TranFuzz: An Ensemble Black-Box Attack Framework Based on Domain Adaptation and Fuzzing

Hao Li[1], Shanqing Guo[1(✉)], Peng Tang[1], Chengyu Hu[1,2], and Zhenxiang Chen[3]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, China
202020883@mail.sdu.edu.cn, guoshanqing@sdu.edu.cn

[2] Key Laboratory of Network Assessment Technology, CAS (Institute of Information Engineering, Chinese Academy of Sciences), Beijing 100093, China

[3] University of Jinan, Jinan, China

Abstract. A lot of research effort has been done to investigate how to attack black-box neural networks. However, less attention has been paid to the challenge of data and neural networks all black-box. This paper fully considers the relationship between the challenges related to data black-box and model black-box and proposes an effective and efficient non-target attack framework, namely TranFuzz. On the one hand, TranFuzz introduces a domain adaptation-based method, which can reduce data difference between the local (or source) and target domains by leveraging sub-domain feature mapping. On the other hand, TranFuzz proposes a fuzzing-based method to generate imperceptible adversarial examples of high transferability. Experimental results indicate that the proposed method can achieve an attack success rate of more than 68% in a real-world CVS attack. Moreover, TranFuzz can also reinforce both the robustness (up to 3.3%) and precision (up to 5%) of the original neural network performance by taking advantage of the adversarial re-training.

Keywords: Domain adaptation · AI security · Fuzzing · Black-box attack

1 Introduction

Recently, Deep Neural Networks (DNNs) have been applied to many realistic AI systems, such as image classification [1]. However, due to the catastrophic overfitting or underfitting problem, the DNN-based systems always show a very vulnerable behavior in many corner cases [2]. In other words, if the DNN models are not tested in particular corner cases (which is referred to as the adversarial example, i.e., clean data that adds a well-designed noise), there would be devastating consequences. Thus, like traditional software testing, it is particularly important to systematically test and check the quality of the DNN-based models.

D. Gao et al. (Eds.): ICICS 2021, LNCS 12918, pp. 260–275, 2021.
https://doi.org/10.1007/978-3-030-86890-1_15

There exists plenty of works to test the quality of DNN-based systems by taking advantage of the model attack method [2,3,20]. These works always belong to white-box attacks, where the adversary usually first draws upon knowledge of the structure and parameters of the target model and then injects some imperceptible perturbation into a test example to create an adversarial example, and attacks the victim model. Nevertheless, in real-world scenarios, the target victim model is always a black box, where an attacker cannot access a complete knowledge of the target model. This will increase the attack difficulty. How to successfully attack the target model under the black-box condition is a very challenging issue.

To solve the above problem, the researchers have proposed two kinds of adversarial black-box attack methods: i) *Query-based attack* method [5,15], where the target black-box model is treated as an optimization problem, and the attackers use query prediction information (for example, a probability value) as an instruction to generate adversarial examples. Although the query-based attack method usually has a high attack success rate, it does take a very high number of requests, which will incur a higher cost [6]. ii) *Transfer-based attack* method [4,20], where the adversary ought to construct a comparable model as the local substitution to the target, and construct highly transferable adversarial examples that can successfully attack the local model. Then, these adversarial examples are transferred to attack the target black-box model. In this paper, we mainly focus on the transfer-based attack.

Unfortunately, the existing transfer-based attack methods [4,20] usually only consider that the network architecture of the target model is a black box, but do not consider the target training data is a black box, and it is assumed that the training data of the source and target models are following the same data distribution [20]. However, it is not the case for practical application. As a result, these methods often suffer from low attack success rates or poor transfer efficiency in the black-box attack task. To remedy the above deficiency, in our black-box attack problem, we summarize the following two key challenges:

C1: Target Training Data Is Black-Box. Due to commercial confidentiality, the training data used in the target model will not be publicly available, and only some of the test/validation data examples may be provided to developers with some special scenes (e.g., adversarial competition[1]). This will lead to a different data distribution between training data used in the local (or source) model and that used in the target model. How to exploit the limited test/validation data examples of the target model to achieve a successful attacking purpose is a critical challenge.

C2: Target Model's Network Is Black-Box. Developing an effective adversarial example always requires that the attacker has complete information about

[1] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset.

the target model, such an attack method is also known as a white-box attack. However, the attacker is generally unaware of what kind of neural network architecture is implemented in the target model. How to successfully tackle the target model under the $C2$ is another major challenge.

In this paper, we fully consider and explore the relationship between challenges $C1$ and $C2$. On the one hand, to address the challenge of different data distributions, we propose a methodology based on domain adaptation to reduce the difference between the data of the source and target domains by using subdomain mapping. Based on this method, we can implement a transfer-based attack. On the other hand, to counter the challenge of the model's black-box ($C2$), we illustrate an adversarial example (AE) generation framework based on neuron coverage to measure the logical runtime of the DNN model. Within our fuzzing framework, we also propose a novel ensemble-based seed mutation strategy to improve AEs attack transferability. The strategy introduces a small change in input mutations and to maximizes the expected difference between the original and the adversarial example. Certainly, there exist some transfer-based adversarial attack methods [3,20] which are used in iterative gradient attack methods [20]. But they do not have a guide for exposing incorrect corner case behaviors. This can result in incorrect DNN behaviors remaining unexplored after thousands of iterations (low transferability caused by overfitted issues [3]).

