Research on Adversarial Sample Detection Method Based on Image Similarity

Xiaoxue Wu[1,2], Shuqi Zuo[1], Shiyu Weng[1], Yongkang Jiang[1], Hao Huang[3]*

[1] School of Information Engineering, Yangzhou University, China
[2] Key Laboratory of Safety-Critical Software (Nanjing University of Aeronautics and Astronautics),
Ministry of Industry and Information Technology, China
[3] School of Software, Northwestern Polytechnical University, China
wuxiaoxue00@gmail.com, 1652299374@qq.com, shiyuweng0226@163.com, 1022609482@qq.com, 463248663@qq.com

Abstract

With the widespread application of deep neural networks in image detection, adversarial sample attacks have gradually become a hot issue of concern for researchers. In this paper we propose a new adversarial sample detection approach called **AdvDetector**, which combines image generation through label fusion with image similarity detection. AdvDetector enhances sample quality and effectively identifies adversarial samples. Specifically, the method generates images by selecting seed pixels, the labels of deep neural network classification, and the pixel distribution learned from training data, and detects them using image similarity comparison methods. During the sample generation process, we introduce the AdvDetector method for adversarial sample detection to improve the quality of generated samples. We evaluated the effectiveness of the method on three publicly available image datasets, MNIST, Cifar-10, and GTSR, and the results show that the method is superior to existing baseline methods in terms of adversarial sample detection rate and sample generation quality.

Keywords: Adversarial samples, Deep neural networks, Image generation, Similarity detection, Label fusion

1 Introduction

With the widespread application of deep learning models in various fields, the security issues of these models are receiving increasing attention. One of the most important security issues is the vulnerability of adversarial samples [1-4], which are carefully crafted by adding carefully designed perturbations to legitimate inputs to deceive the model. Adversarial samples [5] not only reduce the accuracy of the model, but also pose potential threats to safety critical applications such as autonomous driving [6-8], facial recognition [9], image recognition [10], speech and text processing [11-14], and medical diagnosis.

To address this problem, various adversarial defense methods have been proposed. In this paper, we propose a new method for detecting adversarial samples called **AdvDetector**. This method is based on the FGSM adversarial sample generation method and uses a complementary network to detect adversarial samples by calculating the similarity between samples. Compared to other methods, AdvDetector has higher accuracy and lower false positive rates, and can detect more adversarial samples. We conducted comprehensive experimental validation on multiple datasets, including ImageNet, CIFAR-10, and CIFAR-100. Our method is based on image similarity and gradient background information, and detects adversarial perturbations by dividing the image into small blocks. Our results demonstrate that AdvDetector performs well in detecting adversarial attacks and has higher robustness and accuracy than existing methods. Furthermore, our method can help improve the robustness and accuracy of deep learning models by better understanding adversarial perturbations.

We organize the rest of this paper as follows. In Section 2, we provide a brief overview of background on adversarial sample detection methods. Section 3 introduces the AdvDetector method in detail. In Section 4, we describe our experimental settings and present the results of our experiments. Finally, we conclude the paper in Section 5 with a summary of our findings and suggestions for future work.

The contributions of our work are as follows:

- We propose a new method for detecting adversarial samples called AdvDetector, which uses image similarity and gradient background information to detect adversarial perturbations.
- We conduct comprehensive experimental validation on multiple benchmark datasets to demonstrate that AdvDetector performs well in detecting adversarial attacks.
- We compare the performance of AdvDetector with existing methods and show that it has higher robustness and accuracy.
- We improve the robustness and accuracy of deep learning models by analyzing the impact of adversarial perturbations.

2 Background

The complexity of constructing deep neural networks and the issues caused by gradient descent, like misclassification,

*Corresponding Author: Shiyu Weng; E-mail: shiyuweng0226@163.com

DOI: 10.53106/160792642024012501012

have led to a significant focus on high-quality adversarial sample detection methods in academic research. Adversarial examples refer to subtle, imperceptible perturbations that can lead to target model classification errors. Input samples with such perturbations are termed adversarial samples, and the process of crafting them to deceive neural networks is called adversarial attacks. The concept of adversarial samples was introduced by Szegedy et al. in 2013 [5].

To counter adversarial threats, research has explored methods to improve model accuracy in classifying or detecting adversarial samples. Adversarial training, which involves adding adversarial samples during training, has been effective. For instance, GoodFellow proposed using the FGSM method for adversarial sample generation [15]. However, the effectiveness of such methods depends on the quality and quantity of adversarial samples used in training.

Li et al. [8] introduced a method for adversarial sample detection by compressing samples through pixel reduction and blurring. While this approach simplifies detection, it focuses solely on data-related issues, neglecting the inherent characteristics of deep neural networks. This can limit its ability to enhance network robustness.

