A New Method to Detect the Adversarial Attack Based on the Residual Image

Feng Sun[1], Zhenjiang Zhang[2], Yi-Chih Kao[3], Tianzhou Li[1], Bo Shen[1]

[1] School of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, China
[2] School of Software Engineering, Beijing Jiaotong University, Beijing, China
[3] Information Technology Service Center, National Chiao Tung University, Taiwan
{sunfeng, zhangzhenjiang}@bjtu.edu.cn, ykao@mail.nctu.edu.tw, {17120189, bshen}@bjtu.edu.cn

Abstract

Nowadays, with the development of artificial intelligence, deep learning has attracted more and more attention. Whereas deep neural network has made incredible progress in many domains including Computer Vision, Nature Language Processing, etc, recent studies show that they are vulnerable to the adversarial attacks which takes legitimate images with undetected perturbation as input and can mislead the model to predict incorrect outputs. We consider that the key point of the adversarial attack is the undetected perturbation added to the input. It will be of great significance to eliminate the effect of the added noise. Thus, we design a new, efficient model based on residual image which can detect this potential adversarial attack. We design a method to get the residual image which can capture these possible perturbations. Based on the residual image we got, the detection mechanism can help us detect whether it is an adversarial image or not. A serial of experiments has also been carried out. Subsequent experiments prove that the new detection method can detect the adversarial attack with high effectivity.

Keywords: Adversarial attack, Detection mechanism, Residual image

1 Introduction

Deep Learning may be more and more sought after with the popularity of the Machine Learning (ML). Due to the universality of the deep neural network which is the heart of the deep learning, it has been applied in many domains, such as computer vision, DNN, natural language processing etc.

Meanwhile, some attractive applications also have been proposed. In image generation fields, Ian Goodfellow put forward the concept of generative adversarial network (GAN) [1]. The main idea is to establish a generator and a discriminator and the picture produced by the model can be more and more realistic according to the adversary of the generator and discriminator. Advanced model based on GAN such as Conditional GAN (CGAN) [2], Wasserstein GAN [3] has made great progress in the image generation field.

Generally, convolutional neural network (CNN) is the most universal model in Computer Vision. And recurrent neural network (RNN), or more specifically, Long Short-Term Memory recurrent network (LSTM) is more acceptable for people when it comes to natural language processing. However, all the applications or models mentioned above almost completely rely on the deep neural network. Thus the security of it rise to a high level.

Whereas deep learning performs a large number of tasks with high accuracies, Szegedy et al [4] find a potential risk of it when they do the experiment about image recognition. Though the high accuracy the network gets, it is vulnerable to the adversarial attack which put synthetical images with small perturbations as the input of neural network. Intriguingly, the image with small perturbation which is hard for human to detect leads to a misclassification for the neural network. For example, the following pictures will give us an intuitive display.

With the first sight, the pictures shown in Figure 1 appear to be the same. Not only they belong to the same category, but also they seem to be completely same as each other. Many may agree with the point that they are both a stop sign without hesitation.

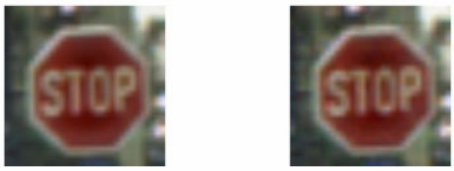

Figure 1. Example of images with & without adversarial attack

Actually, the picture on the left is the original one. It

*Corresponding Author: Zhenjiang Zhang; E-mail: zhangzhenjiang@bjtu.edu.cn
DOI: 10.3966/160792642019072004028

can be recognized as a stop sign by neural network as we expected. The picture on the right is the synthetical image with undetected perturbation. It can be classified as a yield sign by a specific DNN [5]. Obviously, it is dangerous when we use the automatic drive with deep neural network. The reason why this phenomenon happens we analyze as follow. As is known, neural network is the heart of deep learning. Due to the satisfactory accuracy, people use it by forward and backward propagation simply. Nevertheless, the mechanism behind it is not known to us. Deep neural network is treated as a black box. Thus, it is hard to predict the influence if a perturbation is added to the picture. If the specific place of image which plays an important role in classification is modified, it seems reasonable that the neural network predicts totally different outputs with two similar images.