In the end, we design a black-box model attack framework, namely, TranFuzz, which combines the domain adaptation method with the fuzzing strategy. Evaluation experiments show that the proposed TranFuzz method is best able to achieve an attack success rate of over 68% in the real-world Cloud Vision Service (CVS) scenario.

Summary of Contributions – The major contributions to this paper are shown as follows:

- We propose a black-box attack framework, namely, TranFuzz, which can generate highly transferable adversarial examples by interconnecting the domain adaptation-based local alternative model construction method and fuzzing-based method, respectively. To the best of our knowledge, it is the first work to combine domain adaptation and fuzz methods against the black-box model.
- TranFuzz[2] takes full account of the challenges of the data black-box and the neural network black-box. To create highly transferable AEs, we propose an ensemble-based seed mutation strategy, which can rapidly and efficiently trigger objective functions in our fuzzing framework. The experimental results show that the average attack success rate of TranFuzz can exceed 10%, compared to the state-of-the-art baselines.
- In five real-world Cloud Vision Services (i.e., Aliyun, Baidu, Tencent, Azure, and Clarifai) attacking scenes, the TranFuzz can better perform over 68% attack success rate. Furthermore, our proposed method can also enhance the robustness of the victim's model with adversarial training.

[2] https://github.com/lihaoSDU/ICICS2021.

Organization – The remainder of this paper is organized as follows. We present an overview methodology of the TranFuzz framework in Sect. 2. In Sect. 3, we illustrate details of the evaluation experiments and results. Section 4 highlights the related work of the paper and we conclude our work in Sect. 5.

2 Methodology

In this section, we first introduce an overview of the TranFuzz (Sect. 2.1), and then we illustrate the local model construction method based on domain adaptation to break the barrier of the data black-box challenge (Sect. 2.2). Finally, we generate optimal adversarial examples with high transferability by presenting a fuzzing-based method to address the neural network black-box challenge (Sect. 2.3).

2.1 Overview of TranFuzz

TranFuzz takes full account of the unique nature of the data black-box challenge co-existing with the model black-box challenge in the realistic application scene. In this paper, we design an effective adversarial example generation framework, named TranFuzz. The TranFuzz framework is depicted in Fig. 1. In TranFuzz, we first develop an algorithm based on a deep sub-domain adaptation network (DSAN) to construct a local substitute model. Afterward, we manufacture an adversarial example of high transferability on the strength of the mutation-based fuzzing strategy.

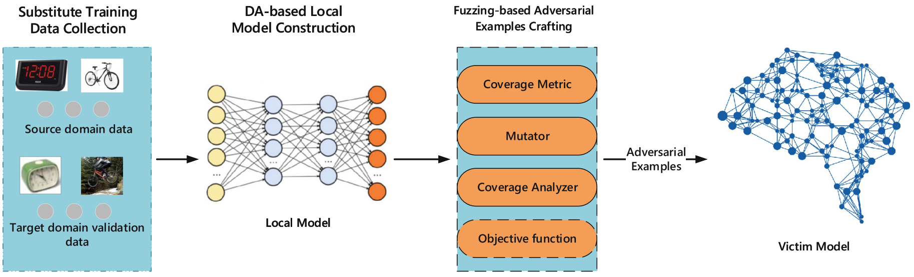

Substitute Training Data Collection
Source domain data
Target domain validation data
DA-based Local Model Construction
Local Model
Fuzzing-based Adversarial Examples Crafting
Coverage Metric
Mutator
Coverage Analyzer
Objective function
Adversarial Examples
Victim Model

Fig. 1. Framework of TranFuzz.

2.2 Domain Adaptation-Based Local Model Construction

In the data black-box challenge, we assume that the attacker cannot get the target model's training data. Only unsupervised validation/test data can be accessed. To resolve the problem mentioned above and construct a local replacement model, in this section, TranFuzz uses a deep subdomain adaptation network

(DSAN) algorithm [9] with a certain improvement of the DSAN pseudo-labeling part. DSAN uses a classification loss function and an adaptive loss function to close the huge gap between the data of the source and target domain. We formulate the objective function of the DSAN as:

$$\min_{f} \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i^s), y_i^s) + \sum_{l \in L} LMMD_l(p, q) \tag{1}$$

where $J(\cdot,\cdot)$ is a cross-entropy function, $f(\cdot)$ is the predict function, n_s is the number of a source domain's samples, and x^s and y^s corresponding to the source domain's samples with the label. $LMMD(\cdot,\cdot)$ is the *local maximum mean difference* function to calculate the loss in the process of subdomain domain adaptation. The l is an active layer in the subdomain distribution L. p and q are the data distributions of the source and target domains. To address the challenge of the data black box, our optimization goal is to minimize Eq. 1 under the conditions of different data distributions between p and q.

