Miao *et al. Journal of Cloud Computing* (2024) 13:61
https://doi.org/10.1186/s13677-024-00617-9

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH **Open Access**

A secure data interaction method based on edge computing

Weiwei Miao[1], Yuanyi Xia[1*], Rui Zhang[1], Xinjian Zhao[1], Qianmu Li[2], Tao Wang[2] and Shunmei Meng[2]

Abstract

Deep learning achieves an outstanding success in the edge scene due to the appearance of lightweight neural network. However, a number of works show that these networks are vulnerable for adversarial examples, bringing security risks. The classical adversarial detection methods are used in white-box setting and show weak performances in black-box setting, like the edge scene. Inspired by the experimental results that different models give various predictions for the same adversarial example with a high probability, we propose a novel adversarial detection method called Ensemble-model Adversarial Detection Method (EADM). EADM defenses the prospective adversarial attack on edge devices by cloud monitoring, which deploys ensemble-model in the cloud and give the most possible label for each input copy received in the edge. The comparison experiment in the assumed edge scene with baseline methods demonstrates the effect of EADM, with a higher defense success rate and a lower false positive rate by an ensemble-model consisted of five pretrained models. The additional ablation experiment explores the influence of different model combinations and adversarial trained models. Besides, the possibility about transfering our method to other fields is discussed, showing the transferability of our method across domains.

Keywords Deep learning, Adversarial detect, Edge scene

Introduction

Deep learning achieves outstanding success in several fields due to increasing data quantity and quality in recent years [10, 11, 18, 27, 28, 45, 55]. It plays a role in multiple applications [40, 41, 49, 52, 56], e.g., autonomous driving [31], facial recognition [36] and fingerprint payment. The breakthroughs in hardware and algorithms facilitate the deployment of deep learning models on edge devices which own weak computing resource, accelerating the birth of edge computing [1, 2, 25, 32, 51]. One of the most critical deep learning technologies applying in edge computing is the appearance of lightweight deep neural network that contains a small number of parameters and has the close performance to big model in most tasks, e.g., MobieNet designed by Google. There are also some applications of deep learning in the edge computing. In the Computer Vision (CV), e.g., Vigil is a successful camera system which are deployed in the edge and has powerful functions such as searching for designated personnel intelligently. In the Natural Language Processing (NLP), a famous application is Siri, the voice assistant designed by Apple. The wake word is recognized by two deep learning networks. In the other hand, with the development of cloud service, it is common that users who need massive computing resource obtain deep learning services from the cloud, known as cloud computing [48].

However, a number of works demonstrate that deep neural networks are vulnerable for adversarial examples [9, 16, 22, 33, 47, 54], which are crafted by adding the elaborate and imperceptible noise on benign images. The existence of adversarial examples reveal defects of neural networks, challenging the

*Correspondence:
Yuanyi Xia
xiayuanyi@yeah.net
[1] State Grid Jiangsu Electric Power Co., Ltd. Information & Telecommunication Branch, Nanjing, China
[2] Nanjing University of Science and Technology, Nanjing, China

trustworthiness and robustness of deep learning models. There is a more worrisome thing that the adversarial examples crafted on source model can transfer to other models [29, 42, 44], known as transferability.

With the discovered transferability of adversarial examples, some dangerous safety issues occur under the edge scene [17, 20, 23]. For examples, due to the widespread use of MobileNet on edge devices, attackers conduct malicious attack by swapping adversarial examples on pretrained MobileNet model which is released on the Internet. This lightweight model has a weak robustness and it is easy to attack such deep learning model successfully. Some facial recognition systems work based on neural networks and attackers can wear carefully designed hats or glasses which can mislead the systems to recognize them as designated individuals, causing property security issues.

Thus, it is necessary and meaningful to design a adversarial detection strategy which is suitable for the edge to protect the safety of the edge devices. To our best knowledge, the classical adversarial detection strategies that are proposed can be divided into three categories. The methods based on distribution statistics utilize the property that adversarial examples have the different digital features with benign images. The methods based on feature learning remove unnecessary features from the input to reduce the probability of adversarial samples without compromising the accuracy of the classifier. The methods based on intermediate output train a specific detector with its input as the intermediate output of the input data. However, some detection methods are just tested on the white-box attack setting while has weak performance on the black-box setting such as the edge scene.

Based on the phenomenon that it is challenging for attackers to achieve targeted attacks that require attackers to mislead the victim model to the specify category, attackers are more likely to conduct no-targeted attacks if they need to ensure a high attack success rate. We argue that different models might output the diverse results for the most adversarial examples generated in untargeted attack methods and propose a novel adversarial detection method called Ensemble-model Adversarial Detection Method (EADM), which can significantly assist the edge devices in defending against adversarial attacks. The core theory of EADM is the robustness of different models, which means different models will give different results on adversarial examples. EADM first deploys multiple models in the cloud to give the most possible classification results for the input image. Then, the adversarial example can be identified if the classification results are not same between the lightweight model in the edge and the ensemble-model in the cloud. The framework of EADM is shown in Fig 1. Compared with the classical adversarial defense methods, our EADM has better performance than these methods in the edge scenarios, which can be demonstrated by the experimental results. In the other hand, EADM has no threshold to set while classical methods need to confirm it by conducting lots of experiments.

In the other hand, we discovered that there are still a small number of adversarial examples can mislead all test models to the same wrong classification, which may make our method ineffective. This is because the similarity of different model decision boundaries and poor robustness of these models. To help improve the model robustness, the adversarial training method is usually used [9]. By adding the adversarial examples to the train dataset, the models have stronger ability to defense adversarial

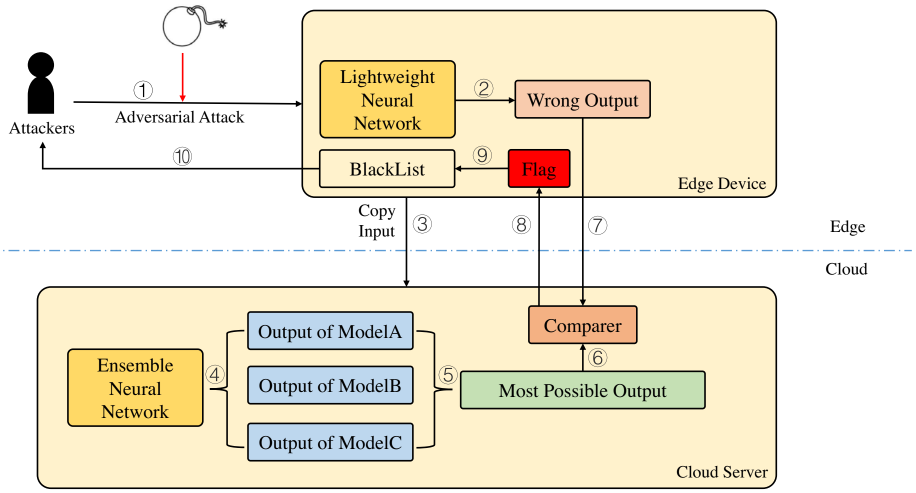

Fig. 1 Illustration of the framework which uses Ensemble-model Adversarial Detection Method (EADM)

attacks. Thus, the models can be trained for different parameters through adversarial training.