Detecting misclassified samples, especially for unlabeled data, is a challenging task. Manual inspection of large datasets is inefficient and lacks long-term accuracy. Therefore, this article proposes a novel approach: amplifying features in adversarial samples that affect deep neural network classification, using label-related features to generate potential samples, and then comparing them to the originals to detect adversarial samples. Figure 1 shows the framework of AdvDetector.

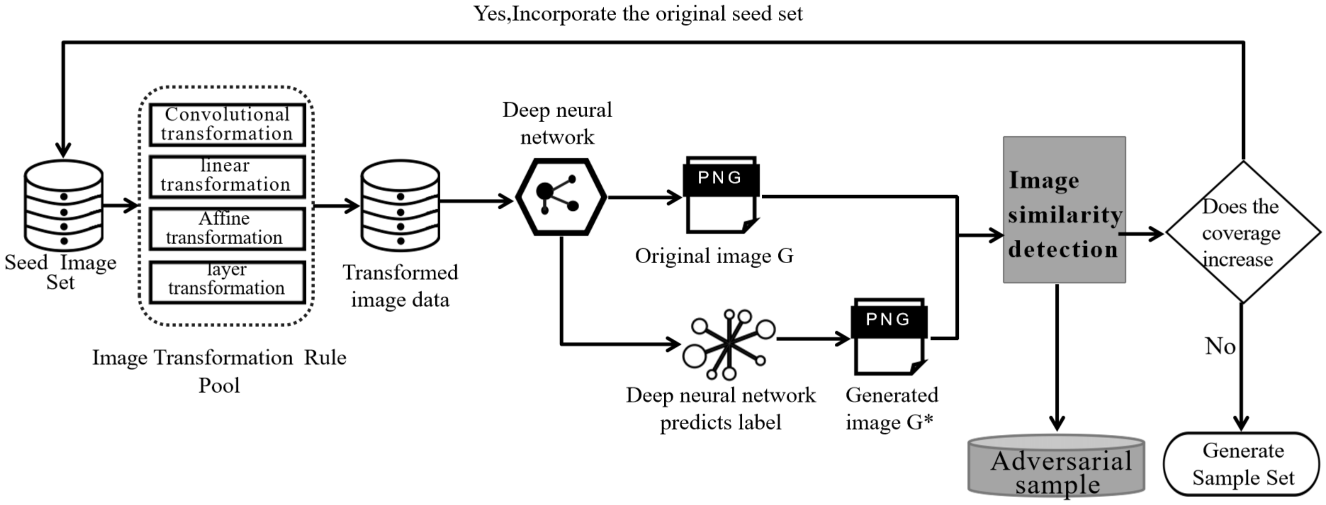

Figure 1. The framework of AdvDetector

3 AdvDetector

3.1 Image Generation with Tag Fusion

Existing methods like DeepFool [16] and GAN [17] networks generate images based on sample features. This paper proposes a new method incorporating pixel continuity into label fusion to improve image generation quality and reliability. The method trains the likelihood distribution of pixels and improves joint distribution using the guiding effect of labels.

3.2 Image Similarity-Based Adversarial Sample Detection

We propose an adversarial sample detection method based on image similarity comparison. Traditional image similarity comparison methods, including Euclidean distance, cosine similarity, and Kullback-Leibler divergence (KLD) [18-19], are introduced. We also present the PxielDefend (PD) method for detecting adversarial samples.

3.3 Sample Generation Based on Adversarial

Sample Detection

1) Image Transformation Rule Pool We construct an image rule mutation pool, including affine transformations, layer-based shape mutation, and image filters. We aim to establish a transferable adversarial sample rule library to simulate real camera angles or object motion and test model robustness.

2) Coverage-Guided Sample Generation Using neuron coverage in deep neural networks, we guide image generation by determining whether the image coverage rate increases. We generate three types of image samples: benign, adversarial, and incorrect samples. For benign and incorrect samples, we decide whether to add the sample to the original dataset based on the coverage rate increase. For adversarial samples, we use the GGBLF adversarial sample detection method to detect and record misclassification rules.

4 Experiment

4.1 Adversarial Sample Detection Datasets

The purpose of facilitating experimental comparison with other benchmark methods, this study employs widely used image datasets, including MNIST [20], CIFAR-10 [21], and the German Traffic Sign Recognition Benchmark (GTSRB

[22]), to evaluate the performance of adversarial sample detection methods. These datasets have been widely adopted in the field of adversarial sample detection. Table 1 presents relevant information for these datasets.

Table 1. Information on adversarial sample detection datasets

Dataset	Number of classes	Training set size	Test set size	Total
MNIST	10	50,000	10,000	60,000
CIFAR-10	10	50,000	10,000	60,000
GTSRB	43	39,209	12,630	51,839
Autonomous driving dataset	42	52,000	11,000	63,000

4.2 Evaluation Metrics

Adversarial Sample Detection Evaluation Metrics

In the adversarial sample detection methods, this experiment uses the same detection methods as existing research to ensure the reliability of the experimental results.