Aiming at solving the problems mentioned above, we propose a new mechanism that can detect the image with adversarial attack. The main idea is to reduce the impacts that the added perturbation makes using the residual image. We first downsampling the raw image to a lower resolution in order to eliminate the effect of perturbation. Afterwards, we upsampling the thumbnail we get before. The target is to reconstruct the image without perturbation. Thus, we can get the residual image according to the raw image and the reconstruction image. The residual image we get can carry the information of perturbation added by attackers. We can finally detect the adversarial attack by the residual image.

Our contributions are:

1. Summary the popular model of adversarial attack. Analyze the principle of this type of attack and the attack procedure the attackers may take. Furthermore, the model introduced in this paper is based on the black-box situation which is more practical for the real.
2. We propose a new method to detect the adversarial attack based on the residual image. The residual image we get can capture the feature of perturbation effectively.
3. Extensive experiment results are provided to validate the effectiveness of our detection algorithm. We test our model on many datasets with different architecture of neural network.

The paper is organized as follows: some related works are listed in section II, and a threat model is introduced in section III. We propose our detection algorithm to defense the adversarial attack in section IV. Section V presents the experiment results and related analysis. The concluding remarks will be given in section VI.

2 Related Work

Since the findings of Szegedy [4], a serial of interesting theories and research results has been proposed by scholars. Moosavi-Dezfooli et al. [6] shows the universal among any image having the potential to fool the image classifier with the added perturbation. Athalye et al. [7] demonstrated that it is also vulnerable to the adversarial attack even in the 3-D print real-world domain. GoodFellow et al. [8] proposed an efficient Fast Gradient Sign Method (FGSM) to calculate the adversarial perturbation which can be added to the original image to fool the classifier. Su et al. [9] presented an aggressive attack named One Pixel Attack, trying to fool the classifier with only one pixel changed. The One Pixel Attack generate the adversarial images by testing the on every pixel iteratively. The adversarial image with the best effect will be reserved compared to the original image based on the selection standard. The advantage is that it doesn't need either the parameters of the network nor the information about the gradient any more. Meanwhile, Moosavi-Dezfooli et al. [10] find that the DeepFool which iteratively generates minimal perturbation by taking a step in the direction of the closest decision boundary has a better performance than the Fast Gradient Sign Method (FGSM). Whereas the adversarial attack models mentioned above show that they can effectively fool the classifier, they can only generate the adversarial perturbation aiming at single image. The Universal Adversarial Perturbations [6] can generate the adversarial perturbation universal to any type of picture. The perturbation generated by this method is also undetected to people.

While the number of adversarial attack model is increasing, more and more protection mechanisms to defense this type of attack are proposed. Some literatures [8, 11] attempt to take the adversarial examples into consideration in the process of train. Nevertheless, Moosavi-Dezfooli [6] points out the existence of new adversarial examples no matter how many adversarial examples are used in the progress of training. Data compressing was used by Dziugaite et al. [12]. However, the compression will affect the accuracy of the classification. Ross et al [13] proposed a novel perspective to defense the adversarial attack by regularizing the input gradients. Regularize the gradient of the input to improve the robustness of the defense to the adversarial attack. Despite it shows great performance on the task of defensing the adversarial attack, the algorithm complexity of it is particularly high. Xie et al [14] study the effect of the random-rescaling to the images in the training sets which generates a large family of adversarial examples and applies to a wide range of state-of-the-art deep networks for segmentation and detection. Gu and Rigazio [15] introduced the Deep Contractive Networks to improve the vulnerable network caused by the added Denoising Auto Encoders with simply stacked to the original network. Distillation [16] means migrating knowledge from complex networks to simple networks. Papernot et al [17] presents defensive distillation based on the distillation. And they prove

that it can resist the adversarial attack with small amplitude.