Finally, we discuss the realizability of transfering our EADM to other field briefly, such as from CV to NLP. The difference between CV and NLP is that the processed data are changed to texts and the classifier is the emotion classification model in NLP. There are also several adversarial attack methods which can apply in the text. Our EADM is still workable and detects adversarial examples by prediction difference.

In conclusion, our contributions are summarized as follows:

- We define an adversarial attack setting in the edge scene where the substitute model is the pretrained MobileNet and attackers use black-box adversarial attack algorithm.
- We study the outputs of the same adversarial examples in different models and propose an adversarial detection method, namely Ensemble-model Adversarial Detection Method (EADM), which utilizes several models to give predictions and determine whether image is an adversarial example through prediction difference.
- We conduct a comparison experiment on ImageNet dataset, using two classical adversarial detection methods. The experimental results demonstrate the effectiveness of our proposed method. Compared with the baseline methods, EADM has a lower false positive rate and a higher defense success rate.
- We conduct ablation experiments to explore the effect of different model combinations and test the influence of models which are adversarial trained. We prove that the adversarial train method can address the disadvantages of pretrained models and improve the performance of EADM significantly.
- We discuss the possibility about transfering our EADM to other fields. We hope that the idea of our adversarial examples detection method can help deep learning models defense adversarial examples in more fields.

Related work

Deep learning in edge computing

For overcoming the challenges when deploying DNNs on edge devices, such as insufficient computing resource or memory space inadequate, the lightweight deep neural network named MobileNet is designed by Google and improved for several times, i.g. MobileNetV1, MobileNetV2 and MobileNetV3.

MobileNetV1 [14] is based on depthwise separable convolutions which is a form of factorized convolutions that factorize a standard convolution into a depthwise convolution and a 1 × 1 convolution called a pointwise convolution. Besides, MobileNetV1 introduces two hype-parameters to build the models that has smaller capacity and less latency in certain tasks, where the first hype-parameter width multiplier α aim to make models thinner and the second hype-parameter resolution multiplier ρ can reduce representation. The experimental results show that the accuracy rate of MobileNetV1 on ImageNet is just 0.9% lower than VGG16 but the size of model parameters is 1/32 of VGG16.

There are still some disadvantages about MobileNetV1. Firstly, it use the simple structure like VGG net which has low cost performance. Secondly, its depthwise convolution usually causes activation of 0 and lead to the problem of failed training with ReLu activation function. Inspired by ResNet, MobileNetV2 [35] introduces inverted residual to improve the architecture. The first 1 × 1 convolution layer of inverted residual is aim to expand channels and the last 1 × 1 convolution layer is designed to reduce channels. The middle 3 × 3 convolution layer is changed to depthwise separable convolution with ReLu6 activation function. Besides, The last 1 × 1 convolution layer replace ReLu function with the linear activation function, which called linear bottleneck. This is because ReLU will destroy low dimensional features and the output of last 1 × 1 convolution layer is just low dimensional.

In order to achieve the more accuracy and lower latency, two novel networks based on the improvement of MobileNetV2 are proposed, which called MobileNetV3-Large and MobileNetV3-Small [15]. The difference between these two models is the parameter size. Users can choose the suitable model according to their needs and resource. MobileNetV3 model has several modifications based on its predecessor. First, MobileNetV3 add the Squeeze-and-Excitation(SE) module which has significant performance improvements with minimal computational costs. Second, MobileNetV3 modifies sigmoid function and swish activation function to h-sigmoid and h-swich which accelerate forward propagation and derivation. Third, MobileNetV3 redesigns the layers that cost time by reducing the number of convolutions and simplify last stage.

Adversarial attack in the edge

According to the knowledge attackers own about the victim model, the adversarial attacks can be divided into two categories: white-box attacks and black-box attacks. When deep learning networks are deployed on the edge devices, it is a setting like white-box, where attackers have the whole knowledge about the deployed models. However, in most scenes, attackers have no access to obtain the model information, known as black-box setting. To conduct black-box attack, attackers can utilize the transferability of adversarial examples, generating adversarial examples from a local surrogate model.

The adversarial attack settings under edge scene are that attackers obtain the model on the edge device and conduct white-box attack or the attackers use the black-box attack methods, crafting adversarial examples on the lightweight model deployed on edge devices and attack the models in the edge. Below are some adversarial attack methods that can be applied under the edge scene.

Fast Gradient Sign Method (FGSM) [9] is the first white-box attack method which utilizes the gradient information to generate adversarial examples. FGSM obtains the gradient of the input image by back propagation algorithm and makes noise which has the same direction with computed gradient. The operation about disturbance addition is only executed once which may lead to poor attack performance. Iterative Fast Gradient Sign Method (I-FGSM) [21] is an iterative version of FGSM. It decomposes the total noise into multiple small noise and add divided noise to image in each iteration.

Although the white-box attack performance of I-FGSM is outstanding with nearly 100 % attack success rate, when the white-box adversarial examples are tested on other networks with different architecture and parameters, they achieve the poor black-box attack performance. Researchers infer that the main reason for such phenomenon is that the white-box adversarial examples are overfitting for the source model and it is necessary to alleviate the overfitting problem and improve transferability when conducting black-box attack, e.g., attacking the cloud models.

Lots of methods are proposed to boost the transferability in black-box attacks. Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [4] is one of the most classical attack methods which introduce the momentum to optimize the search process of adversarial examples, escaping from poor local minima. MI-FGSM can be also integrated with other black-box attack methods, achieving a higher attack success rate.

Besides the momentum-based methods, some attacks improve the transferability by input transformation. Diverse Input Method (DIM) [46] is the first method which utilize input transformation to improve transferability. In each iteration, DIM has a probability to conduct input transformation on the input image.The input transformation first resizes the input image to a random size then expand it to a preset size by randomly filling pixels. Translation-Invariant Method (TIM) [5] uses a set of translated images to optimize adversarial examples. In order to reduce computation complexity, TIM apply convolution kernel to convolve the gradient of the input image based on the translation-invariant property of DNNs. Scale-Invariant Method (SIM) [26] calculates the gradient by several scaled images. Because of the assumed scale-invariant property of DNNs, SIM uses scaling transformation which is a type of exponential scaling function to change input image. Admix [43] first randomly choose several images from other categories and mix input image with them by an addition strategy. Then Admix keeps the rule of SIM, uses the same scaling function to craft mixed images.