In the adversarial sample detection method proposed in this paper, since the situation where the classifier can correctly detect adversarial samples is not within the scope of adversarial sample detection, there are only two detection situations in this experiment:

- True Negative (TN) and True Positive (TP): The classification result is y or y’ , and the actual detection is a benign sample.
- False Negative (FN) and False Positive (FP): The classification result is y or y’ , and the actual detection is an adversarial sample.
- Adversarial Sample Detection Accuracy (ADR), i.e., the proportion of adversarial samples detected by the method in the existing dataset to the total number of adversarial samples:

$$ADR = TPR = \frac{TP}{TP + FN} = \frac{n}{N}. \quad (1)$$

In this passage, n represents the number of adversarial samples actually detected, while N denotes the total number of adversarial samples in the dataset. ADR stands for the probability of detecting adversarial samples. The False Positive Rate (FPR) of adversarial sample detection refers to the proportion of benign samples that are mistakenly identified as adversarial samples in the given dataset:

$$FPR = \frac{FP}{FP + TN} = \frac{ps}{ns}. \quad (2)$$

In this context, ps represents the number of benign samples that are misidentified as adversarial samples by the method, while ns denotes the total number of benign samples in the dataset. FPR, or False Positive Rate, refers to the rate of false positives in sample detection during the experiment.

For the rule-guided image generation experiment, this paper evaluates the proposed method based on the neural coverage (NC) of deep neural networks and the accuracy of detecting adversarial samples. Here, we focus on three criteria: neural coverage, the number of generated samples, and the accuracy of detecting adversarial samples.

Inject test cases into the neural network, and during each training or testing process, each neuron in each layer of the neural network has an output value. If this output value is greater than a certain threshold, it indicates that the neuron is activated. [23] The ratio of the number of activated neurons to the total number of neurons is defined as neuron coverage. The neuron coverage can be defined as follows:

$$NCov(T, x) = \frac{\{n \mid \exists \in T : \phi(x, n) > t\}}{|N|}. \quad (3)$$

The deep neural network coverage obtained under different sample quantities can be compared by comparing the coverage of different sample quantities:

$$CovG = \frac{Cov}{N}. \quad (4)$$

In the equation, N represents the number of generated samples, Cov represents the coverage of the deep neural network for N generated samples, and CovG represents the coverage of the deep neural network for a single sample.

Samples generated during the process may cause deep neural networks to fail to classify normally due to their inherent uncertainty. The DeepHunter method does not detect adversarial samples, while the DeepSmartFuzz method restricts the labels of generated samples, and labels that are different from the original samples are considered adversarial samples. This paper will detect the number of adversarial samples using an adversarial sample detection method, and the accuracy of the adversarial sample detection will be measured by comparing the number of detected adversarial samples with the total number of generated samples:

$$AdvRate = \frac{adv_n um}{N}. \quad (5)$$

4.3 Benchmark Methods

To evaluate the effectiveness of the proposed AdvDetector method, we selected relevant baseline methods for comparison, including the I-Defender method by Zheng [24] et al. the PixelDefend method by Song [25] et al. These methods use similar datasets and are briefly introduced below:

- I-Defender method:An unsupervised approach for detecting adversarial inputs, it builds a hidden state distribution for natural data, excelling in black-box and gray-box attacks without the need for specific attack method attention or adversarial sample training.

- PixelDefend method: Purifying tampered images by repositioning them within the training data distribution, it enables correct classifier operation without requiring knowledge of the adversarial attack method. This forward approach allows easy reuse across various models and integration with other defense methods, mitigating the impact of adversarial attacks on deep neural networks.

After validating the proposed adversarial sample detection method, this paper introduces a data augmentation method based on coverage-guided mutation pool to generate samples. To evaluate the effectiveness of this method, comparisons are made with DeepHunter, a coverage-guided fuzzing framework proposed by Xie et al. [26], and DeepSmartFuzz, a test case generation method proposed by Demir et al. [27].

- DeepHunter:A coverage-guided fuzzing framework, introducing image mutation techniques for efficient image augmentation. It prioritizes seeds, ensuring testing efficiency, and employs testing and assertion on mutated images to identify error triggers.
- DeepSmartFuzz: This approach utilizes Monte Carlo Tree Search (MCTS) [28] and coverage guidance for test case generation. It employs MCTS to select image mutation rules, guide the mutation process, and limit the maximum mutation distance to prevent deviations from benign samples. The generated samples are then compared with testing predictions to assess their utility.