From Eq. 1, DSAN leverages an algorithm based on domain adaptation networks (DANs) [10] and designs a local maximum mean difference as the difference metric between source and target domains. To compute the LMMD and reduce the data distribution difference between source and target domains. In our proposed method, we leverage a query-based strategy that adopts the target victim model as a benchmark to predict the target samples. For each sample, a single request is necessary for our proposed method. The improved method can significantly enhance the generalization capabilities of the local surrogate model's construction. Formally, the proposed method is formalized in Eq. 2.

$$LMMD_l(p, q) \overset{\triangle}{=} \mathbf{E}_c ||\mathbf{E}_{p^{(c)}}[\phi(x^s)] - \mathbf{E}_{q^{(c)}}[f^t(x^t)]||^2 \tag{2}$$

where $\mathbf{E}[\cdot]$ is the mathematical expectation function, c is the different classes (e.g. labels), x^t is the target domain's example. $\phi(\cdot)$ means the feature mapping function. In this paper, we use a universal function of the *Gaussian Kernel* as a mapping function between source and target domains. Our proposed approach can be applied to any neural network architecture to construct a local model with a promising performance.

2.3 Fuzzing-Based Adversarial Examples Crafting

In this section, we leverage the coverage-based fuzzing method to fuzz our local substitution model and generate high transferability adversarial examples. In the following chapter, we first describe the coverage gauge to guide our fuzzing neural network framework. Then we also elucidate the fuzzing objective function. Second, we describe our coverage analysis method (also called coverage analyzer) of the TranFuzz. Finally, we propose a new comprehensive mutation strategy to generate highly transferable adversarial examples.

Definitions of the Neuron Coverage and Objective Functions. The following are several definitions that are used in the fuzzing framework.

Neuron Coverage. TranFuzz exploits neuron coverage (NC) as our fuzzing coverage criteria, proposed by DeepXplore [2]. Neuron coverage is a metric for testing the comprehensiveness of the DNN model. NC can also calculate how many neurons are at least activated once during the current process. The formula of the neuron coverage is shown below:

$$NCov(TS, seed) = \frac{|\{n_i|\forall seed \in TS, f(n_i, seed) > th\}|}{K} \tag{3}$$

where TS is a set of test seeds $\{ts_1, ..., ts_n\}$. We suppose all neurons of the model as $N = \{n_1, ..., n_K\}$, K is the number of neurons in the model, th is the fuzzing threshold to be considered as an activated neuron (in this paper we define the value as zero). $f(\cdot)$ is the function that allows you to send back the output value of the neuron. It is worth mentioning that in our proposed TranFuzz framework, we did not intentionally pursue a higher neuron coverage as our optimization objective. We took neuron coverage as a guide instruction metric to discover more exceptional adversarial examples that can crash the local substitute model.

Objective Functions. TranFuzz fully considers the high transferability and human imperceptibility as the objective fuzzing functions to craft adversarial examples. If an adversarial example complies with the objective function constraint, the fuzzing process will jump out of the execution loop.

On the one hand, TranFuzz loosens the differential testing objective function used in [2] and proposed a novel function, specifically,

$$obj_{DX} : O_{f_1(x)} \cup O_{f_2(x)} \cup ...O_{f_k(x)} = 1, \tag{4}$$

where $f(\cdot)$ is the predicted function of local models, $\cup$ is the union function, k is the number of local models. If $f_i(x)$ is not equal to the true label of x, then $O_{f_i(x)} = 1$. In addition, the differential testing method requires several local DNN models with the same prediction task, which will increase training costs. Unlike the differential testing method above-mentioned, TranFuzz only needs a local substitution model as our fuzzing framework input ($k = 1$ in Eq. 4). Accordingly, one of the fuzzing objective functions obj^1_{TF} in TranFuzz describes as following:

$$obj^1_{TF} : f(x_{adv}) \neq true\ label\ of\ x_{adv} \tag{5}$$

On the other hand, to generate imperceptible adversarial examples, the structural similarity between adversarial examples and the original examples is another objective function used in the fuzzing framework. We introduce an Average Structural Similarity (ASS) [11] as the similarity metric. To deduce structural changes, ASS captures pixel intensity patterns, especially among adjacent pixels. ASS can also measure the brightness and contrast of the image which affects the perceived quality of the image. The formalization of the objective function based on structural similarity is

$$obj_{TF}^{2} : ASS(x_{avd}, x) > \tau \tag{6}$$

where τ is the pre-setting threshold value of ASS, we set it as 0.96 in our evaluation experiments. In the end, we combine obj_{TF}^{1} and obj_{TF}^{2} as our objective functions. If these conditions are triggered, the current fuzzing loop will shut down.

Coverage Analyzer. In the coverage parser section, if the adversarial example x_{adv}^{i} has not satisfied the objective functions, the coverage analyzer will randomly select unfuzzed network layer neurons. After that, the coverage analyzer will split a new fuzzing path. Next, the coverage analyzer will calculate the current neuronal loss values and gradient values $grads^{i}$. The coverage analyzer will combine the x_{adv}^{i} and $grads^{i}$ as the *Mutator* inputs. If the x_{adv}^{i+1} can reach the objective functions, the coverage analyzer will update the neuron coverage and break the current fuzzing loop.