Adversarial detection method

According to the main idea applied in adversarial detection methods, the common detection methods can be divide into three categories, i.e. distribution statistics, feature learning and intermediate output.

The methods based on distribution statistics detect adversarial example by checking the distribution of output. Kullback-Leibler (KL) divergence [12] is usually used to measure the degree of dispersion between two probability distributions. The greater the difference in distribution, the greater the KL divergence is. In deep neural networks, there is a significant difference in the softmax output between adversarial example and clean image where the difference can be computed with KL divergence. This method is relatively easy to implement, but its effectiveness depends on the setting of hyper-parameter thresholds. There is also a high probability to a higher false positive rate.

The methods based on feature learning remove unnecessary features from the input to reduce the probability of adversarial samples without compromising the accuracy of the classifier. Principal Components Analysis (PCA) [12] method reduces the feature dimension of input image and learning the knowledge from low dimension data. The experimental results indicate that the adversarial example is different with origin image after dimension reduction so PCA can help detect adversarial examples. Feature Squeezing (FS) [50] method use the different features of adversarial examples and original image to detect adversarial examples. FS compares the difference between the input image and the feature compressed image, identifying the adversarial example when the difference is bigger or smaller than threshold. The common measures used in FS are color depth compression and feature smoothing. Color depth compression method works by an operation combination composed of multiplication, rounding and division. Feature smoothing method apply median filtering or other smoothing ways for input image. Although there is no difficulty to implement these methods, PCA and FS have the hyper-parameter thresholds that are difficult to set.

The methods based on intermediate output train a specific detector with its input as the intermediate output of the input data. Input refactoring method [12] obtain the intermediate output as the input of refactoring network. The clean image can be recovered easily but the image restored from the adversarial example will be irregular and fuzzy. Adversarial Detection Network [30] method treats adversarial detection tasks as a binary classification problem and train a model which is trained on a dataset consist of the intermediate outputs of clean images and adversarial examples. These methods rely on the network training and have significant differences in effectiveness, which is both an advantage and a disadvantage.

Methodology

In this section, we will first define the notations used for adversarial attacking in the edge scene. Then, we state our motivation from solving the existing questions by conducting experiments. Finally, we introduce our Ensemble-model Adversarial Detection Method (EADM) and provide an illustration and an algorithm for a better understanding for our proposed method.

Problem settings

In the edge scene, attackers are in the edge and the task of attackers is to mislead the deep neural network on the edge devices while the model is usually the MobileNet. There are two tasks for the defense system deployed in the edge while the first is to accept the clean or natural images and give the right predictions, which means a low positive rate, and the second task is to defense the adversarial examples, including closing the Application Programming Interface(API) for the users who input suspicious images when system detects the adversarial examples or giving the true model prediction for input, which means a failed attack for attackers. The six attack and defense situations that will happen in the edge scene are shown in Table 1.

Motivation

There are a large number of works which show that targeted attacks are more difficult to conduct than untargeted attacks in black-box setting where the targeted attacks require attackers to mislead the model prediction to a prescribed category and untargeted attacks just make deep learning model having a wrong classification. So attackers are more likely to choose untargeted approach when they have a plan to conduct adversarial attack.

However, there is a question: *whether the misclassification labels are the same when adversarial example attack successfully on two or more networks in untargeted attacks*? To solve this problem, we conduct adversarial attack experiment on MobileNetV3-Small (MobileNetV3) [15] with 1000 images. In the experiment, we use SIM [26] and Admix [43] to generate adversarial examples where they are the state of the art black-box attack methods and test their attack performance on the different pairwise combinations of five pretrained models, i.e. Inception-v3 (Inc-v3) [39], VGG16 [37], ResNet50 (Res50) [13], Inception-v4 (Inc-v4) and Inception-ResNet-v2 (IncRes-v2) [38]. Before this experiment, we compute the classification accuracy on each model first. Results are shown in Table 2.

The experimental results about attacking pairwise combinations of models are shown in Table 3. We can observe that the examples which models give different predictions account for the majority proportion of adversarial examples in untargered attack. The different attack methods and distinct model combinations have few influence on such tendency.

Another question is that *what may cause that multiple models give an unknown input image different predictions*? Thanks to above experimental results, excluding the reasons for the low classification accuracy of the model itself, we speculate that the most possible answer for the problem might be that the unknown input image is an adversarial example. Based on this assumption, we realize that we can utilize several models to detect adversarial example through collecting ensemble-model predictions and checking the difference. Figure 2 can help understand our motivation for the method that we propose next.

In the cloud, a large number of deep learning models can be deployed because of its abundant computing resource. Thus, the cloud can monitor the edge and help edge devices defense adversarial attacks.

Table 1 The situations that will happen on edge devices

Examples in the Edge	API on Edge Device	Model Prediction	Type
Clean	Open	True	Normal
	Open	False	Normal
	Close	None	False Positive
Adversarial	Open	True	Failed Attack
	Open	False	Successful Attack
	Close	None	Failed Attack

Table 2 The classification accuracy for clean images and adversarial examples generated by black-box attack methods on five pretrained models. The black-box attack are based on MI-FGSM

Attack	Inc-v3	VGG16	Res50	Inc-v4	IncRes-v2
Clean	98.9%	90.8%	95.6%	99.7%	99.0%
SIM	44.0%	21.9%	33.9%	46.6%	51.4%
Admix	38.0%	16.3%	29.7%	42.5%	48.4%

Table 3 The experimental results about attacking pairwise combinations of five pretrained models. The attacks are based on MI-FGSM. The value of adversarial examples is represents the number of images that are able to mislead the two models at the same time. The last two columns in the table represents the amount of images which two models give the same predictions and different predictions

Attack	Inc-v3	VGG16	Res50	Inc-v4	IncRes-v2	Adversarial Examples	Same Predictions	Different Predictions
SIM	✓	✓				516	131	385
	✓		✓			490	148	342
	✓			✓		447	154	293
	✓				✓	413	144	269
		✓	✓			620	183	437
		✓		✓		493	120	373
		✓			✓	457	110	347
			✓	✓		465	135	330
			✓		✓	434	132	302
				✓	✓	411	148	263
Admix	✓	✓				588	136	452
	✓		✓			548	162	386
	✓			✓		492	150	342
	✓				✓	460	163	297
		✓	✓			677	196	481
		✓		✓		553	125	428
		✓			✓	496	93	403
			✓	✓		519	143	376
			✓		✓	477	138	339
				✓	✓	450	156	294