4.4 Experiment Settings

Experiment 1: Adversarial Sample Detection Comparative Experiment

To verify the accuracy of the Adversarial Sample Detection (ASD) part of our proposed method, we replaced 10% of the original dataset with adversarial samples that were misclassified by the target deep neural network. We compared our method with existing ones by identifying the number of detected adversarial samples and the false positive rate (FPR) of benign samples. The experiment process is shown in Figure 2.

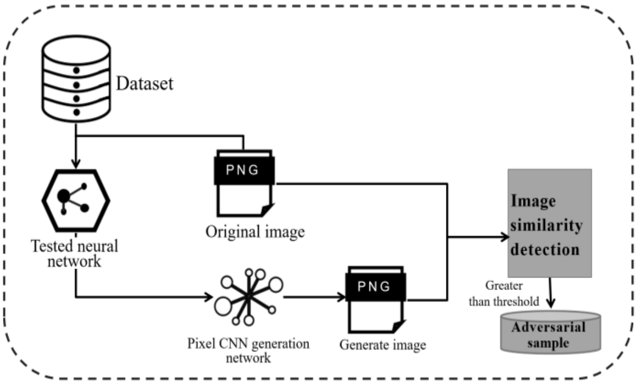

Figure 2. Adversarial sample detection experiment flowchart

Specifically, the experiment was conducted in two ways:

- Replacing 10% of the dataset with adversarial samples, we evaluate detection using metrics like Adversarial Detection Rate (ADR) and False Positive Rate (FPR) for comparison.
- Assessing defense against various attacks, we added diverse attack types to the neural network, generating 1000 samples each. We manually assessed attack accuracy, comparing defense effectiveness of the three methods against different tools.

In this experiment, we test the adversarial sample detection method proposed for the mentioned datasets. We conducted 30 control experiments. The results are shown in Figure 3 to Figure 5:

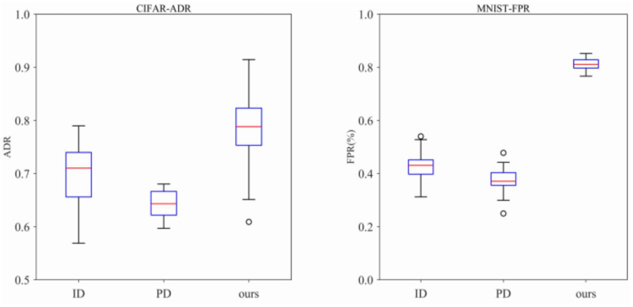

Figure 3. Experiments on the MNIST dataset (no adversarial attacks)

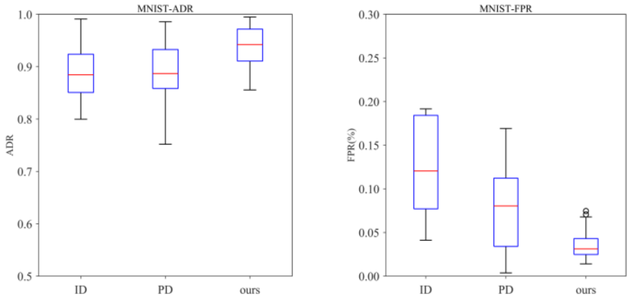

Figure 4. Experiments on the CIFAR-10 dataset (no adversarial attacks)

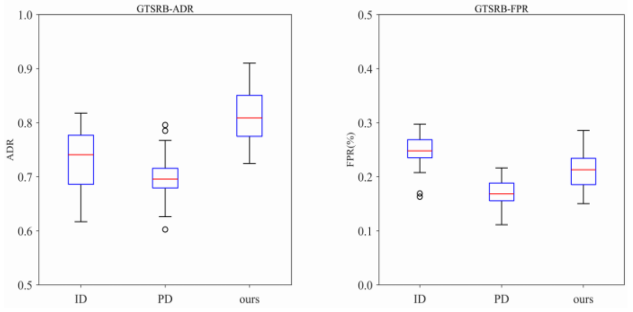

Figure 5. Experiments on the GTSRB dataset (no adversarial attacks)

The average results on the initial test set are in Table 2. The proposed method performs better in terms of adversarial sample detection rate and false positive rate. For the MNIST dataset, our method improves detection accuracy by 9.7% compared to I-Defender and reduces the false positive rate by

50% compared to Pixel-Defend. For CIFAR-10, it improves the detection rate by 10.3% compared to IDefender but has a lower misjudgment rate for benign samples. For GTSRB, it outperforms the other methods, improving the detection rate by 12. 1% compared to IDefender, but the misjudgment rate for benign samples is 23.5% lower than Pixel-Defend. Overall, the proposed method has a higher detection rate and can identify benign samples, proving its effectiveness.