Mutator. Mutator is the schedule against too many fuzzing execution iterations. The mutator can also craft adversarial examples of high transferability and imperceptibility. In this section, TranFuzz proposes a novel ensemble-based mutation method that leverages multiple perturbation strategies to generate adversarial examples. The formal representation is $x_{adv} = x + \delta$, where δ is the optimal adversarial perturbation.

The mutator is based on the gradient value which is computed by the local surrogate model output layer and the hidden layer. To generate adversarial perturbation δ_{dx}, we adopt the occlusion strategy described by DeepXplore to simulate the camera lens that may be accidentally or deliberately occluded. In contrast to DeepXplore's strategy, we are implementing a smaller occlusion adversarial perturbation $\delta_{DX} = occlusion_{i:i+m,j:j+n}$ (smaller rectangle that has $m * n$ pixels, and (i, j) is the coordinate of a pixel) and operating randomly in multiple seed positions.

Moreover, to improve the success of the attack under the premise of imperceptibility, TranFuzz does not implement general mutation methods with various fuzzers, e.g., image blurring, image contrast adjusting, image brightness adjusting [13]. The TranFuzz proposed a novel method based on the scale of images [12] to transform adversarial perturbations. The mutator first leverages the cumulative distribution function (CDF) to calculate the equalization of the image perturbation histogram. The formalization of the histogram equalization function $(T(\cdot))$ is as follows:

$$T(r_k) = \frac{G-1}{MN} \sum_{j=0}^{k} n_j, k = 0, 1, ..., G-1 \tag{7}$$

where MN is the sum of pixels, r_k is the gray level of the image, $\sum_{j=0}^{k} n_j$ is the number of r_k, and G is the number of possible gray levels of the image.

After equalizing the histogram, we adopt the linear interpolation method to insert the initial Gaussian derived noise from the gradient. Then we calculated the adversarial perturbation δ_{TF}. To avoid invalid values in δ_{TF}, the mutator also quantifies the adversarial perturbations [14]. Finally, we describe the optimal adversarial perturbation as $\delta = \delta_{DX} + \delta_{TF}$.

3 Evaluation

In this section, we first introduce specific details of the data sets adopted in our evaluation experiments, then the attack configurations are also depicted in this section (Sect. 3.1). After that, we attack both the non-robustness (Sect. 3.2) and robustness (Sect. 3.3) of the black-box models that leverage the adversarial examples generated by TranFuzz. We also compare our method with eight state-of-the-art baseline methods. In addition, five different business Cloud Vision services are conducted as black-box victim targets in real-world scenarios (Sect. 3.4). We also analyze the positive impacts of the proposed method on the target model defensibility by leveraging adversarial retraining (Sect. 3.5).

3.1 Experimental Setting

Datasets. To build black-box data with experimental domain adaptation environments, we use two different image data sets (namely, Office-31 [16] and Office-Home [19]), which are often the benchmark dataset in the domain adaptation field. In our evaluation experimental setting, to simulate the $C1$ challenge, different categories are regarded as the source and target domains (specifically, *Amazon* and *Webcam* of the Office-31, *Product* and *RealWorld* of the Office-Home), respectively. All domain adaptation data are downloaded from the open-source site[3].

Attack Configurations. The settings for black-box model attacks and evaluation baselines are described in this section.

Black-Box Model Setting. ResNet50, AlexNet, VGG19, and DenseNet121 are different models of neuronal network structures. We conducted the abovementioned models in our black-box attack evaluation experiments. Moreover, to build the experimental configuration about the model black-box under the $C2$ challenge, we use the ResNet50 neural network as the source domain model. The other three models are as the target domain attacked model. For example, under the $C2$ challenge, on the one hand, the source domain model with its training data is the ResNet50 and *Webcam*. On the other hand, the target domain model with its training data is defined as DenseNet121 and *Amazon*. While the two different models all have 31 different classes (`backpack`, `bike`, etc.), the training data of the models are different and obeys the $C1$ challenge.

[3] https://github.com/jindongwang/transferlearning/tree/master/data.

Baselines. 1) In the non-robustness black-box model attack, we leverage eight different state-of-the-art attack methods as the baselines in our comparison experiments. Specifically, five white-box attack methods (namely, DDN [14], PGD [4], FGSM [20], L-BFGS [17], C&W [18]), and three black-box attack methods (ZOO [15], Pixel Attack (PA) [21], and Spatial Transformation (ST) [22]). 2) In the robustness black-box model attack task, we use adversarial training algorithms to build robust models as the victim models. The compared baselines are the same as the above-mentioned in step 1). The details of the adversarial training algorithms are Fast is Better than Free (FBF) [25] and Madry's Protocol [4]. 3) To demonstrate the effectiveness of our proposed approach in real-world black-box attack scenarios, we are also attacking five state-of-the-art commercial Cloud Vision Services (CVS), namely, Aliyun[4], Baidu[5], Tencent[6], Azure[7], and Clarifai[8].

