Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8891204, 8 pages
https://doi.org/10.1155/2021/8891204

Research Article

Detection Mechanisms of One-Pixel Attack

Peng Wang, Zhipeng Cai, Donghyun Kim, and Wei Li

Department of Computer Science, Georgia State University, Atlanta 30303, USA

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 21 September 2020; Revised 22 December 2020; Accepted 3 February 2021; Published 23 February 2021

Academic Editor: Wenzhong Li

In recent years, a series of researches have revealed that the Deep Neural Network (DNN) is vulnerable to adversarial attack, and a number of attack methods have been proposed. Among those methods, an extremely sly type of attack named the one-pixel attack can mislead DNNs to misclassify an image via only modifying one pixel of the image, leading to severe security threats to DNN-based information systems. Currently, no method can really detect the one-pixel attack, for which the blank will be filled by this paper. This paper proposes two detection methods, including trigger detection and candidate detection. The trigger detection method analyzes the vulnerability of DNN models and gives the most suspected pixel that is modified by the one-pixel attack. The candidate detection method identifies a set of most suspected pixels using a differential evolution-based heuristic algorithm. The real-data experiments show that the trigger detection method has a detection success rate of 9.1%, and the candidate detection method achieves a detection success rate of 30.1%, which can validate the effectiveness of our methods.

1. Introduction

Deep learning is an artificial intelligence technique that follows the structure of a human's brain and imitates the neural cells in the human brain [1]. Over the past decades, deep learning has made significant progresses in speech recognition, natural language processing [2], computer vision [3], image classification [4], and privacy protection [5–7]. In particular, with the increase of data volume, traditional machine learning algorithms, such as SVM [8] and NB [9], suffer a performance bottleneck, in which adding more training data cannot really enhance their classification accuracy. Differently, the deep learning classifiers can continue to get improvements if more data is fed. However, it has been revealed that artificial perturbation can make the deep learning models misclassify easily. A number of effective methods have been proposed to produce so-called "adversarial samples" to fool the models [10, 11] and some work focused on fighting against adversarial attack [12–14].

1.1. One-Pixel Attack. Among the existing works, the one-pixel attack takes an extreme scenario into consideration, where only one pixel of an image is allowed to be modified to mislead the classification models of the Deep Neural Network (DNN) such that the perturbed image is classified to another label different from the image's original label [15]. As shown in Figure 1, with the modification of one pixel, the classification of images is changed to totally irrelevant labels.

The one-pixel attack is harmful to the performance guarantee of DNN-based information systems. Via modifying only one pixel in an image, the classification of the image may change to an irrelevant label, leading to performance degradation of DNN-based applications/services and even other serious consequences. For examples, in medical image systems, the one-pixel attack may make a doctor misjudge the disease of patients, and in autodriving vehicles, the one-pixel attack may cause serious traffic accidents on roads.

More importantly, the one-pixel attack is more threatening than other types of adversarial attack as it can be implemented easily and effectively to damage system security. Since the one-pixel attack is a type of black box attack, it does not require any additional information of the DNN models. In practice, the one-pixel attack only needs the probabilities of different labels instead of the inner information about the target DNN models, such as gradients and hyperparameters. The effectiveness of the one-pixel attack towards DNNs has been validated in [15], where the attack success rate is 31.40% on the original CIFAR-10 image dataset and

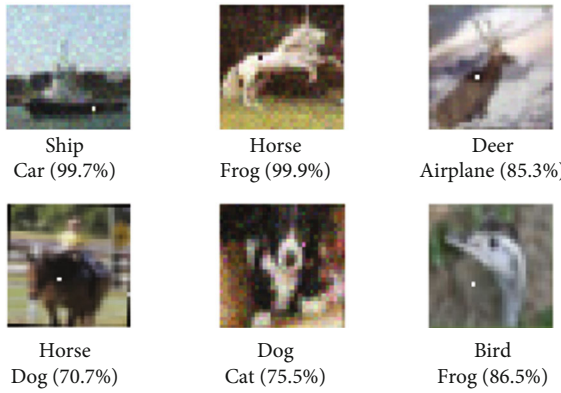

Figure 1: An example of one-pixel attack from [15].

16.04% on the ImageNet dataset. Such a success rate is large enough to break down an image classification system.

Therefore, to avoid the loss of system performance, detecting the one-pixel attack becomes an essential task.