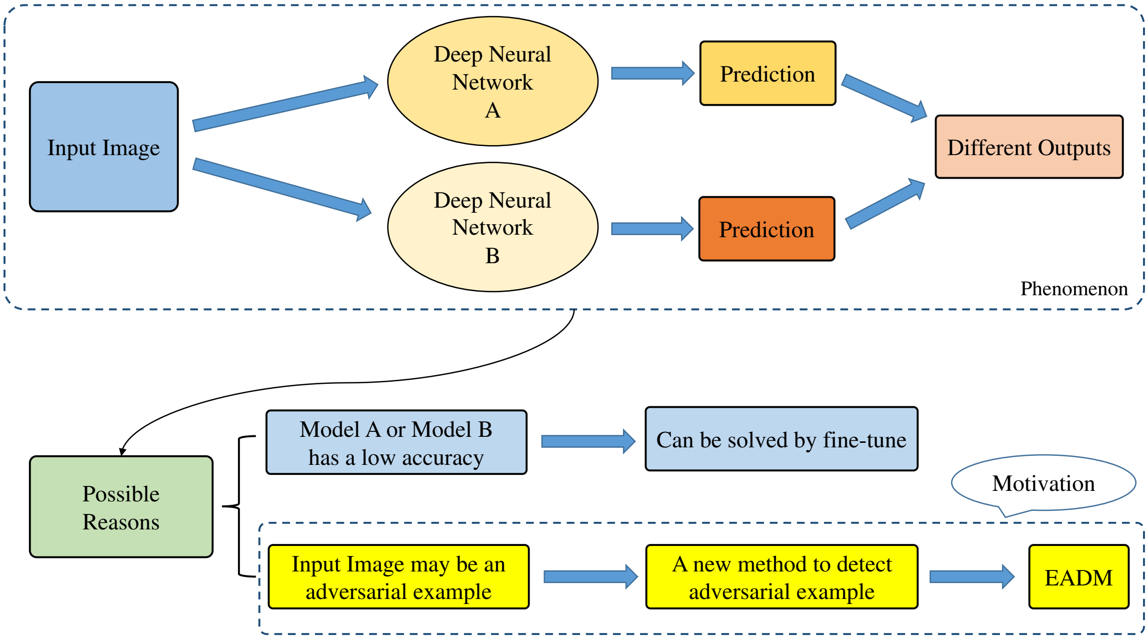
Input Image
Deep Neural Network A
Deep Neural Network B
Prediction
Prediction
Different Outputs
Phenomenon
Possible Reasons
Model A or Model B has a low accuracy
Can be solved by fine-tune
Motivation
Input Image may be an adversarial example
A new method to detect adversarial example
EADM

Fig. 2 Illustration of the phenomenon that we observe from the experiments and the motivation for our method

Ensemble-model adversarial detection method

To help prevent the edge devices from adversarial attacks, we propose a novel adversarial detection method called Ensemble-model Adversarial Detection Method (EADM), which works by deploying multiply models in the cloud and detects adversarial examples by prediction comparation. Considering EADM as a function F which returns whether the input is an adversarial example, the formulaic description of our proposed method is shown in Eq. (1):

$$F(x,f,g) = \begin{cases} True, & if\ f(x) \neq g(x) \\ False, & if\ f(x) = g(x), \end{cases} \tag{1}$$

where x is the input image while $f(x)$ is the output of the lightweight model f in the edge and $g(x)$ is the most possible label that is the result of the ensemble-model g in the cloud. The return value equals true means that EADM determines that the input is an adversarial example.

Then we give the detailed calculation way of $g(x)$. We define that g is an ensemble-model consisted of n models and g_i represents the model of number i. The value of $g(x)$ can be calculated as Eq.(2) when n is set to 3. When n is set a different number, $g(x)$ can be computed similarly.

$$g(x) = \begin{cases} g_1(x), & if\ g_1(x) = g_2(x)\ or\ g_1(x) = g_3(x) \\ g_2(x), & if\ g_1(x) \neq g_2(x)\ and\ g_2(x) = g_3(x) \\ g_1(x), & if\ g_1(x) \neq g_2(x)\ and\ g_2(x) \neq g_3(x)\ and\ g_1(x) \neq g_3(x) \end{cases} \tag{2}$$

Algorithm 1 Ensemble-model Adversarial Detection Method

```
Input: An input images $x$ without true label
Input: A lightweight classifier $f$ with parameter $\mu$
Input: A set $M$ of $K$ classifiers$\{g_1, ..., g_K\}$ with parameters $\{\theta_1, ..., \theta_K\}$
Output: $x$ is a clean image or $x$ is an adversarial example
$Pred = f(x, \mu)$
for $i = 1$ to $K$ do
    $pred_i = g_i(x, \theta_i)$
end for
Find the most possible label $pred$ that occurs most between $\{pred_1, ..., pred_K\}$
if $Pred == pred$ then
    return $x$ is a clean image
else
    return $x$ is an adversarial example
end if
```

The general description of EADM is summarized in Algorithm 1. The EADM algorithm needs a set of classifiers and their corresponding model parameters. The tested input is an unknown image. For each input image, the lightweight classifier output its result while other classifiers give their predictions. Then the most possible label is computed. If the situation that there is a different prediction happens, the input image is masked as an adversarial example.

For helping understand the workflow of EADM in practical edge scenarios, the structure of our method is exhibited in Fig. 3. EADM deploys multiply deep neural models in the cloud. When the edge device received an image from an user, the lightweight deep neural network on the edge device gives its prediction as p and sends a copy of input image to the cloud at the same time. Then all models in the cloud give their predictions. Later, EADM collect the model predictions and compute the most possible label. If p is the same as most possible label, EADM believe it a clean image and give the model output, otherwise, EADM regard input as an adversarial example and the edge marks the user as an attacker.

Experiments

In this section, we conduct experiments to verify the effectiveness of our proposed method in an assumed edge scene. First, we specify the setup of experiments. Then we report the results about the baseline methods and our EADM. Finally, we do ablation study to explore the the role of different model combinations and adversarial trained models.

Experiment setting

Dataset. The attackers in the edge choose 1000 images from ILSVRC 2012 validation set [34] to generate adversarial examples, which are provided by Lin et al. [26]. Attackers send the 1000 clean images and 1000 adversarial examples to the edge devices.

Baselines. We choose two detection methods to our baseline methods, i.e. Kullback-Leibler (KL) method and Feature Squeezing (FS) method.