3 Threat Model

We consider the attacker targeting a multiclass classifier which is based on neural network. We assume that the neural network D is the target neural network where the $\tilde{D}(x)$ is the output of the target neural network. Here, $\tilde{D}(x)$ is the index assigned to the largest probability predicted by D:

$$\tilde{D}(x) = \arg \max_{j \in 1,2,\ldots,N} D_j(x) \quad (1)$$

where j is the j-th component of the target D, and is the input of the neural network.

Considering that the probability of every possible may also contain many information about the structure of the neural network, we assume that these probabilities should be carefully protected and can't be accessed by users. Our threat model is also based on this assumption. Attackers don't need to know the specific parameters of D or the dataset used to train it. The only capability the threat model has is accessing the label $\tilde{D}$.

The goal of adversarial is to produce a picture with undetected perturbation that can be misclassified by the target neural network D. Therefore, the adversarial is trying to find the optimal solution of the given problem as follows:

$$\begin{aligned} x^* &= x + \arg\min\{z: \tilde{D}(x+z) \neq \tilde{D}(x)\} \\ &= x + \delta_x \end{aligned} \quad (2)$$

where x^* is the adversirial sample and x is the legitimate input of the neural network. The formula above is to find the smallest perturbation δ_x added to the original picture that can lead to the misclassification. That is, the adversary is trying to find the shortest path that transform x to x^*.

Here, we take a practical black-box attack strategy as example. This method has two attractive advantages. One is that attackers don't need the detailed parameters to construct the network, the other is that there is no need to obtain the dataset used to train the destination network. This model actually brings tough challenges to the existing algorithm due to simple requirement and fairly good effect.

The practical black-box attack strategy contains attack model training and crafting adversarial examples. The essence of attack model training is to imitate the destination network locally. As the substitute network established locally, we can find the boundary of the classifier as both substitute network and destination network have similar decision boundary. It has been proved that adversaries can train a model with a different architecture with access to an independent training set from the oracle dataset [4] to replace the destination network. Once the process of training the substitute network is finished, we can then produce the adversarial examples using the fast gradient sign method [8] to gain the adversarial examples. The advantage of this method is that it finds the gradient of the function contained in the classifier. Therefore, we can get the expected adversarial examples with less queries to the destination network compared to the other method. We will introduce the model training and adversarial examples crafting respectively.

The main steps of this attack model training contains: 1. select substitute network architecture; 2. gain a synthetic dataset; 3. train substitute neural network. Figure 2 describe the flow chart of training the adversarial model.

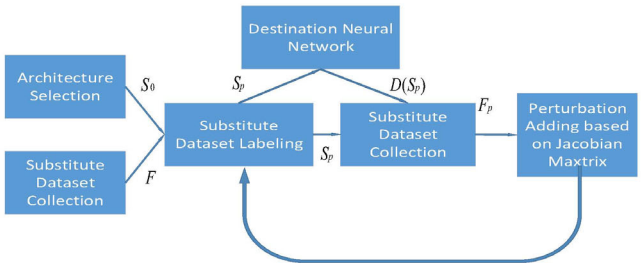

Figure 2. The steps of adversarial attack training based on black-box

Setting substitute architecture: This part is not very difficult for the adversary as they at least have some common knowledge about the neural network. They know the legitimate inputs (e.g., picture, text) and the type of the expected outputs (e.g., classification). Thus, the adversary can select the architecture of the neural network along with the relation between the input and output (e.g., the convolutional neural network is suitable for the image processing).

Generate a synthetic dataset: To generate a serial of legitimate inputs efficiently in an infinite number of queries to the destination neural network D, the attack model adopt a heuristic [8, 18]. The heuristic is trying to find the directions in which the output is varying. These directions are identified with the Jacobian matrix J_F of the network imitated by the adversary. By calculating the sign of the Jacobian matrix dimension related to the label with the input x, the adversary can thus get these directions: $\text{sgn}(J_F(x)[D(x)])$. Then, a perturbation $\tau \cdot \text{sgn}(J_F(x)[D(x)])$ is added to the original input x. The parameter τ decides the amplitude of the perturbation added to the picture. If the value of τ is set much higher than expected, it may differ from the original picture intuitively for people.