1.2. Technical Challenges. The following two facts result in the difficulty of examining a one-pixel attack in images.

(1) *Extremely Small Modification*. The one-pixel attack modifies only one pixel in an image, which is significantly less than other types of adversarial attack. This makes the detection of the one-pixel attack very challenging.

(2) *Randomness of Pixel Modification*. For an image, there may be more than one feasible pixel that can cause the change of classification. In [15], the one-pixel attack randomly selects one of those feasible pixels to mislead the classifiers. Such randomness makes the detection of the one-pixel attack harder.

1.3. Contributions. In this paper, we develop two methods to detect a one-pixel attack for images, including trigger detection and candidate detection. In the trigger detection method, based on a concept named the "trigger" [16], we use gradient information of the distance between the pixels and target labels to find the pixel that is modified by the one-pixel attack. By considering the property of the one-pixel attack, in the candidate detection method, we design a differential evolution-based heuristic algorithm to find a set of candidate victim pixels that may contain the modified pixel. Intensive real-data experiments are well conducted to evaluate the performance of our two detection methods. To sum up, this paper has the following multifold contributions.

(i) To the best of our knowledge, this is the first work to study the detection of the one-pixel attack in literature, which can contribute to the defense of the one-pixel attack in future research

(ii) Two novel detection mechanisms are proposed, in which the modified pixels can be identified in two different ways based on our thorough analysis on the one-pixel attack

(iii) The results of real-data experiments validate that our two detection methods can effectively detect the one-pixel attack with satisfied detection success rates

The rest of this paper is organized as follows. In "Related Works," the existing works on adversarial attacks and detection schemes are briefly summarized. The attack model and the detection model are presented in "System Models." Our two detection methods are demonstrated in "Design of Detection Methods." After analyzing the performance of our methods in "Performance Validation," we conclude this paper and discuss our future work in "Conclusion and Future Work."

2. Related Works

A one-pixel attack is a special type of adversarial attack and is designed based on a differential evolution scheme. Thus, this section summarizes the most related literatures from the following two aspects: adversarial attack and detection of an adversarial attack.

2.1. Adversarial Attack. In an adversarial attack, attackers intend to mislead classifiers by constructing adversarial samples. Nguyen et al. made efforts on fooling a machine learning algorithm [17] and found that DNNs give high confidence results to random noise, which indicates that universal adversarial perturbation in a single crafted perturbation can cause a misclassification on multiple samples. In [10, 18], back-propagation is used to find gradient information of machine learning models, and gradient descent methods are used to build adversarial samples.

Since it might be hard or impossible to learn the internal information of a DNN model in practice, some approaches have been proposed to generate adversarial samples without using the internal characteristics of DNN models. Such approaches are called a black box attack [19–21]. Particularly, a special type of black box attack is the one-pixel attack, in which only one pixel is allowed to be modified. Under the one-pixel attack of [15], an algorithm was developed to find a feasible pixel for malicious modification based on differential evolution that has a higher probability of finding an expected solution compared with gradient-based optimization methods. Due to the concealed modification of only one pixel, it becomes more difficult to detect the one pixel attack. As mentioned in [15], the one-pixel attack requires only black box feedback that is the probability label without any inner information of the target network, like gradients or structure.

2.2. Detection of Adversarial Attack. On the other hand, research attention is also paid to work out detection methods for adversarial attack. Papernot et al. provided a comprehensive investigation into the security problems of machine learning and deep learning, in which they established a threat model and presented the "no free lunch" theory showing the tradeoff between accuracy and robustness of deep learning models. Inspired from the fact that most of the current datasets are compressed JPG images, some researchers designed a method to defend image adversarial attack using image compression. However, in their proposed method, a large

compression may also lead to a large loss of classification accuracy of the attacked images, while a small compression cannot work well against adversarial attack. In [16], Neural Cleanse was developed to detect a backdoor attack in neural networks, and some methods were designed to mitigate backdoor attack as well.

Compared with the existing works, this paper is the first work that focuses on the detection of the one-pixel attack. In particular, two novel detection mechanisms are proposed with one using a gradient calculation-based method and the other using a differential evolution-based method.

3. System Models

The attack model and detection model in our considered DNN-based information systems are introduced as follows.