Models. The model in the edge is fine-tuned MobileNetV3-Small (MobileNetV3) [15], The ensemble-model in the cloud are consisted of five pretrained models, i.e. Inception-v3 (Inc-v3) [39], VGG16 [37], ResNet50 (Res50) [13], Inception-v4 (Inc-v4) and Inception-ResNet-v2 (IncRes-v2) [38]. We use SIM [26] and Admix [43] to generate adversarial examples on pretrained MobileNetV3. All these models can be found in[1] and[2].

Metric. There are two metrics to help evaluate the performance of methods. For clean images, the false positive (FP) rate is to estimate how much the method mistakes for adversarial examples. The lower FP is better. For

[1] https://github.com/Cadene/pretrained-models.pytorch

[2] https://github.com/pytorch/vision/tree/main/torchvision

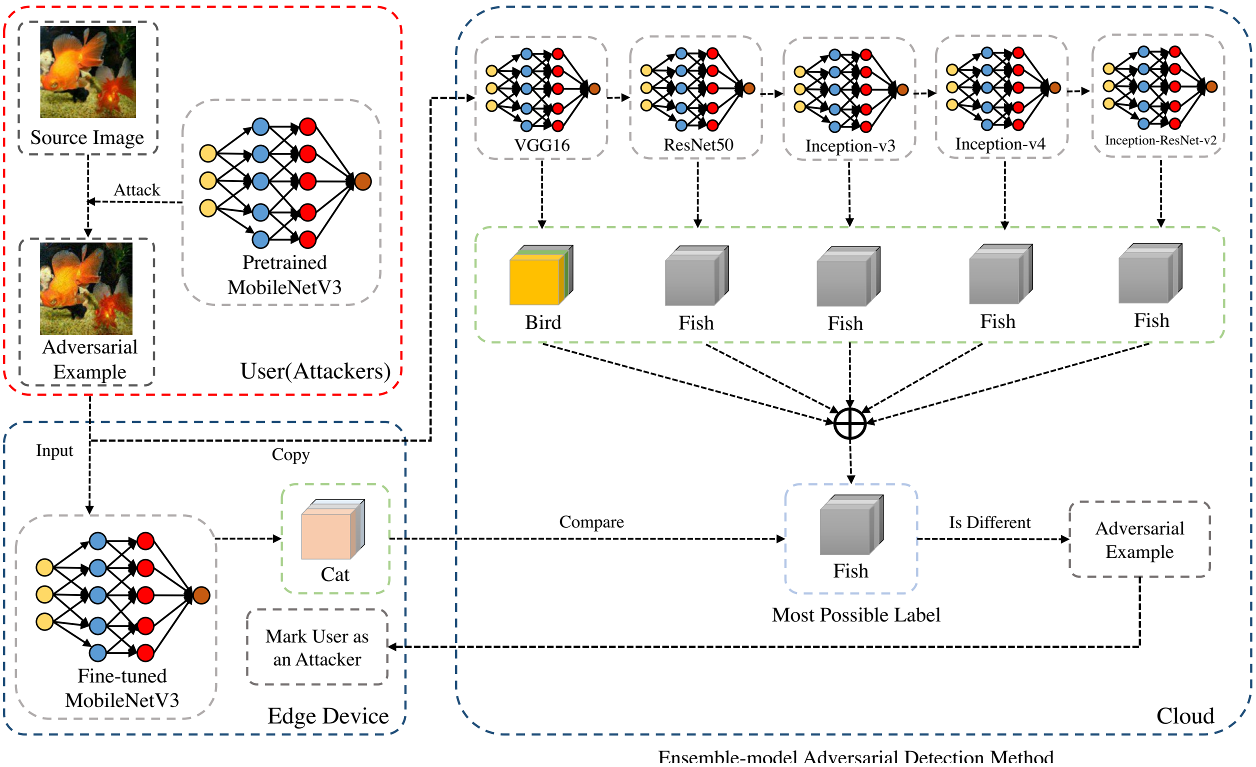

Fig. 3 Illustration of Ensemble-model Adversarial Detection Method (EADM)

adversarial examples, the defense success rate (DSR) is to evaluate the effect of methods. The higher DSR is better.

Experiment result

The experimental results are shown in Tables 4 and 5. We find that our proposed EADM has the highest defense success rate in the all three methods. Besides, EADM has a low false positive rate, which means it may not mislead the clean images to adversarial examples. The two baseline methods have the poor performance in the experiment. We guess that there are two reasons for this phenomenon. The first reason is that the model is fine-tuned which means a lower generalization while the second reason is that the test is on the black-box setting and such methods are invalid in this setting.

Table 4 The false positive rate (FP) and the defense success rate (DSR) against the adversarial examples crated on MobileNetV3 by SIM attack method. The attacks are based on MI-FGSM

Method	Threshold	FP	DSR
KL	5.204	1.0%	0.1%
	7.326	5.0%	1.4%
	8.124	10.0%	2.3%
	9.180	20.0%	3.7%
FS	14.614	1.0%	1.7%
	11.266	5.0%	5.7%
	10.254	10.0%	8.7%
	8.333	20.0%	19.9%
EADM (ours)	\	0.2%	91.4%

Table 5 The false positive rate (FP) and the defense success rate (DSR) against the adversarial examples crated on MobileNetV3 by Admix attack method. The attacks are based on MI-FGSM

Method	Threshold	FP	DSR
KL	5.204	1.0%	1.6%
	7.326	5.0%	4.7%
	8.124	10.0%	7.1%
	9.180	20.0%	10.2%
FS	14.614	1.0%	0.3%
	11.266	5.0%	1.6%
	10.254	10.0%	3.3%
	8.333	20.0%	8.6%
EADM (ours)	\	0.2%	87.9%

Another observed phenomenon is that when implementing more aggressive adversarial attacks, namely Admix, KL performs better while FS and our method have lower success rates in defense. For KL, we hold a viewpoint that the adversarial examples generated by

Admix will lead to larger differences in KL divergence, making it easier to identify. For FS, we assume that Admix disrupts higher dimensional features of the image, which FS cannot find. For our method, we believe that the majority of models in the ensemble-model give consistent erroneous results for the adversarial examples generated by Admix, which happens to be the same as the network in the edge.

We also observed that EADM can not recognize all adversarial examples successfully, thus we guess that higher success rates in adversarial defense can be achieved through adversarial training or removing the less robust models from the ensemble-model.

Ablation study

Adversarial trained models vs. Pretrained models. The adversarial train method is usually used to improve the robustness of deep neural model. We retrain the five pretrained models on a dataset consisted of original images and the adversarial examples that are crated on MobileNetV3 by MI-FGSM.

The results are shown in Figs. 4 and 5. Compared with the pretrained models, the ensemble-model consisted of adversarial trained models classify the clean images almost correctly, which means a very low false positive rate. Because of the improvement of robustness, the defense success rate is higher than the pretrained models.