Table 2. Adversarial sample detection experiment results (no adversarial attacks)

	Dataset	ADR	FPR
I-Defender	MNIST	0.896	0.0012
Pixel-Defend	MNIST	0.873	0.0006
AdvDetector	MNIST	**0.983**	**0.0003**
I-Defender	CIFAR-10	0.695	0.0043
Pixel-Defend	CIFAR-10	0.638	**0.0037**
AdvDetector	CIFAR-10	**0.767**	0.0081
I-Defender	GTSRB	0.728	0.0023
Pixel-Defend	GTSRB	0.685	**0.0017**
AdvDetector	GTSRB	**0.816**	0.0021

To demonstrate the ability and generalization of the proposed method, we will detect adversarial attacks under several attack methods. This experiment follows the universal standard, ignoring samples that cannot be successfully attacked.

We will conduct data analysis through 30 control experiments for different attack methods. The average values in the table, and the better results highlighted in bold.

Table 3 shows the accuracy of the three adversarial example detection methods under different adversarial attacks in this experiment. For all attacks, the proposed method in this study outperformed the other two detection methods, except for a slight decrease in detection rate compared to the PD method for the PGD-4 adversarial attack. In particular, the proposed method achieved a 97.7% detection rate for the DeepFool adversarial sample detection method and successfully detected 100% of cases in 30 controlled experiments. In white-box attacks, the proposed method showed significant improvement over the other two methods, with a detection rate increase of 108.9% compared to the ID method and 77.8% compared to the PD detection method.

Table 3. Adversarial sample detection methods on the MNIST dataset

	ADR (%)		
Attack-ε	I-Defender	Pixel-Defend	AdvDetector
DeepFool	0.753	0.882	**0.977**
C&W	0.649	0.902	**0.936**
FGSM-8	0.782	0.869	**0.916**
MIN-4	0.687	0.765	**0.831**
MIN-8	0.638	0.825	**0.855**
PGD-4	0.693	**0.837**	0.835
PGD-8	0.657	0.875	**0.895**
PGD-16	0.737	0.819	**0.836**
WB	0.371	0.436	**0.775**

For the CIFAR-10 dataset, Table 4 demonstrates that the proposed method maintains an advantage in detecting adversarial samples. The detection accuracy of the three methods varies with the DeepFool attack, especially for Pixel-Defend, which only detects 18.9% of adversarial samples. However, the proposed method increases the detection rate by 8 percentage points compared to IDefender. For the C& W attack method, all three detection methods perform well, with the proposed method achieving 99. 1% detection accuracy and even reaching 100% in multiple trials. The box plot in Figure 7 demonstrates that the proposed adversarial example detection method exhibited more stable detection rates across 30 controlled experiments compared to the other two methods. Figure 8 shows that it was more stable in 30 controlled experiments for different attacks compared to the other two methods.

Table 4. Adversarial sample detection methods on the CIFAR-10 dataset

	ADR (%)		
Attack-ε	I-Defender	Pixel-Defend	AdvDetector
DeepFool	0.536	0.189	**0.582**
C&W	0.831	0.984	**0.991**
FGSM-8	0.857	**0.992**	0.982
MIN-4	0.382	0.765	**0.811**
MIN-8	0.396	0.730	**0.804**
PGD-4	**0.725**	0.648	0.696
PGD-8	**0.784**	0.682	0.547
PGD-16	**0.810**	0.714	0.602
WB	0.165	0.139	**0.264**

Table 5 shows that the proposed method had a high detection accuracy for adversarial samples in the GTSRB dataset under various attack methods. The detection accuracy for the proposed method for DeepFool, C&W, and FGSM-16 attack methods was above 90%. In white-box attacks, the proposed method achieved a detection accuracy of 88.6%, which was an improvement of 161% and 123.8% compared to the other two adversarial sample detection methods.

Table 5. Adversarial sample detection methods on the GTSRB dataset

	ADR (%)		
Attack-ε	I-Defender	Pixel-Defend	AdvDetector
DeepFool	0.683	0.479	**0.925**
C&W	0.601	0.734	**0.971**
FGSM-16	0.611	0.884	**0.963**
MIN-8	0.325	0.791	**0.864**
MIN-16	0.384	**0.975**	0.881
PGD-4	0.412	**0.840**	0.812
PGD-8	0.327	0.841	**0.869**
PGD-16	0.386	**0.993**	0.886
WB	0.331	0.386	**0.864**

Based on the experimental results and analysis above, the proposed method of adversarial sample detection based on label fusion image generation network achieved good detection results compared to existing adversarial sample detection methods under different datasets and adversarial attacks. Especially in white-box attacks, the proposed method of adversarial sample detection on the CIFAR-10 dataset outperformed other detection methods significantly. The effectiveness of the proposed method has been validated through the experimental analysis.