Substitute network training: The substitute network is aimed at simulate the target network locally. We assume that the adversaries have the minimal knowledge about the neural network. This means that they can select appreciate architecture and parameters

of the substitute network even with no knowledge about the target network. Simply, the adversaries can train the substitute network with the common knowledge about the neural network with the synthetic dataset we get before.

After the local substitute network training is finished, the next step of the adversarial model is to produce the adversarial examples using the substitute network. Here, we use the fast gradient sign method to find the gradient of the destination function of the target network. Thus, the adversarial examples can be described as follow:

$$x^* = x + \delta_x = x + \varepsilon \operatorname{sgn}(\nabla_D(F, x, y)) \quad \textbf{(3)}$$

where x represents the original input and x^* represents the adversarial examples. This formula is to find the cost function F of the of the target network D with the input x and its correct output y. Therefore, the adversaries can find the shortest path to transform the correct output to the expected output. We can finally obtain the adversarial examples.

4 A New Method Detecting the Adversarial Attack Based on Residual image

Aiming at detecting the adversarial attack mentioned before, we proposed a new algorithm to detect the adversarial attack. The heart of the adversarial attack theory is to add undetected perturbation to the original picture to fool the classifier based on the neural network. Thus, we can eliminate the perturbation by modify the picture the adversary sends.

The structure of our framework is shown in Figure 3.

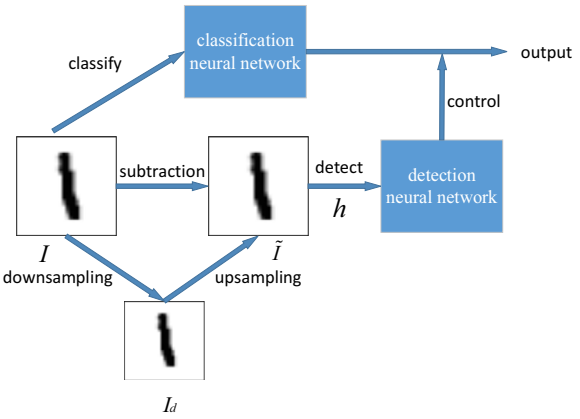

Figure 3. The structure of the detection algorithm

We first establish the classifier in the traditional way. At first, we should select the architecture of the network as it has a great effect on the recognition ratio. As we take the image classification as example, the convolutional neural network seems to be a better choice for the classifier architecture. Then it comes to the step of selecting the dataset which we use for training. What we should select into the dataset should correspond to the domain we are going to study. The quality of the dataset decides the final success ratio of the network we train. Using the dataset we select before to train the network with the back-propagation algorithm. Thus, we can get a standard neural network aiming at processing the task of classification. Notice that the output can be accessed by users only if the detection algorithm we proposed assigns the user with the normal user. We consider our detection model as an additional network to the classification network. The detailed information about our detection model will be given in the following.

The basis of our detection mechanism is residual generation and process, which will be introduce in the following. We find that the residual image has a good characteristic at reflecting the features of the perturbations added to the adversarial examples. We then select the appreciate residual image as the dataset to train the network used for detection and make a good result on defensing the adversarial attack. Some definitions associated with our model are given as follows. We assume that $d(\bullet)$ is a downsampling operation which decimates an $i \times i$ image I to get a new image I_d of size $j \times j$ by calculating $d(I, j)$, where $j < i$. Furthermore, we define $u(\bullet)$ a upsampling operator which smooths and expands image I with size of $i \times i$ to be a new image $\tilde{I}$ of size $j \times j$ by calculating $u(I, j)$, where $j > i$. There are four main steps in our detection mechanism: 1. downsampling image and upsampling; 2. get the residual image; 3. train the detection network; 4. detect the adversarial attack.

