, China Academy of Engineering Physics. His research interests include cyber security, AI security, and explainable AI.