The number of adversarial trained models in ensemble-model. Because of the high cost of adversarial training, it is necessary to study the balance between the number of pretrained models and the number of adversarial trained models. We first replace one pretrained model to adversarial model and then increase the number of adversarial models till five. The order of replacing models is based on the accuracy for Admix method, from low to high, shown in Table 2. from low to high. The results are shown in Figs. 6 and 7. Along with the increase of the number of adversarial trained models in ensemble-model, the defense success rate becomes higher. However, we need to make trade-offs because we find that DSR has a lower improvement when the number of adversarial trained models is a big value.

The transferability of Ensemble-model Adversarial Detection Method

In the prior context, we introduce the detail of our EADM. However, the data are just images, which means EADM is limited to Computer Vision (CV). In Natural Language Processing (NLP), when the data are texts, there is a need to make some adjustments to EADM. Table 6 shows the difference of EADM between CV and NLP. If researchers are familiar with CV and NLP at the same time and focus on the adversarial attack and defense, they are able to deploy our EADM in NLP with a low cost easily.

To better demonstrate the transferability of EADM, we give an example about the application of our method in electricity scenario, which is a common scene in edge

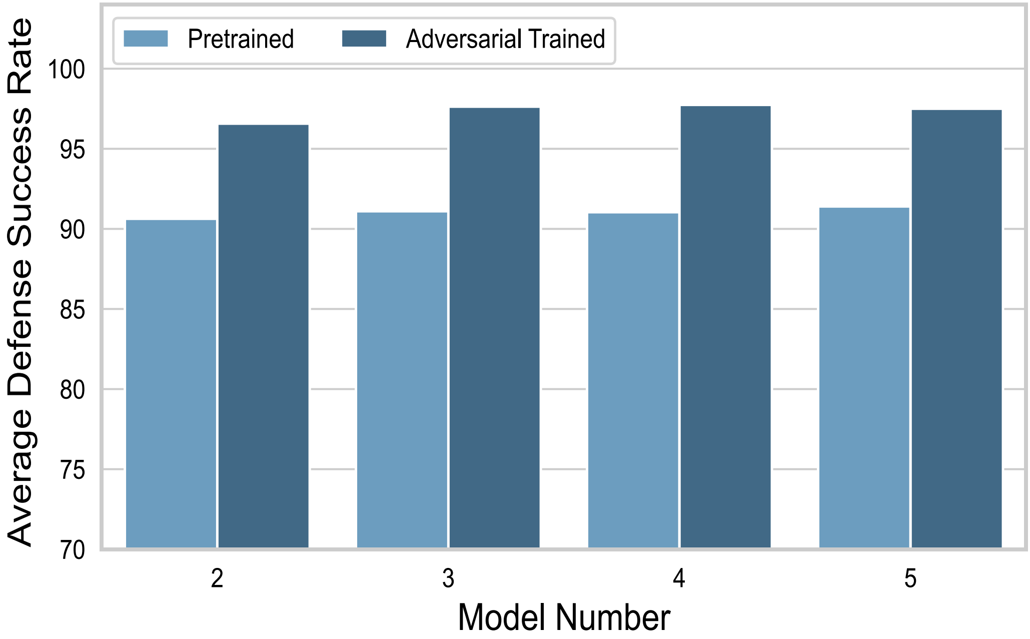

Fig. 4 The average defense success rate (ADSR) of various model combinations against the adversarial examples crated on MobileNetV3 by SIM attack method

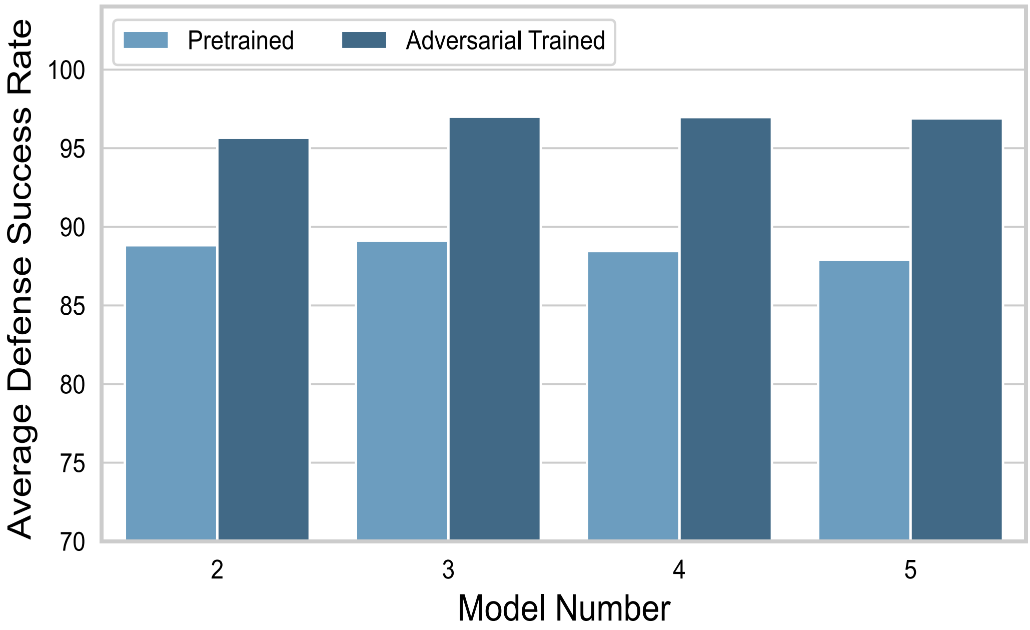

Fig. 5 The average defense success rate (ADSR) of various model combinations against the adversarial examples crated on MobileNetV3 by Admix attack method