4.1 Downsampling Image and Upsampling

The first step in our detection algorithm is downsampling and then upsampling images. As what we mentioned before, the adversarial attack is based on the undetected perturbation added into the original image. Comressing the size of the picture can significantly improve the ability to defense the adversarial attack. Downsampling the image thus can reduce the impact of the perturbation as there is great possibility to eliminate the noise the adversary added to the image by calculate the mean value in the process of downsampling. Afterward, we use the picture $d(I, j)$ which eliminate the effect of the adversarial attack to reconstruct the original image according to the following formula.

$$\tilde{I} = u(I_d, i) = u(d(I, j), i) \quad \textbf{(4)}$$

where I_d is the downsampling image of raw image I. And it is required that $j < i$. What should be brought to our attention is that the value of the parameter j. The value of this parameter can affect the performance of the detection success rate, which be discussed in section V.

4.2 Residual Image

The significance of the residual image is that it provides us a different perspective to review the image we get. Adversarial examples with added noise will contain some regulation behind the original image. The residual, however, can help us to enlarge the hidden regulation.

We get a pure image by downsampling and upsampling the raw image. And we can get the residual by comparing the difference between the reconstruction image and raw image:

$$\tilde{h} = I - \tilde{I} = I - u(d(I, j),\ i) \quad \textbf{(5)}$$

The target of residual image is to obtain an intuitive statement of the possible perturbation as the image information has been eliminated to a great degree. Therefore, we can use the information of residual image to decide if the image is an adversarial attack or not.

4.3 Train the Detection Network

After the residual image is generated, the next step is to establish the detection network. The first and the most chanllenging problem is the establishment of the dataset which is used to train the detection network. Considering that the adversarial attack aims at changing the output result by updating the model along the maximum gradient direction, it turned out that the adversarial model train locally has the similar attribute to the real attack model. Therefore, we establish a local adversarial model to imitate the attacker's behavior.

Our detection mechanism is based on the black-box strategy. Thus, we assume that the local adversarial model has no access to the dataset as well as the parameters of the targer network. We train the adversarial model based on the method mentioned in Section 3. The purpose we train the local adversarial model is to obtain the adversarial examples to establish the dataset used to train the datection network. Meanwhile, the dataset we established should also contain the same number of the original images with their label. The two types of train data containing original image and adversarial examples output for the train step in equal probability.

4.4 Detect the Adversarial Attack

We finally establish a specific neural network to recognize the adversarial attack. The network takes the residual image as input. The features of the residual image will be organized and classified in the neural network. Note that the output should be secret and can't be accessed by users. Otherwise, the detection neural network may be attacked by others as it is also vulnerable to the adversarial attack.

The procedure of the detection algorithm is shown as follows:

5 Experiment Analysis

We first validate the threat model on some popular dataset. It proves the conclusion that the neural network is indeed vulnerable to the adversarial attack. The success rate of attack can be up to 90%. Thus it is an emergency to defense this type of attack. Then, we test our model based on the adversarial attack we validate before. It shows great performance on detecting the adversarial attack. Detailed experiment results and analysis will be given in the following.

5.1 Verification on thrEat Model

We first set the destination neural network. We used the MNIST hand-written digit dataset to train the neural network. It contains 60000 training images and 10000 test images. Each image has the corresponding label. Note that in order to ensure that the parameters keep secret to our attack model, we use the third-party website MetaMind to train the model of classifier. In the whole process of training, we have no access to the model architecture or the specific parameters. More information can be found in the MetaMind website.

It takes 36 hours to get a classifier with 94.4% accuracy. After training, we can get the output of the classifier. We compare two sources of the dataset which is used for the substitute network training in threat model: MNIST dataset, and Handcrafted dataset.

MNIST dataset: we select 200 samples from MNIST dataset. We assume that the adversarial can collect some legitimate inputs. Subsequently, the attacker uses the dataset to gain the adversarial examples by the method in section III.

Handcrafted dataset: this is based on some potential situation that the adversarial can't obtain the original inputs. And we handcrafted 100 samples instead.

Note that all the assumption we proposed is based on that the attacker has a minimal knowledge about neural network. To validate the accuracy of the local substitute network for adversarial attacker, we randomly select 200 samples from MNIST test set. The accuracy can be seen in Table 1.