Implementation Details. 1) In our evaluation experiments, all the training iterations numbers are set as 200. 2) Also, implementation information of the eight baseline methods are depicted in the AdverTorch [23] and ART [24]. It is worth noting that the Spatial Transformation parameters of `max_translation` and `max_rotation` are all equal to 30 degrees (which is consistent with the [22]). Other employment parameters are all set to default. 3) In the implementation of adversarial training, we conduct the ART tool to re-train the robustness models (`AdversarialTrainerFBF` and `AdversarialTrainerMadryPGD`). The maximum number of training iterations also is set to 200. The maximum perturbation parameter with its step is set to 0.3 and 0.1, respectively. 4) Considering the cost of the commercial Cloud Vision Services (e.g., one thousand API access will need to be 3$ for Aliyun), we randomly select 50 adversarial examples to attack the five CVS which are generated by TranFuzz. The detailed description is shown in Sect. 3.4. Among the five CVS, we access the provided API of Aliyun, Baidu, and Clarifai. In addition, the Tencent and Azure attacks are making use of web browsers upload manually. 5) We randomly divided the target domain data into the train (80%) and test (20%) parts. The train part is for training the target victim model and the test part is for constructing the source local substitute model. 6) In the experimental evaluation, we perform Attack Success Rate[9] to evaluate our proposed framework.

We provide a summary of our trained DNN-based models of the target domain in Table 1.

[4] https://vision.aliyun.com/imagerecog.

[5] https://ai.baidu.com/tech/imagerecognition/general.

[6] https://ai.qq.com/product/visionimgidy.shtml.

[7] https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision.

[8] https://www.clarifai.com/label.

[9] The Attack Success Rate is the proportion of adversarial examples misclassified by the target DDN [14].

Table 1. Summary of the target DNN-based models

Target model type	Dataset	Index	Architecture	Train & test data	Testing accuracy	Training iterations
Non-robust	Office-31	1	AlexNet	Amazon	77.53%	200
		2	AlexNet	Webcam	95.91%	200
		3	VGG19	Amazon	83.62%	200
		4	VGG19	Webcam	91.81%	200
		5	DenseNet121	Amazon	79.97%	200
		6	DenseNet121	Webcam	93.57%	200
Non-robust	Office-Home	7	AlexNet	Product	80.22%	200
		8	AlexNet	RealWorld	69.6%	200
		9	VGG19	Product	85.68%	200
		10	VGG19	RealWorld	80.4%	200
		11	DenseNet121	Product	85.68%	200
		12	DenseNet121	RealWorld	82.85%	200
Robust (FBF)	Office-31	13	DenseNet121	Amazon	62.89%	200
		14	DenseNet121	Webcam	83.04%	200
Robust (Madry's Protocol)	Office-31	15	DenseNet121	Amazon	52.88%	200
		16	DenseNet121	Webcam	76.02%	200

3.2 Black-Box Attack Against Non-robustness Model

This section primarily describes our proposed method to attack the non-robustness black-box model's performance. Two different image datasets (Office-31 and Office-Home) were implemented in the experiments, mentioned in Sect. 3.1. Details of the comparison experimental results are given in Table 2 and Table 3.

On the one hand, we use ResNet50 as the source neural network to train a local substitute model for attacking other three different networks. Training data of the local substitute model differs from the target model on the promise of C1 challenge. From Table 2, TranFuzz can better achieve more than 33.9% and 37.5% attack success rates on *Webcam* and *Amazon* data, respectively. On average, TranFuzz can perform a Top-1 attack success rate (specifically, 25.6%) compared to the other baseline methods. On the other hand, from Table 3, comparison experiments also use ResNet50 as the source neural network. The TranFuzz can mislead the target victim model over 31.3% and 46.9% on *RealWorld* and *Product* data, respectively. For the average attack success rate, TranFuzz is also capable of making the Top-1 success rate (28.5%) compared to other baseline methods.

From Table 2 and 3, we observe that AlexNet is not robust against the nine different black-box attack methods, compared with the DenseNet121 and VGG19. This demonstrates that the defender should design a more robust and complex network structure to enhance the DNN-based model's performance. Furthermore, the C&W attack is one of the most effective and widely used among the primary attacks. From the evaluation experiments of non-robustness black-box attack, our proposed method can surpass the C&W method over 2.38% and 6.4% in Office-31 and Office-Home datasets. In addition, the L-BFGS attack uses L-BFGS to minimize the distance of the original and perturbed images.