3.1. Attack Model. In this paper, the attack model of [15] is taken into account, in which an adversarial image is generated by modifying only one pixel in the victim image. The purpose of a one-pixel attack is to maliciously change the classification result of a victim image from its original label to a target label. As shown in Figure 2, the image is correctly classified as an original label, "sheep," by a given DNN model. After being modified one pixel, the output label with the highest preference of the model is changed to a target label, "airplane," leading to a wrong classification result. The attackers perform a black box attack only, which means they have the accessibility to the probability labels and cannot get the inner information of the network. Also, considering that the attacker aims to make the attack as efficiently as possible, it is supposed that all the images in the dataset are altered.

3.2. Detection Model. Suppose that a set of adversarial images, which are modified by the aforementioned one-pixel attack, are given to the system. With these given images, our objective is to distinguish which pixel has been modified by a one-pixel attack.

To detect the one pixel attack, two novel methods are developed. The first method is the "trigger detection" to identify the modified pixel, and the second one is the "candidate detection" to find a set of victim pixels. The "trigger detection" model is designed for white box detection that requires all the network information including inner gradients and network structure. In the trigger detection model, we first propose a new concept named "trigger" for image data and then detect the trigger in a given adversarial image. If the detected trigger is the pixel modified by the one-pixel attack, our detection is successful. The "candidate detection" is for the black box detection, where only the output probabilities of labels are needed for the detection. In the candidate detection model, we aim to find a set of pixels as the candidate victim pixels. If the selected victim pixels include the pixel modified by the one-pixel attack, our detection is successful. The details of our two detection mechanisms are demonstrated in "Design of Detection Methods."

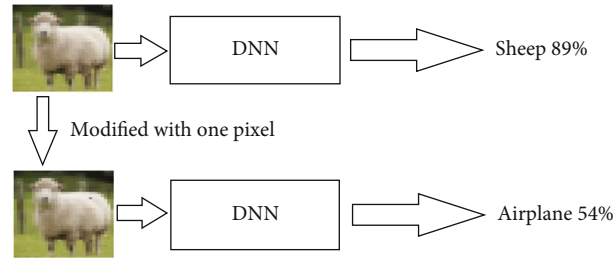

Figure 2: Illustration of one-pixel attack.

4. Design of Detection Methods

Our proposed detection methods are described as follows. For a better presentation, the major notations are summarized in Table 1.

4.1. Trigger Detection Method

4.1.1. Main Steps. Formally, the trigger of an image is defined to be the pixel that has the greatest impact on the model classification. Thus, any image, which has a properly modified trigger, should have a higher confidence on a target label and will be likely to be classified as the target label regardless of other unchanged image features. In other words, the classification result would be changed to a target label if the trigger is modified properly using DNNs' properties.

Let $\mathbb{L}$ represent the set of output labels in a DNN model. For any adversarial image, we assume its original label is $L_{\text{in}} \in \mathbb{L}$ and its target label is $L_t \in \mathbb{L}$, where in $\neq t$. Under a one-pixel attack model, the modification on the trigger pixel can transform all inputs of L_{in} to be classified as L_t.

Motivated by the above observations, the one-pixel attack can be detected via identifying the triggers of adversarial images according to the following steps.

Step 1. For a given label, we treat it as a potential target label in the one-pixel attack. We use an optimization-based scheme to find the trigger that can misclassify all samples from other labels into the target label.

Step 2. We repeat Step 1 for each output label in the DNN model and obtain N potential triggers, in which $N = |\mathbb{L}|$ is the number of labels of the DNN model.

Step 3. After calculating N potential triggers, we measure the size of each trigger. The size of a trigger is defined to be the modified RGB value of the trigger pixel. Then, the outlier detection algorithm of [22] is adopted to detect whether the perturbation of each potential trigger is significantly smaller than others. A significant outlier is likely to indicate a real trigger, and the label matching the real trigger is the target label in the one-pixel attack.

4.1.2. Trigger Identification. The details of our optimization-based scheme for identifying trigger are addressed below.

Table 1: Notations.