Table 1. The pseudo code of detection

The procedure of the detection algorithm
Input: images that users submit with size $s \times s$
1. **For** image i from 1 to N in images:
2. $\tilde{I}_i = u(I_{di},\ \mathrm{s}) = u(d(I_i, j),\ \mathrm{s})$
3. $\tilde{h} = I - \tilde{I}$;
4. put $\tilde{h}$ into detection neural network;
5. **if** image i is adversarial, then:
6. report the attack and don't response users;
7. **else**
8. show classification output to users;
9. **end**
10. **end**

From the Table 2, we can see that the recognition accuracy grows steady with the training going on. The accuracy reaches 75.8% in 5 epochs. Although the performance of the accuracy based on Handcrafted dataset is not as good as MNIST dataset, it still reaches 55.7% accuracy.

Table 2. The success rate of adversarial model with different dataset

Epoch	Training by MNIST dataset	Training by Handcrafted dataset
1	23.7%	19.1%
3	60.3%	35.4%
5	75.8%	55.7%
10	89.3%	75.2%
20	90.7%	83.3%

As the training is going on, the accuracy with the Handcrafted dataset eventually reaches 83.3%, almost close to the destination neural network. This table can prove that even if the adversarial can obtain few information about the original dataset, the destination network can still be modified by the attacker in remote. If the adversarial has access to part of the original data used for training the destination network, the training for the adversarial model will be accelerated. Thus, protecting the dataset from being attacked is an important plan for protecting the target network, but it seems far from enough.

Then, we test the adversarial model on MNIST dataset and GTRSRB dataset as shown in Figure 4. The adversarial model tested on the MNIST dataset divides into two part: the substitute network of one is trained using MNIST dataset while the other is Handcrafted dataset. We select the result while the epoch is 40 to get a better performance.

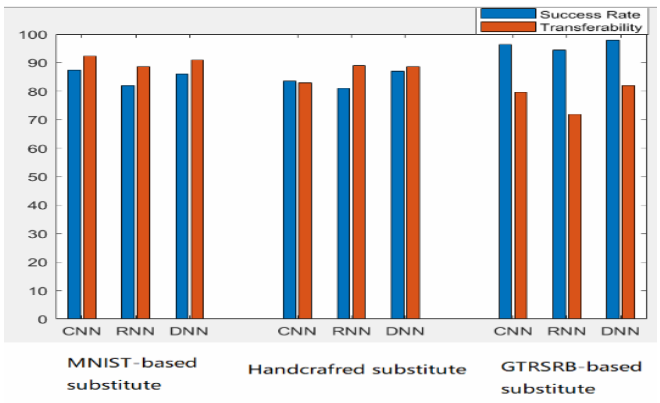

Figure 4. The success rate of the adversarial model on different dataset

We not only test the success rate of adversarial attack on the substitute network trained by attackers which labeled as "Success Rate", but also contains the success rate on destination network labeled as "Transferability". The "Success Rate" means the rate of successful attack on substitute network. This rate reveals the performance of the adversarial attack preliminarily. Considering that the substitute network may differ from the destination network, we set the observation value "Transferability" to validate the performance on the destination network, which may be more significative to the attacker. As you can see, the success rate and transferability appear similar among all the structure of neural network. It appears that it is a common problem for all architectures of neural network. Meanwhile, we can see that the attack model performs well on the MNIST dataset no matter whether it is trained by MNIST dataset or the HandCrafted dataset. However, when it comes to the GTRSRB dataset, its performance on the substitute networks differs greatly from that on the destination network. The reason to phenomenon perhaps lies in the fact that the image from the GTRSRB dataset is fairly complex to the image from MNIST. Therefore, the network for classification should be more complex. Thus, the result we get may not satisfactory as the model tested on MNIST. The complexity of the data from the GTRSRB dataset increase the difficulty of adversarial attack. Nevertheless, the success rate still can reach up to 97% while the transferability can reach up to nearly 80%.