Table 2. The success rate of attacks against the non-robustness black-box model in the Office-31 dataset

Source model	Source data	Target model	Target data	Attack								
				DDN	PGD	FGSM	L-BFGS	C& W	ZOO	PA	ST	TranFuzz
ResNet50	Amazon	AlexNet	Webcam	12.9%	19.2%	22.4%	21.64%	25.25%	1.92%	**34.1%**	33.9%	33.9%
		VGG19		9.9%	14.6%	17.4%	15.35%	20.05%	2.44%	17.9%	**26.9%**	19.3%
		DenseNet121		16.9%	25.6%	18.8%	22.22%	**27.49%**	1.22%	11.9%	11.7%	18.1%
	Webcam	AlexNet	Amazon	10.6%	19.9%	23.4%	10.28%	20.04%	2.93%	22.3%	35%	**37.5%**
		VGG19		10.6%	11.7%	12.9%	10.62%	20.9%	1.18%	11.9%	**23.7%**	20.7%
		DenseNet121		11.1%	19.3%	18.1%	14.12%	25.61%	2.34%	8.4%	16.2%	**24.2%**
Average attack success rate				12.0%	18.4%	18.8%	15.71%	23.22%	2.01%	17.8%	24.5%	**25.6%**

Table 3. The success rate of attacks against the non-robustness black-box model in the Office-Home dataset (DN121: DenseNet121)

Source model	Source data	Target model	Target data	Attack								
				DDN	PGD	FGSM	L-BFGS	C& W	ZOO	PA	ST	TranFuzz
ResNet50	Product	Alexnet	RealWorld	13.0%	18.0%	11.8%	17.93%	21.49%	2.1%	21.5%	30.7%	**31.3%**
		VGG19		14.5%	22.3%	19.6%	13.92%	20.71%	2.8%	16.4%	**29.9%**	24.8%
		DN121		13.4%	**26.9%**	19.9%	13.25%	21.27%	1.9%	11.0%	21.7%	20.7%
	RealWorld	AlexNet	Product	13.5%	23.1%	24.8%	16.29%	23.72%	2.1%	15.0%	36.7%	**46.9%**
		VGG19		14.9%	18.5%	19.2%	14.1%	**24.15%**	1.9%	10.0%	23.5%	22.5%
		DN121		12.1%	**30.4%**	20.9%	10.27%	21.23%	0.1%	8.1%	20.9%	25.1%
Average attack success rate				13.6%	23.2%	19.4%	14.29%	22.1%	1.8%	13.7%	27.2%	**28.5%**

The TranFuzz method also can be better than L-BFGS under the premise of C1 and C2 (specifically, 9.9% in Office-31, 14.2% in Office-Home).

Besides, from the results of the experiment, we also observe that the Spatial Transformation (ST) method has a promising effect on the local black-box model attack under $C1$ and $C2$ challenges, but is still weaker than TranFuzz (4.2% lower than us). From the adversarial examples generated by ST, we conclude that the ST method is not similar to the original natural structure of the images. Hence, from the ST [22] evaluation result, the adversarial example will have a partial loss compared to the original due to image rotations. Consequently, the target black-box model cannot predict the successful adversarial examples which trade by spatial transformation. While the adversarial examples generated by TranFuzz can retain the original ASS leverages our proposed image mutating strategy (examples of the AE can be found on our website mentioned before).

3.3 Black-Box Attack Against Robustness Model

In our evaluation experiments, to achieve the proposed robust networks as a black-box victim model, we implement commonly adversarial training methods, specifically, Fast Is Better Than Free (FBF) [25] and Madry's Protocol [4]. Furthermore, the neural network structure of the target model is also set as DenseNet121, and the source model is ResNet50. The data set that we have implemented in this section is the same as in Table 2. Detailed information on the deployment and implementation of robust models can be found in Sect. 3.1, Table 1.

Table 4. The success rate of attacks against the robust black-box model

Adversarial trainer	Source data	Target data	Attack								
			DDN	PGD	FGSM	L-BFGS	C& W	ZOO	PA	ST	TranFuzz
FBF [25]	Amazon	Webcam	38.0%	9.9%	11.7%	9.64%	15.32%	11.7%	48.3%	56.7%	**74.8%**
	Webcam	Amazon	6.4%	3.0%	1.9%	7.33%	18.58%	0.5%	24.8%	51.0%	**53.8%**
Madry's Protocol [4]	Amazon	Webcam	5.3%	1.2%	1.8%	19.89%	22.81%	1.2%	31.7%	**64.3%**	53.8%
	Webcam	Amazon	4.9%	4.0%	3.0%	4.18%	8.88%	2.1%	6.8%	9.9%	**13.6%**
Average attack success rate			18.1%	8.9%	9.1%	10.26%	16.4%	8.6%	29.4%	44.4%	**48.6%**

Table 4 shows the black-box attacks result against two distinct robustness models between TranFuzz and the other eight baseline methods. According to the table, TranFuzz can achieve a maximum attack success rate of 74.85%, and the proposed method also is able to accomplish an average attack success rate of 48.6%. From the evaluated experiment results, we can conclude that our proposed method is optimal compared with others. Additionally, we also observe that Madry's and FBF's defense methods are effective in resisting gradient attacks (i.e., FGSM-based attacks like PGD, L-BFGS, and FGSM). Specifically, in PGD, FGSM, and L-BFGS, the worst one only reaches 1.2% attack success rate, and the average attack success rate only achieves 8.9%, 9.1%, and 10.26%, respectively. The attack success rate has dropped by more than half compared to Table 2, Sect. 3.2. Besides, the C&W method can achieve a success rate of 16.4%, which is also lower than the proposed TranFuzz method. Nevertheless, Madry's protocol and FBF defense methods are unable to effectively defend Spatial Transformation and TranFuzz methods. The reason is that TranFuzz performs an ensemble-based AEs generation method to enhance transferability. But the Spatial Transformation algorithm adopts the technique of image transformation in space that will decrease the imperceptible performance of the image. In addition, the TranFuzz method can also exceed the ST method on a mean attack success rate of 4.2%.