Notations	Meaning
$\mathbb{L}$	Output labels in a DNN model
L_{in}	The original label
L_t	The target label of a one-pixel attack
$A(\bullet)$	The function that applies a trigger to an image
x	The original image
x'	The modified image A pixel of x and $x_{i,j,c}$ represents a pixel of x'
$x_{i,j,c}$	i and j are the x and y coordinates of the pixel, respectively, and c is the color channel
Δ	A trigger of an image
$\text{Loss}(\bullet)$	The loss function measuring classification error
x_{next}	An element of the candidate solution
$T_{\max}$	Maximum iteration numbers

Suppose X is a set of clean images without modification. A generic form of injecting trigger for any original image, $x \in X$, is given in

$$A(x, m, \Delta) = x', \tag{1}$$

where $A(\cdot)$ represents the function that applies a trigger to x. Correspondingly, the modified image is denoted as x'. Let $x_{i,j,c}$ represent a pixel of x and $x'_{i,j,c}$ represent a pixel of x', where i and j are the x and y coordinates of the pixel, respectively, and c is the color channel. The relationship between x and x' can be mathematically expressed by

$$x'_{i,j,c} = (1 - m) \cdot x_{i,j,c} + m \cdot \Delta_{i,j,c}, \tag{2}$$

in which Δ represents a trigger of x and $m \in [0, 1]$ describes how much Δ can overwrite the original image. Particularly, when $m = 1$, the pixel of the trigger completely overwrites the original color, and when $m = 0$, the original color is not modified at all. The original image x is classified to the original label L_{in}, and the modified image x' is classified to the target label L_t.

Then, given the target label L_t, the problem of finding a trigger can be formulated as a multiobjective optimization problem, i.e.,

$$\min_{m,\Delta} \text{Loss}(L_t, f(A(x, m, \Delta))) + m, \quad \forall x \in X. \tag{3}$$

In Eq. (3), $\text{Loss}(\cdot)$ is the loss function measuring classification error that is computed by cross entropy and $f(\cdot)$ is the prediction function of the DNN model.

In this paper, L1 norm of m is adopted to measure the magnitude of the trigger. By solving the above optimization problem, we get the trigger, Δ, for each target label and its L1 norm. Next, in Step 3, we identify the triggers that show up as outliers with smaller L1 norm by utilizing the outlier detection algorithm of [22].

4.2. Candidate Detection Method. Notably, to generate adversarial images, the one-pixel attack of [15] uses a differential evolution algorithm to randomly select a pixel that can lead misclassification on an image. Mathematically speaking, for the problem of image generation, the selected pixel is a feasible solution and may not be an optimal solution. Therefore, the obtained trigger pixel might not be actually modified in the one-pixel attack, resulting in failed detection. In order to improve the attack detection success rate, the goal of our candidate detection method is to find a set of victim pixels (i.e., a set of feasible solutions), each of which satisfies the requirement of adversarial image generation in the one-pixel attack.

4.2.1. Problem Formulation. Without loss of generality, we assume an input image can be represented by a vector in which each scalar element represents one pixel. In a DNN model, $f(\cdot)$ receives an image as input and gives confidence of N labels. Accordingly, the probability of x being classified to a label $L \in \mathbb{L}$ is $f(x)[L]$. For an original image x, an additive adversarial perturbation is represented by a vector v. The modification degree is measured by the length of v, and the allowable maximum modification is 1 in the one-pixel attack model. The problem of generating adversarial images using the one-pixel attack is formulated as follows.

$$\begin{aligned} \max_{\Delta} \quad & f(x + \Delta)[L], \\ \text{s.t.} \quad & \|\Delta\| \le 1. \end{aligned} \tag{4}$$

4.2.2. Differential Evolution-Based Heuristic Algorithm. To obtain the solution to Equation (4), a heuristic algorithm is designed based on differential evolution (DE), which brings the following benefits.

(i) DE gives a higher probability of finding global optimal solutions as well as a lower probability of getting "trapped" in the local solutions compared with gradient descent and greedy searching algorithms

Input: an adversarial image generated by a one-pixel attack, a DNN classifier, and a label set $\mathbb{L}$ with $|\mathbb{L}| = N$
Output: a set of pixels containing C candidate victim pixels
1: Initialize model with the target DNN model
2: Randomly chose N_{ini} pixels from the image, and for each point, randomly set a color as present pixel set
3: Calculate the confidence of the image on all N labels
4: **for all** each label $L \in \mathbb{L}$ **do**
5: **while** true **do**
6: Calculate the change of the confidence on L when modifying with parent pixels
7: Generate offspring pixels based on Equation (5)
8: Calculate the change of confidence on L when modifying with offspring pixels
9: Select N_c pixels with the highest confidence changed as new parent pixels
10: **if** all top C confidences changed on targets are larger than p_{th} **then**
11: Save the top C pixels as candidate victim pixels
12: Set AttackSucc = true
13: Break while loop
14: **end if**
15: **if** while loop is over $T_{\max}$ times **then**
16: Break while loop
17: **end if**
18: **end while**
19: **end for**
20: **if** AttackSucc is true **then**
21: **return** C candidate victim pixels
22: **else**
23: **return** fail to find candidates
24: **end if**

Algorithm 1 Algorithm of candidate detection method.