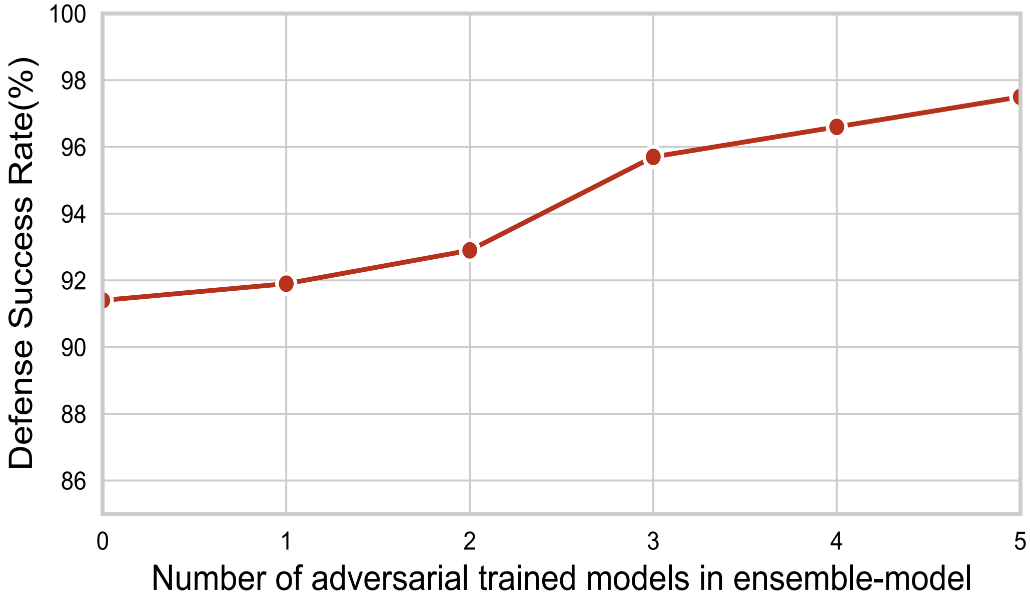

Fig. 6 The defense success rate (DSR) of number of adversarial trained models in ensemble-model against the adversarial examples crated on MobileNetV3 by SIM attack method

computing. In such scene, the data is control instruction that belongs to text. The control instruction will be modified maliciously by attackers who use attack methods such as TextFool. For this falsified instruction, the model on the edge device gives an incorrect classification. Firstly, the edge copies the control instruction and sends to the cloud. Secondly, the big model in the cloud gives the most possible label for the sent instruction. Finally, the attack is detected if there is a difference between the output in the edge and the most possible label. The edge is able to implement the corresponding defense strategy according to the result of detection.

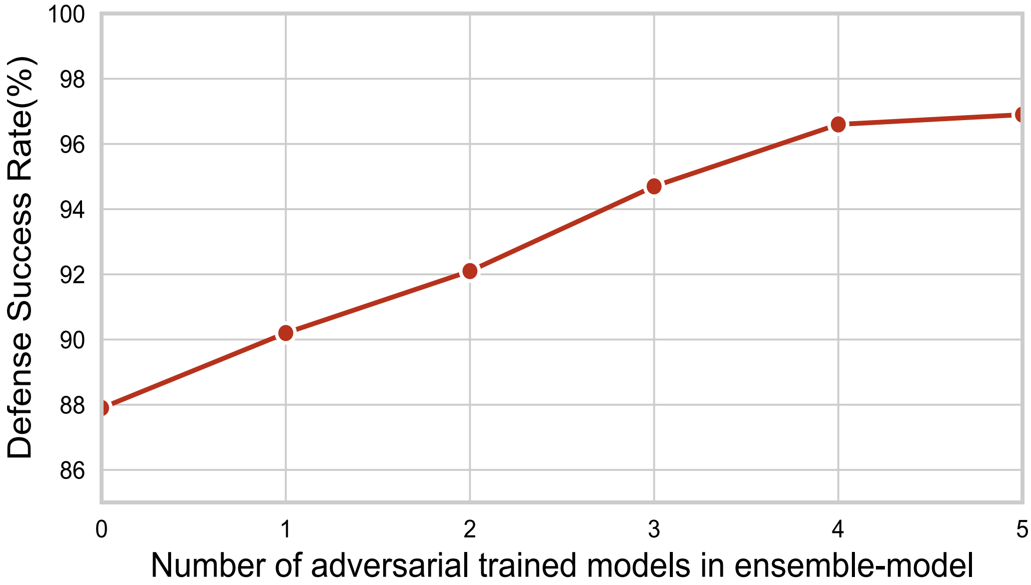

Fig. 7 The defense success rate (DSR) of number of adversarial trained models in ensemble-model against the adversarial examples crated on MobileNetV3 by Admix attack method

Table 6 The difference of EADM between CV and NLP

Type	CV	NLP
Data	Image	Text
Model	Convolutional Neural Network [37] Vision Transformer [6]	Recurrent Neural Network [53] BERT [3]
Attack Method	FGSM [9], MI-FGSM [4] SIM [26], Admix [43]	TextFool [24], HotFlip [7] DeepWordBug [8], TextFooler [19]

Conclusion

In this paper, we propose a novel adversarial detection method, namely Ensemble-model Adversarial Detection Method (EADM). EADM helps the edge devices defense the adversarial attacks with the cloud monitoring, working by the ensemble-model giving the most possible label for the input that received on edge devices. The adversarial example is detected when the edge output is different from the most possible label. The results of the comparison experiment demonstrate the effectiveness of our EADM. We also introduce the way to transfer our method into other fields, such as NLP. However, we think that our proposed EADM may be limited by the the model with the worst robustness in the ensemble-model and we suggest that using more advanced models such as Vision Transformer in ensemble-model to enhance the stability of EADM. Researchers can focus on exploring the more suitable model combinations to get the best performance of EADM. We hope that our adversarial detection method can be applied in the edge scenes that using neural networks in industry. In the future, we attempt to study more feasible methods to detect and defense adversarial attacks happened on the edge devices.

Acknowledgements
This work was supported by the Science and Technology Project of State Grid Jiangsu Electric Power Company Ltd., under Grant J2023124.

Authors' contributions
Miao gave the idea of EADM first and conducted the main experiments. Xia and Zhang helped finish experiments. Miao, Xia, Zhang, Zhao and Li wrote the main manuscript. Wang and Meng prepared the figures and tables. All authors reviewed the manuscript.

Funding
Science and Technology Project of State Grid Jiangsu Electric Power Company Ltd., under Grant J2023124.

Availability of data and materials
All the data are available through the corresponding author.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 26 December 2023 Accepted: 24 February 2024
Published online: 18 March 2024