Accordingly, based on the above-mentioned investigation, we have put forward some hypotheses and conjectures for the defense strategy here. The defender should consider various algorithms to generate adversarial examples (e.g., gradient-based, space-based, and color-based changes strategy) under the process of building an adversarial trainer. In our evaluation experiments, we also use adversarial examples generated by TranFuzz as the robust retraining data to defend against other attack methods. The implementation details are illustrated in Sect. 3.5.

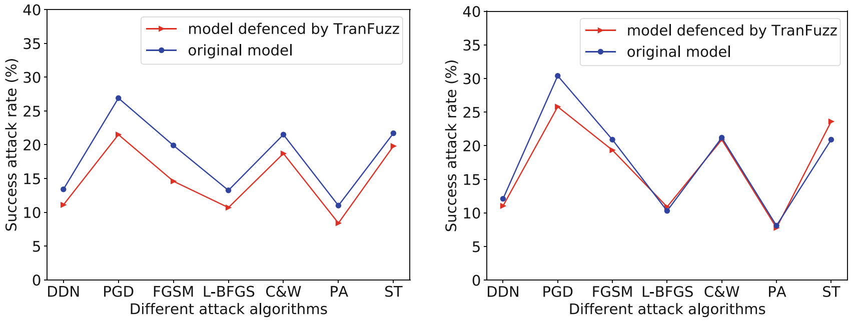
3.4 Black-Box Attack Against Cloud Vision Services

In this section, we focus on the black-box attack in real-world scenarios. The attacking targets are five different businesses Cloud Vision Services (Aliyun, Baidu, Tencent, Azure, and Clarifai). Considering the cost issue mentioned in Sect. 3.1, we randomly select 50 images from the Office-Home data set and develop adversarial examples using our proposed approach. It should be noted that we define a new attack success metric: if the Top-1 prediction tag (which access from Cloud Vision Services response) is different between the adversarial example and the original natural one, we consider the attack as successful.

On the one hand, we call the API provided by Aliyun, Baidu, and Clarifai and return the detection result. After that, the experimental results of the attack success rate are being calculated. On the other hand, we also take advantage of a method that manually uploads the picture via the web browser (Tencent and Azure), and we also record the response detection results on each of the adversarial examples. Ultimately, the success rate for the attack on Aliyun, Baidu, Tencent, Azure, and Clarifai is 19/50, 18/50, 34/50, 13/50, and 8/50 respectively. From the CVS detection results, our proposed method can perform a 68% higher attack success rate on Tencent, and can also do 16% to attack Clarifai even though it is the worst.

3.5 Adversarial Defending

In this section, we demonstrate that TranFuzz can also enhance the robustness of the target model. To meet this objective, we are focusing on an adversarial training strategy based on additional data. We retrain the victim model from scratch on the union of the TranFuzz crafted adversarial examples with the original natural images. In this section, to retrain the target neural network, we implement the Office-Home dataset and DenseNet121 shown in Table 3. The maximum number of training iterations is also set to 200.



(a) Defended model in *RealWorld*'s data detection.

(b) Defended model in *Product*'s data detection.

Fig. 2. Comparison with success attack rate before and after TranFuzz defend in seven different attack methods.

In the evaluation experiments, we defend against other Top-7 different attacking methods (DDN, PGD, FGSM, L-BFGS, C&W, PA, and ST). The source model also is the ResNet50 and the defense results are illustrated in Fig. 2. From Fig. 2(a), our proposed model defending method is implemented on the DenseNet121 that can hamper more than 3.3% average success rate on the non-target attacks, compared with the non-defense original model. Furthermore, from Fig. 2(b), the retrained model can also improve robustness performance by around 1% on average in the *Product* data.

Moreover, our proposed retraining method can also improve the detection accuracy of the model over the clean data. We perform the defended network to predict the clean data in *RealWorld* and *Product*, respectively. Further investigation shows that our retrained model can achieve classification accuracy over 87.2% and 92.27%, which are improving more than 5% on average compares to the original model (the classification accuracy of the original model is 82.85% and 85.68%, as shown in Index 11 and Index 12 of Table 1).

4 Related Work

In this section, we list several related works with TranFuzz, specifically, attack methods and adversarial defenses, and DNN-based model fuzzing techniques.