(ii) DE requires less information from the optimization objectives. DE does not require gradient information from the dataset, which means it even does not require the problem to be differentiable. Under the extremely strict constraint of modifying only one pixel in an image, the problem is not differentiable and can be effectively resolved by DE

(iii) To detect a one-pixel attack, we only need to know whether the confidence changes after modifying a pixel, which can be formulated and solved in a simple way using DE

In this paper, we encode the perturbation into an array (i.e., a candidate solution) which is optimized (evolved) by differential evolution. One candidate solution contains a fixed number of perturbations, and each perturbation is a tuple holding five elements including x-y coordinates and RGB value of the perturbation, where one perturbation modifies one pixel. The DE algorithm is performed iteratively and is terminated when one of the two conditions is satisfied: (i) the maximum number of iteration $T_{\max}$ is reached or (ii) the probability of being classified to the target label exceeds a threshold p_{th}. Let N_{ini} be the initial number of candidate solutions (population) and N_c be the number of candidate solutions (i.e., children) produced in each iteration. At the $(g+1)$-th iteration, N_c candidate solutions are produced from the g-th iteration via the following DE formula:

$$x_{\text{next}}(g+1) = x_{r_1}(g) + F\left(x_{r_2}(g) - x_{r_3}(g)\right), \tag{5}$$

where x_{next} is an element of the candidate solution; r_1, r_2, and r_3 are random values with $r_1 \neq r_2 \neq r_3$; and F is the scale parameter. After being generated, each candidate solution competes with their parents according to the index of the population, and the winners survive in the next iteration. When the algorithm terminates, C candidates are output.

The pseudocode of our algorithm is shown in Algorithm 1. For each image, the above algorithm will go through all the N labels, i.e., the "for loop" in lines 4-19 of Algorithm 1. For each label, the candidate selection process will run up to $T_{\max}$ iterations, i.e., the "while loop" in lines 5-18 of Algorithm 1. Each iteration costs a constant time to generate N_c children and pick N_c winners. As a result, the time complexity of Algorithm 1 is $O(N \cdot T_{\max})$.

5. Performance Validation

In this section, extensive real-data experiments are conducted to evaluate the performance of our two detection methods.

5.1. Experiment Settings. Our experiments adopt CIFAR-10 as the dataset and VGG-16 as the DNN model. Table 2 shows the structure of the VGG-16 network which is the same as the network used in a one-pixel attack. After training, we get the model with the accuracy as shown in Table 3.

To measure the performance of the one-pixel attack, we calculate the classification accuracy of the VGG-16 model on the test images. Also, we calculate the success rate of launching the one-pixel attack, in which "airplane" is set as

Table 2: Network structure of VGG-16.

Conv2d layer (kernel = 3, stride = 1, depth = 64)
Conv2d layer (kernel = 3, stride = 1, depth = 64)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 128)
Conv2d layer (kernel = 3, stride = 1, depth = 128)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Conv2d layer (kernel = 3, stride = 1, depth = 256)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Max pooling layer (kernel = 2, stride = 2)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Conv2d layer (kernel = 3, stride = 1, depth = 512)
Max pooling layer (kernel = 2, stride = 2)
Flatten layer
Fully connected (size = 2048)
Fully connected (size = 2048)
Softmax classifier

Table 3: Classification accuracy of VGG-16.

Model	Accuracy on test set	Claimed accuracy on test set
VGG-16	93.4%	94%

Table 4: Success rate of one-pixel attack.

Model	Accuracy on test set	Claimed accuracy on test set
VGG-16	93.4% ± 1.65%	17.3% ± 3.61%

the target label. The success of the one-pixel attack means after being modified a pixel, an image whose original classified label is not airplane is misclassified to airplane by the DNN network. Moreover, to reduce the influence caused by the randomness in the differential evolution algorithm of the one-pixel attack, the classification process is repeated 5 times, and the average accuracies and their variances are presented in Table 4.

Specifically, the one-pixel attack is launched towards 60,000 testing images and succeeds in making 8655 nonairplane images get classified to airplane. In the experiments, we pick 8000 such adversarial images to evaluate our detection method.