5.2 Test on the Detection Model

Based on the adversarial attack model mentioned before, we test our detection model. We assume that the attack is based on the black-box. Thus, there is no need to concern about the leak of the parameters and the dataset of the target model. Also, we select different architecture as to test the universal of the efficiency of our model to defense the adversarial attack. Here, we take the detection model trained on the MNIST dataset as example. Also, we train an adversarial model locally as the local adversarial attack model in order to provide the adversarial examples to train the detection neural network. The local adversarial attack model is independent of the attack model. After 30 epochs training, the success rate of adversarial attack made by the local adversarial model can reach 90.1%. We use the adversarial examples produced by the local adversarial model to train the detection model.

We also research the effect on the detection accuracy when we set different compression rate (CR) in the process of downsampling. The compression rate is also an important assessment index as it may affect the accuracy of the destination network. The compression rate mentioned here presents the proportion of the compressed images compared to the original image.

From the Figure 5, we can see that our method completes the mission of defensing the adversarial attack with nearly 90% after 100 epochs when the CR is set to 5%. With the training going on, the detection accuracy grows with different degrees. However, when it comes to the situation where CR is set to 10% or

more, the performance declines sharply. This may because that some information missed in the process of downsampling. The image of reconstruction differs from the original image as well, leading to the result that the residual image can't reflect the features of the perturbation exactly. If the reconstruction image differs from the original image greatly, then the residual image we get will contain some information about the image itself, which can be known from the formula 5. And the detection network has no ability to recognize these residual image. Our model has a certain degree of universality. There is no clear difference among the three neural network model we test.

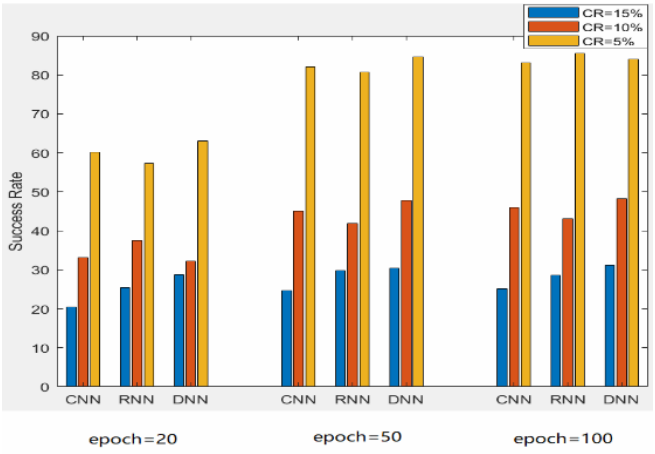

Figure 5. Success rate of detection with different CR

The performance of the detection model based on the different dataset can be seen in the Table 3. Here, we set CR = 5% and epoch = 50.

Table 3. Success rate on different dataset

Dataset	Success Rate
MNIST	87.1%
Handcrafted Dataset	85.5
GTRSRB	83.2%
AlexNet	84.1%

It appears that the success rate of the GTRSRB dataset and AlexNet database is lower than the others. The main reason for this phenomenon is that the information contained in the image from the GTRSRB dataset and AlexNet dataset is much greater than those from the MNIST dataset. Therefore, these images may loss part of the information, which has great impact on the reconstruction image. As we mentioned before, the detection algorithm we proposed is based on the residual image, which is corresponding to the reconstruction image. So, there is no difficult to understand that the success rate of the detection model on the GTRSRB dataset and AlexNet dataset declines sharply.

6 Conclusion

In this paper, we have investigated a practical adversarial attack model based on black-box at first. We show the existence of potential risk to the neural network. Based on the threat model, we designed a detection mechanism based on residual image to detect the adversarial attack. We show that it is appropriate for our model to defense the adversarial attack with great performance. Meanwhile, we can also see several future research directions. For example, the method we propose can only detect the adversarial attack rather than predicett the attack examples with the right output. In other respect, we test our model based on the practical black-box adversarial attack. And we assume that the attacker has no access to parameters of the destination network as well as the dataset for training. Our detection mechanism should be validated on the other threat model.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, *Conference and Workshop on Neural Information Processing Systems* (NIPS), Montreal, Canada, 2014, pp. 1-9.