Experiment 2: Data Generation Experiment Based on Adversarial Sample Detection

To validate the effectiveness of the data generation, this paper will conduct comparative experiments using the sample detection data mentioned in DeepSmartFuzz and compare it with DeepHunter. However, since DeepHunter and DeepSmartFuzz primarily focus on LeNet-3 and LeNet-5 [29] deep neural networks corresponding to the MNIST dataset, and do not experiment with the wide residual convolutional neural network (w-ResNet) used in this paper, we will compare some experimental results from the aforementioned papers and verify them using the open-source code from DeepSmartFuzz and DeepHunter.

In this experiment, the generation status of the MNIST dataset under the LeNet-5 network and the CIFAR-10 dataset under the W-ResNet network will be analyzed based on neuron coverage rate and adversarial sample detection rate. The experimental architecture of this method is shown in Figure 6.

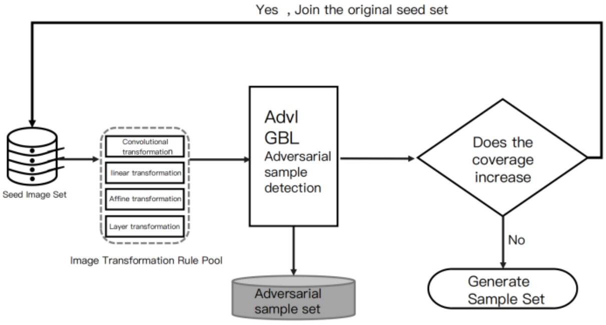
Yes , Join the original seed set
Seed Image Set
Convolutional transformation
linear transformation
Affine transformation
Layer transformation
Image Transformation Rule Pool
Advl
GBL
Adversarial sample detection
Does the coverage increase
No
Adversarial sample set
Generate Sample Set

Figure 6. Sample generation experiment flowchart

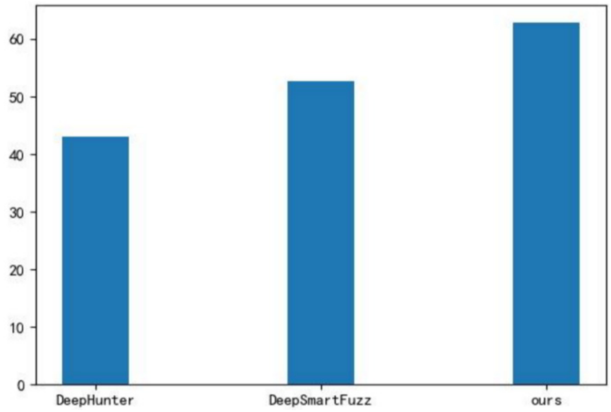

Figure 7. Bar chart of neuron coverage for CIFAR-10 dataset

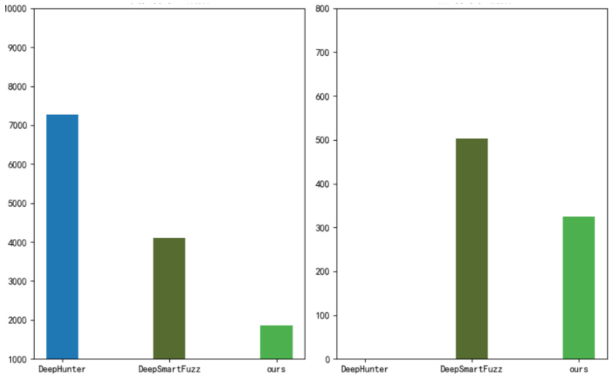

Figure 8. The number of generated samples and detected adversarial samples within 10 hours for the MNIST dataset under the W-ResNet network

Table 6 displays the average results of 10 data generation experiments on the MNIST dateset using the LeNet-5 network model. Compared to the DeepSmartFuzz method, the proposed method achieved the same coverage rate with only 785 generated samples, while DeepSmartFuzz generated 1024 samples. The CovG increased by 30% compared to DeepSmartFuzz, indicating higher sample quality using the proposed method. Although the proposed method detected fewer adversarial samples (139) due to fewer generated samples, its detection ratio of adversarial samples was 0.177 , higher than the DeepSmartFuzz method's.

Table 6. The average results under the LeNet-5 network model

	Neuron coverage rate	Sample generation quantity	Adversarial sample detection quantity
DeepHunter	99.8%	1765	0
DeepSmartFuzz	**100%**	1024	**163**
AdvDetector	**100%**	**785**	139

Table 7 presents the analysis of sample generation results for the MNIST dataset in the W-ResNet network using three experimental methods mentioned in this paper, at a coverage rate of 60%. Under equal coverage rates, the proposed method in this paper demonstrates significantly higher sample quality compared to the other two coverage-guided sample generation methods.