5.2. Performance Metrics. The performance metrics are introduced as follows.

(i) *Label confidence*. The label confidence is the confidence of different labels given by the DNN model to an image. For a label, higher confidence means a higher probability that the image is classified to the label.

(ii) *Detection success rate*. The definition of successful attack detection is different in our two detection methods. In trigger detection, our detection is successful if the detected trigger is the pixel modified by a one-pixel attack, while, in the candidate detection model, our detection is successful if the pixel modified by the one-pixel attack is included in the set of selected victim pixels. For both detection methods, the detection success rate is defined as the ratio of the number of successful detection to the number of adversarial images.

5.3. Performance of Trigger Detection. Since the L1 norm has a better feature selection performance and interpretability [23], our trigger detection method uses the L1 norm to measure the distance between the pixel and a label. If the L1 norm of a pixel is obviously different from others, we can consider that the pixel is infected. Figure 3 shows the L1 norm of all pixels of an infected image.

From Figure 3, one can find that the L1 norm of the 751st pixel is obviously different from others. With verification, we know that the 751st pixel is the pixel modified in the one-pixel attack. Also, to understand how the affected target label is related to the modified pixel, we calculate the L1 norm of the infected pixel to different labels. In Figure 4, we can find that the L1 norm to the airplane is lower than that to the other labels. Thus, our approach can also determine which label is the target label.

The average detection success rate of our trigger detection method is 9.1%.

5.4. Performance of Candidate Detection. In our candidate detection method, the initial number of candidate solutions and the number of produced candidate solutions are set to be 400, i.e., $N_{\text{ini}} = N_c = 400$; the maximum number of iterations is $T_{\max} = 100$; the scale parameter is $F = 0.5$; and the threshold for the probability of being classified to the target label is $p_{\text{th}} = 90\%$. To eliminate the influence of random variables in our Algorithm 1, we run the experiment 5 times with the fixed parameter settings and present the results in Table 5. Moreover, to investigate the impact of the size of candidate set, we also compare the detection success rates when C is set to be different values.

As shown in Figure 5, when $C = 1$, the success detection rate is 5.4% smaller than the success detection rate of our trigger detection method. Particularly, with $C = 1$, both the trigger and the candidate detection methods output one detected pixel but differ in their pixel selection schemes: (i) in out trigger detection, the trigger pixel is an optimal solution of trigger identification problem as well as has a smallest value of the L1 form, while (ii) in our candidate detection, the only one candidate victim pixel is randomly selected by the differential evolution-based heuristic algorithm. From the definition of the trigger pixel in this paper, the probability of the trigger pixel being modified in a one-pixel attack is larger than that of other pixels. Therefore, the trigger

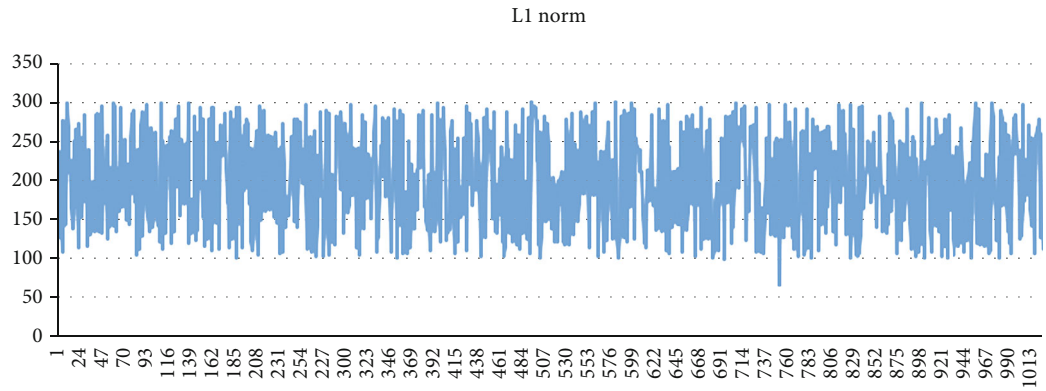

Figure 3: L1 norm of an infected image to label airplane.