[2] M. Mirza, S. Osindero, *Conditional Generative Adversarial Nets*, arXiv preprint arXiv: 1411.1784, 2014.

[3] M. Arjovsky, S. Chintala, L. Bottou, *Wasserstrin GAN*, arXiv preprint arXiv: 1701.07875, 2017.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing Properties of Neural networks, *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014, pp. 1-10.

[5] N. Papernot, P. McDeniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, Practical Black-Box Attacks against Machine Learning, *2017 ACM Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, 2017, pp. 506-519.

[6] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal Adversarial Perturbations, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 86-94.

[7] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing Robust Adversarial Examples, *International Conference on Machine Learning (ICML)*, Stockholm, Sweden, 2018, pp. 1-19.

[8] I. J Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, *International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015, pp. 1-11.

[9] J. Su, D Vargas, S. Kouichi, *One Pixel Attack for Fooling Deep Neural Networks*, arXiv preprint arXiv: 1710.08864, 2017.

[10] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, DeepFool: A

Simple and Accurate Method to Fool Deep Neural Networks, *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 2574-2582, 2016.

[11] S. Sankaranaraynan, A. Jain, R. Chellappa, S. N. Lim, Regularizing Deep Nerworks Using Efficient Layerwise Adversarial Training, *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, New Orleans, LA, 2018, pp. 4008-4015.

[12] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A Study of the Effect of JPG Compression on Adversarial Images, *the International Society for Bayesian Analysis (ISBA 2016) World Meeting*, Sardinia, Italy, 2016, pp. 1-8.

[13] A. S. Ross, F. Doshi-Velez, Improving the Adversarial Robustness and Interpretability of Deep Neural Network by Regularizing their Input Gradients, *Association for the Advancement of Artificial Intelligence*, New Orleans, LA, 2018, pp. 1660-1669.

[14] C. Xie, J Wang, Z. Zhang, Y. Zhou, L. Xie, A. Yuille, Adversarial Examples for Semantic Segmentation and Object Detection, *IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 1378-1387.

[15] S. Gu, L. Rigazio, Towards Deep Neural Network Architectures Robust to Adversarial Example, *International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015, pp. 1-9.

[16] G. Hinton, O. Vinyals, J. Dean, Distilling the Knowledge in A Neural Network. *Deep Learning and Representation Learning Workshop at NIPS*, Montreal, Canada, 2014, pp. 1-9.

[17] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, *IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, 2016, pp. 582-597, 2016.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The Limitations of Deep Learning in Adversarial Settings, *1st IEEE European Symposium on Security and Privacy*, Saarbrucken, Germany, 2016, pp. 372-387.

Biographies

Feng Sun received the bachelor's degree in the School of Electronic and Information Engineering, Beijing Jiaotong University in 2017. He is currently pursuing his Ph.D. degree in communication engineering at the same university.

Zhenjiang Zhang received the Ph.D. degree in communication and information systems from Beijing Jiaotong University. He has been a Professor in BJTU since 2014. He is currently served as the vice dean of School of Software Engineering in BJTU. His research interests include cognitive radio, and wireless sensor networks.

Yi-Chih Kao is currently the director of the Network and System Division of Information Technology Service Center at National Chiao Tung University (NCTU). He received his Ph.D. degree in Industrial Engineering and Management from NCTU. His research interests include cyber security, network performance, software-defined networking, and IT service design.

TianZhou Li received the bachelor's degree in the School of Electronic and Information Engineering, Beijing Jiaotong University in 2017. He is currently pursuing his Master's degree in communication engineering at the same university.

Bo Shen received the Ph.D. degree in communication and information systems from Beijing Jiaotong University (BJTU), Beijing, China, in 2006. He has been a Professor in BJTU. Pro Shen has published about 40 professional research papers. His research interests include cyber security, Information network, and electronics and communication engineering.