References

1. Dai H, Xu Y, Chen G et al (2020) Rose: Robustly safe charging for wireless power transfer. IEEE Trans Mob Comput 21(6):2180–2197
2. Dai H, Wang X, Lin X et al (2021) Placing wireless chargers with limited mobility. IEEE Trans Mob Comput. https://doi.org/10.1109/infocom41043.2020.9155356
3. Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint https://doi.org/10.48550/arXiv.1810.04805
4. Dong Y, Liao F, Pang T, et al (2018) Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9185–9193
5. Dong Y, Pang T, Su H, et al (2019) Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4312–4321
6. Dosovitskiy A, Beyer L, Kolesnikov A, et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint https://doi.org/10.48550/arXiv.2010.11929
7. Ebrahimi J, Rao A, Lowd D, et al (2017) Hotflip: white-box adversarial examples for text classification. arXiv preprint https://doi.org/10.48550/arXiv.1712.06751
8. Gao J, Lanchantin J, Soffa ML, et al (2018) Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, pp 50–56
9. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint https://doi.org/10.48550/arXiv.1412.6572
10. Gu R, Chen Y, Liu S et al (2021) Liquid: intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed gpu clusters. IEEE Trans Parallel Distrib Syst 33(11):2808–2820
11. Gu R, Zhang K, Xu Z, et al (2022) Fluid: dataset abstraction and elastic acceleration for cloud-native deep learning training jobs. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, pp 2182–2195
12. Hendrycks D, Gimpel K (2016) Early methods for detecting adversarial images. arXiv preprint https://doi.org/10.48550/arXiv.1608.00530
13. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
14. Howard AG, Zhu M, Chen B, et al (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint https://doi.org/10.48550/arXiv.1704.04861
15. Howard A, Sandler M, Chu G, et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
16. Huang Q, Katsman I, He H, et al (2019) Enhancing adversarial example transferability with an intermediate level attack. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4733–4742
17. Jiang R, Kang Y, Liu Y et al (2022) A trust transitivity model of small and medium-sized manufacturing enterprises under blockchain-based supply chain finance. Int J Prod Econ 247(108):469
18. Jiang R, Han S, Yu Y et al (2023) An access control model for medical big data based on clustering and risk. Inf Sci 621:691–707
19. Jin D, Jin Z, Zhou JT et al (2020) Is bert really robust? a strong baseline for natural language attack on text classification and entailment. Proceedings of the AAAI conference on artificial intelligence 34:8018–8025
20. Kong L, Wang L, Gong W, et al (2021) Lsh-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web 1–16
21. Kurakin A, Goodfellow I, Bengio S (2016) Adversarial machine learning at scale. arXiv preprint https://doi.org/10.48550/arXiv.1611.01236
22. Li Y, Bai S, Zhou Y et al (2020) Learning transferable adversarial examples via ghost networks. Proceedings of the AAAI Conference on Artificial Intelligence, vol 34:07. pp 11458–11465
23. Li Z, Xu X, Hang T et al (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/tcss.2022.3217790
24. Liang B, Li H, Su M, et al (2017) Deep text classification can be fooled. arXiv preprint https://doi.org/10.48550/arXiv.1704.08006
25. Ling Z, Yu K, Zhang Y et al (2022) Causal learner: A toolbox for causal structure and markov blanket learning. Pattern Recogn Lett 163:92–95
26. Lin J, Song C, He K, et al (2019) Nesterov accelerated gradient and scale invariance for adversarial attacks. arXiv preprint https://doi.org/10.48550/arXiv.1908.06281
27. Liu H, Shen S, Khan AA et al (2023) Microservice-driven privacy-aware cross-platform social relationship prediction based on sequential information. Softw Pract Experience. https://doi.org/10.1002/spe.3240
28. Liu H, Li N, Kou H, et al (2023a) Fdrp: federated deep relationship prediction with sequential information. Wirel Netw 1–23
29. Long Y, Zhang Q, Zeng B, et al (2022) Frequency domain model augmentation for adversarial attack. In: European Conference on Computer Vision, Springer, pp 549–566
30. Metzen JH, Genewein T, Fischer V, et al (2017) On detecting adversarial perturbations. arXiv preprint https://doi.org/10.48550/arXiv.1702.04267
31. Pouyanfar S, Sadiq S, Yan Y et al (2018) A survey on deep learning: Algorithms, techniques, and applications. ACM Comput Surv (CSUR) 51(5):1–36
32. Qi L, Xu X, Wu X et al (2023) Digital-twin-enabled 6g mobile network video streaming using mobile crowdsourcing. IEEE J Sel Areas Commun. https://doi.org/10.1109/jsac.2023.3310077
33. Qin Z, Fan Y, Liu Y et al (2022) Boosting the transferability of adversarial attacks with reverse adversarial perturbation. Adv Neural Inf Process Syst 35:29845–29858
34. Russakovsky O, Deng J, Su H et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115:211–252
35. Sandler M, Howard A, Zhu M, et al (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Salt Lake City, pp 4510–4520
36. Sharif M, Bhagavatula S, Bauer L, et al (2016) Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 acm sigsac conference on computer and communications security, pp 1528–1540
37. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint https://doi.org/10.48550/arXiv.1409.1556
38. Szegedy C, Ioffe S, Vanhoucke V, et al (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 31. San Francisco
39. Szegedy C, Vanhoucke V, Ioffe S, et al (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, pp 2818–2826
40. Wang F, Zhu H, Srivastava G et al (2021) Robust collaborative filtering recommendation with user-item-trust records. IEEE Trans Comput Soc Syst 9(4):986–996
41. Wang F, Li G, Wang Y et al (2023) Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city. ACM Trans Internet Technol 23(3):1–19
42. Wang X, He K (2021) Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1924–1933
43. Wang X, He X, Wang J, et al (2021) Admix: Enhancing the transferability of adversarial attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 16,158–16,167

44. Wang X, Zhang Z, Zhang J (2023) Structure invariant transformation for better adversarial transferability. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4607–4619
45. Wu S, Shen S, Xu X et al (2022) Popularity-aware and diverse web apis recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10(2):771–782
46. Xie C, Zhang Z, Zhou Y, Bai S, et al (2019) Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2730–2739
47. Xiong Y, Lin J, Zhang M, et al (2022) Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans pp 14983–14992
48. Xu X, Li H, Li Z et al (2022) Safe: Synergic data filtering for federated learning in cloud-edge computing. IEEE Trans Ind Inform 19(2):1655–1665
49. Xu X, Tang S, Zhou X et al (2023) Cnn partitioning and offloading for vehicular edge networks in web3. IEEE Commun Mag. https://doi.org/10.1109/mcom.002.2200424
50. Xu W, Evans D, Qi Y (2017) Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint https://doi.org/10.48550/arXiv.1704.01155
51. Xu X, Gu J, Yan H, et al (2022) Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0. IEEE Trans Ind Inform 19(4):5485–5494
52. Yang Y, Yang X, Heidari M et al (2022) Astream: Data-stream-driven scalable anomaly detection with accuracy guarantee in iiot environment. IEEE Trans Netw Sci Eng. https://doi.org/10.1109/tnse.2022.3157730
53. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv preprint https://doi.org/10.48550/arXiv.1409.2329
54. Zhang J, Huang Jt, Wang W, et al (2023) Improving the transferability of adversarial samples by path-augmented method. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8173–8182
55. Zhou X, Zheng X, Cui X et al (2023) Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks. IEEE J Sel Areas Commun. https://doi.org/10.1109/jsac.2023.3310046
56. Zhou X, Ye X, Kevin I et al (2023) Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/tcss.2023.3259431

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.