Table 7. Sample generation results in W-ResNet network for Cifar-10 dataset after 10 hours

	Neuron coverage rate	Sample generation quantity	Adversarial sample detection quantity
DeepHunter	43.1%	10703	0
DeepSmartFuzz	52.6%	4096	503
AdvDetector	**62.8%**	**2781**	**849**

Table 8 presents the sample generation results analysis for the three experimental methods used in this study on the MNIST dataset under W-ResNet network at 60% coverage rate. Due to the uncertainty of generated samples, the coverage rate of the three methods is higher than 60%. Under the same coverage rate, our proposed method can achieve the experimental requirements by generating only 1864 benign samples, which is far superior in sample quality compared to the other two coverage-guided sample generation methods.

Table 8. The results of sample generation with 60% coverage

	Neuron coverage rate	Sample generation quantity	Adversarial sample detection quantity
DeepHunter	62.4%	7265	0
DeepSmartFuzz	60.3%	4096	**503**
AdvDetector	61.2%	**1864**	325

In this study, we will demonstrate the effectiveness of our method by comparing the sample generation results under the autonomous driving dataset. We will prove the effectiveness of the method proposed in this paper by comparing it with the DeepSmartFuzz method. For the autonomous driving dataset, this experiment will be conducted using the Rambo deep neural network for training and testing. The Rambo deep neural network model is implemented with a 32-layer network architecture and can efficiently and accurately classify autonomous driving datasets.

As shown in Table 9, the proposed method achieved a 81.3% deep neural network neuron coverage under 12946 generated samples in 72 hours, while the DeepSmartFuzz generated 10240 samples with a final neuron coverage result of 72.6%. The proposed method had a single sample coverage rate of about 0.092 , which was about 29% higher than the single sample coverage rate of 0.071 in DeepSmartFuzz, indicating higher quality of generated samples. For adversarial sample detection, the proposed method detected 2360 adversarial samples, which was significantly higher than the 783 adversarial samples detected by DeepSmartFuzz. This result demonstrates the ability of the proposed method to accurately identify adversarial samples in complex sample scenarios.

Figure 9 shows the bar chart of the single sample coverage rate and the adversarial sample detection rate in this experiment.

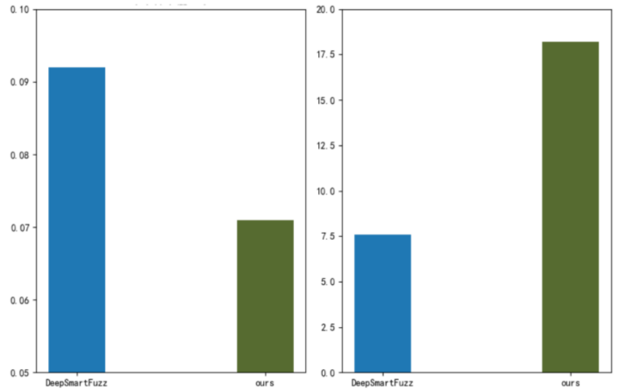

Figure 9. Bar graph of single sample coverage and adversarial sample detection ratio

Table 9. Sample coverage and results of generated samples within 72 hours

	Neuron coverage rate	Sample generation quantity	Adversarial sample detection quantity	Adversarial sample ratio
DeepSmartFuzz	72.6%	10240	783	7.6%
AdvDetector	81.3%	8931	2360	18.2%

5 Conclusion and Future Work

Image recognition is a major research direction in the current and future deep neural network studies. Due to the inherent characteristics of deep neural networks, the existence of adversarial samples is an unavoidable problem. Currently, research on finding methods to generate adversarial samples involves directly attacking the neural network through appropriate attack methods or using metamorphic testing to generate samples that do not affect the original labels. The former method may produce samples with large differences from the original samples, which may hinder the generalization of adversarial sample generation. The latter method requires a high level of understanding in the relevant field and manually defining rules, which is inefficient and may lead to rule correctness issues.To address the above issues, this paper proposes the method AdvDetector to solve the problem.

This paper verifies the applicability and effectiveness of the proposed method by applying it to existing open-source datasets such as MNIST, CIFAR-10, and the German standard traffic dataset GTSRB, as well as the classic deep neural networks LeNet-5 and W-ResNet.

In the future, we will explore approaches of tackling class imbalance problem faced by most HBR prediction scenarios. We will also work on constructing more effective classification algorithms to improve the performance of the model. Besides, future studies could put effort to provide interface for real-time interactive bug report labelling.

References

[1] N. Akhtar, A. Mian, Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey, *IEEE Access*, Vol. 6, pp. 14410-14430, February, 2018.

[2] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*, Toronto, Canada, 2018, pp. 2154-2156.

[3] K. Ren, T. Zheng, Z. Qin, X. Liu, Adversarial Attacks and Defenses in Deep Learning, *Engineering*, Vol. 6, No. 3, pp. 346-360, March, 2020.