4.1 Adversarial Attacks and Defenses

Black-Box Attacks. The transfer-based strategy is an extremely important black-box attack method, and several types of research [3,4] were proposed based on the transfer attack method. Su et al. [21] analyzed an attack situation under extreme conditions and proposed an adversarial perturbation based on differential evolution to perform a single-pixel attack. The results of the experiment show that the reported method can modify the output of the model and only change one pixel of the image. In addition, ZOO [15] is a query-based black box attack method, and they exploit the non-derivative optimization strategy and symmetrical injury difference to estimate the Hessian gradient matrix. The method does not need to obtain the gradient information of the target model. Engstrom et al. [22] proposed an attack method based on spatial transformation, which makes it possible to study the vulnerability of neural network classifiers by carrying out image rotation and translation operations.

Adversarial Training. Adversarial training is a data enhancement strategy to improve the robustness of the model. Madry et al. [4] proposed a min-max optimization framework using the projected gradient descent method to generate conflicting samples as augmentation data. This method first finds several examples by adopting the PGD and then uses these examples as the adversarial training data to decrease the training loss. Wong et al. [25] adopt a weaker, lower-cost adversarial strategy to form a robust model. The method combines the fast gradient sign and random initialization methods in adversarial training. The results of the experiment have shown that it has effective performance with lower cost compared to the PGD-based adversarial training method.

4.2 DNN Model Fuzzers

DeepXplore [2] is a fuzzing-based method to verify the DNN system. The method first proposed Neuron Coverage as a coverage metric for guiding the DNN model's testing. DeepXplore uses differential testing and generates some test inputs to identify the incorrect behavior of the deep learning system without the necessary manual operations. In addition, to effectively mutate test inputs, [13] proposed eight mutation strategies that include neuronal network weight-based mutation, neuron-based mutation, and layer-based mutation.

5 Conclusion

In this paper, we fully consider the relationship between the challenges of data black-box. Based on that, we proposed a non-targeted black-box attack framework. The evaluation experiment results show that our proposed framework can address both the non-robustness and robustness black-box attack tasks. In addition, TranFuzz can perform over 68% attack success rate against real-world Cloud Vision Services. Moreover, by taking advantage of the adversarial training strategy with data augmentation, TranFuzz can also strengthen the robustness of the original model.

Acknowledgment. This research was supported by National Natural Science Foundation of China (No. 62002203), Major Scientific and Technological Innovation Projects of Shandong Province, China (No. 2018CXGC0708, No. 2019JZZY010132), Shandong Provincial Natural Science Foundation (No. ZR2020MF055, No. ZR2020LZH002, No. ZR2020QF045), The Fundamental Research Funds of Shandong University (No. 2019GN095), The Open Project of Key Laboratory of Network Assessment Technology, Institute of information engineering, Chinese Academy of Sciences (No. KFKT2019-002).

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016, pp. 770–778 (2016)
2. Pei, K., Cao, Y., Yang, J., Jana, S.: DeepXplore: automated Whitebox testing of deep learning systems. In: SOSP 2017, pp. 1–18 (2017)
3. Xie, C., et al.: Improving transferability of adversarial examples with input diversity. In: CVPR 2019, pp. 2730–2739 (2019)
4. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. CoRR abs/1706.06083 (2017)
5. Bhagoji, A.N., He, W., Li, B., Song, D.: Exploring the space of black-box attacks on deep neural networks. In: European Conference on Computer Vision (2019)
6. Suya, F., Chi, J., Evans, D., Tian, Y.: Hybrid batch attacks: finding black-box adversarial examples with limited queries. In: USENIX Security Symposium 2020, pp. 1327–1344 (2020)
7. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)

8. Wang, M., Deng, W.: Deep visual domain adaptation: a survey. Neurocomputing **312**, 135–153 (2018)
9. Zhu, Y., Zhuang, F., Wang, J., et al.: Deep subdomain adaptation network for image classification. IEEE Trans. Neural Netw. Learn. Syst. **32**, 1713–1722 (2020)
10. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML 2015, pp. 97–105 (2015)
11. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
12. Xiao, Q., Chen, Y., Shen, C., Chen, Y., Li, K.: Seeing is not believing: camouflage attacks on image scaling algorithms. In: USENIX Security Symposium 2019, pp. 443–460 (2019)
13. Hu, Q., Ma, L., Xie, X., Yu, B., Liu, Y., Zhao, J.: DeepMutation++: a mutation testing framework for deep learning systems. In: ASE 2019, pp. 1158–1161 (2019)
14. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In: CVPR 2019, pp. 4322–4330 (2019)
15. Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.-J.: ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: AISec@CCS 2017, pp. 15–26 (2017)
16. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
17. Szegedy, C., et al.: Intriguing properties of neural networks. In: ICLR (Poster) (2014)
18. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy, pp. 39–57 (2017)
19. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR 2017, pp. 5385–5394 (2017)
20. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015)
21. Jiawei, S., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. **23**(5), 828–841 (2019)
22. Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A.: Exploring the landscape of spatial robustness. In: ICML 2019, pp. 1802–1811 (2019)
23. https://github.com/BorealisAI/advertorch
24. https://github.com/Trusted-AI/adversarial-robustness-toolbox
25. Wong, E., Rice, L., Zico Kolter, J.: Fast is better than free: revisiting adversarial training. In: ICLR 2020 (2020)