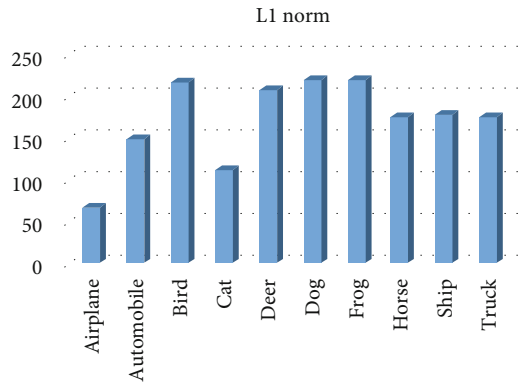

Figure 4: L1 norm of an infected image to airplane label.

Table 5: Detection success rate of candidate detection.

Experiment	1	2	3	4	5
Success rate (%)	20.4	24.3	30.1	21.9	26.7

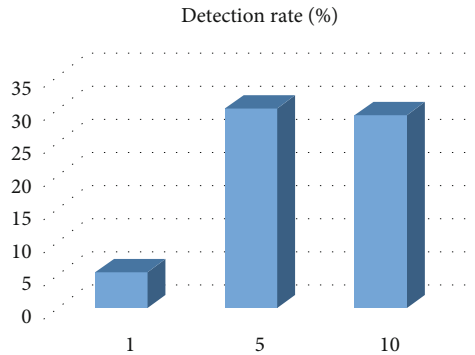

Figure 5: Impacts of the number of output candidates.

detection method outperforms the candidate detection method when $C = 1$, confirming that the idea of finding the trigger pixel to detect the one-pixel attack is solid.

When C is increased from 1 to 5, the detection success rate grows from 5.4% to 30.1%. The main reason is that with a larger number of output candidate pixels, more possible modified pixels can be examined, and the probability of finding the actual modified pixel is increased.

However, when C is increased from 5 to 10, the detection success rate nearly remains the same (in Figure 5, the subtle difference of the detection success rate results from the randomness of the DE algorithm). This illustrates that only increasing the number of candidate pixels may not always enhance the detection success rate. In summary, for the detection success rate, the marginal benefit of enlarging the number of candidate pixels is diminishing, and thus, setting an appropriate value to C can help effectively and efficiently detect the one-pixel attack (e.g., $C = 5$ in our experiments).

6. Conclusion and Future Work

This paper proposes two novel methods, i.e., the trigger detection method and the candidate detection method, to detect a one-pixel attack that is one of the most concealed attack models. The trigger detection method gives the exact pixel that may be modified by the one-pixel attack; the candidate detection method outputs a set of pixels that may be changed in the one-pixel attack. Via extensive real-data experiments, the effectiveness of our two methods can be confirmed; in particular, the detection success rate of our candidate detection can achieve 30.1%.

As a preliminary exploration of the one-pixel attack detection, in this paper, we consider that all the images are attacked and the detection is thus implemented on a dataset full of modified images. In our future work, we will carry out further research activities along two directions: (i) attempting to distinguish between the benign images and the attacked images in the presence of the one-pixel attack and (ii) mitigating the impact of the one-pixel attack by enhancing the resistance to adversarial samples in DNNs.

Data Availability

The data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the National Science Foundation of the U.S. (1704287, 1829674, 1912753, and 2011845).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, Bellevue, Washington, USA, 2011.

[3] A. Kendall and Y. Gal, *What uncertainties do we need in Bayesian deep learning for computer vision?*, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 2017.

[4] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: a simple deep learning baseline for image classification?," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[5] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[6] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[7] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer science & business media, 2013.

[9] D. D. Lewis, "Naive (Bayes) at forty: the independence assumption in information retrieval," in *European conference on machine learning*, pp. 4–15, Springer, 1998.

[10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, Saarbrucken, Germany, March 2016.

[11] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[12] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[13] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, "Adgan: protect your location privacy in camera data of auto-driving vehicles," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[14] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 668–677, Beijing, China, November 2019.

[15] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[16] B. Wang, Y. Yao, S. Shan et al., "Neural cleanse: identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723, San Francisco, CA, USA, May 2019.

[17] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, Boston, MA, USA, 2015.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, Las Vegas, NV, USA, 2016.

[19] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, Honolulu, HI, USA, 2017.

[20] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, Abu Dhabi, United Arab Emirates, April 2017.

[21] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56, San Francisco, CA, USA, May 2018.

[22] I. Ben-Gal, "Outlier detection," in *Data mining and knowledge discovery handbook*, pp. 131–146, Springer, 2005.

[23] S. Wu, G. Li, L. Deng et al., "$L1$-norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2019.