[4] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial Examples: Attacks and Defenses for Deep Learning, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 9, pp. 2805-2824, September, 2019.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, *Intriguing properties of neural networks*, December, 2013. https://arxiv.org/abs/1312.6199

[6] Y, Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, K. Q. Weinberger, Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 8445-8453.

[7] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, R. Yang, ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 5452-5462.

[8] P. Li, X. Chen, S. Shen, Stereo R-CNN Based 3D Object Detection for Autonomous Driving, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 7644-7652.

[9] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, E. Brossard, The MegaFace Benchmark: 1 Million Faces for Recognition at Scale, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 4873-4882.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, F.-F. Li, Imagenet large scale visual recognition challenge, *International journal of computer vision*, Vol. 115, No. 3, pp. 211-252, December, 2015.

[11] D. Tang, F. Wei, B. Qin, T. Liu, M. Zhou, Coooolll: A deep learning system for twitter sentiment classification, *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, Dublin, Ireland, 2014, pp. 208-212.

[12] A.-L. Maas, P. Qi, Z. Xie, A.-Y. Hannun, C.-T. Lengerich, D. Jurafsky, A.-Y. Ng, Building dnn acoustic models for large vocabulary speech recognition, *Computer Speech & Language*, Vol. 41, pp. 195-213, 2017.

[13] A. Mousa, B. Schuller, Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, 2017, pp. 1023-1032.

[14] A. Rafieeinasab, A. Norouzi, S. Kim, H. Habibi, B. Nazari, D.-J. Seo, H. Lee, B. Cosgrove, Z. Cui, Toward high-resolution flash flood prediction in large urban areas–analysis of sensitivity to spatiotemporal resolution of rainfall input and hydrologic modeling, *Journal of Hydrology*, Vol. 531, pp. 370-388, December, 2015.

[15] I.-J. Goodfellow, J. Shlens, C. Szegedy, *Explaining and harnessing adversarial examples*, December, 2014. https://arxiv.org/abs/1412.6572

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2574-2582.

[17] G. Andresini, A. Appice, L. De Rose, D. Malerba, 2021, Gan augmentation to deal with imbalance in imaging-based intrusion detection, *Future Generation Computer Systems*, Vol. 123, pp. 108-127, October, 2021.

[18] B. Bigi, Using kullback-leibler distance for text categorization, *European conference on information retrieval*, Pisa, Italy, 2003, pp. 305-319.

[19] L. M. Bogdanova, S. Y. Nagibin, A. S. Chemakin, Analysis of the Criteria for Assessing the Forecast Quality of Industrial Safety Indicators of Enterprises, *International Journal of Performability Engineering*, Vol. 17, No. 6, pp. 519-527, June, 2021.

[20] Y. LeCun, C. Cortes, C. Burges, *Mnist handwritten digit database*, 2010.

[21] A. Krizhevsky, V. Nair, G. Hinton, *Cifar-10 and cifar-100 datasets*, 2009. https://www. cs. toronto. edu/kriz/cifar. html

[22] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: The German traffic sign detection benchmark, *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, 2013, pp. 1-8.

[23] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: Automated testing of deep-neural-network-driven autonomous cars, *Proceedings of the 40th international conference on software engineering*, Gothenburg, Sweden, 2018, pp. 303-314.

[24] Z. Zheng, P. Hong, Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks, *Advances in Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 7924-7933.

[25] Y. Song, T. Kim, S. Nowozin, S. Ermon, N. Kushman, *PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples*, October, 2017. http://arxiv.org/abs/1710.10766

[26] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, S. See, Deephunter: a coverage-

guided fuzz testing framework for deep neural networks, *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, *ISSTA 2019*, Beijing, China, 2019, pp. 146-157.

[27] S. Demir, H. F. Eniser, A. Sen, *Deepsmartfuzzer: Reward guided test generation for Deep learning*, November, 2019. http://arxiv.org/abs/1911.10621

[28] D. Perez, S. Samothrakis, S. Lucas, Knowledge-based fast evolutionary MCTS for general video game playing, *2014 IEEE Conference on Computational Intelligence and Games*, Dortmund, Germany, 2014, pp. 1-8.

[29] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2324, November, 1998.

Biographies

Xiaoxue Wu assistant professor of Yangzhou University. She received the Ph.D degree in Cyberspace Security from Northwestern Polytechnical University in June 2021. Her main research direction is software quality assurance and testing. She has published more than 15 papers in prestigious journals of software testing.

Shuqi Zuo undergraduate student at Yangzhou University.

Shiyu Weng graduate student at Yangzhou University.

Yongkang Jiang undergraduate student at Yangzhou University.

Hao Huang graduate student at Northwestern Polytechnical University.