Defense Mechanism against Adversarial Attacks Based on Chaotic Map Encryption

Ping Wang¹, Qianyu Wang², Yuting Zhang³ and Yifan Wu⁴

¹Chongqing University of Posts and Telecommunications, Chongqing, China
²Chongqing Nankai Middle School, Chongqing, China
³China Mobile Chengdu Institute of Research and Development, Chengdu, China
⁴*Corresponding author e-mail: S190231088@stu.cqupt.edu.cn

Abstract. During recent years, image classification through DNN has been applied to various fields, including payment security and image search. DNN in image classification is effective and convenient, yet susceptible to perturbations: non-targeted and targeted adversarial attacks against neural networks, such as FGSM and BIM respectively, exert modifications that are unrecognizable to naked eyes to image inputs, and will probably result in wrong classifications. To ensure the degree of safety of DNN image classification, researchers have been dedicated to the study of defense mechanisms to diminish or even eliminate the effects brought by adversarial attacks. Our proposed approach, aims at increasing the classifier’s resistance to perturbations by adding a pseudo-random matrix key generated by Logistic Chaos. Our defense mechanism with Logistic Chaos-generated secret random key utilized 1 key with mere 3 elements and is of high generality. We show empirically that our approach is efficient against most attacks.

Keywords: Chaotic Map Encryption, Adversarial Attacks, Defense, Secret Random Key

1. Introduction
The deep neural network underwent numerous breakthroughs in the last decade, thanks to the endeavor and contributions of scientists. Because of the supreme accuracy which can be achieved by trained DNN through machine learning, deep neural network has been widely used in various artificial intelligence applications such as speech recognition, image recognition and point cloud recognition. However, DNN remains vulnerable to adversarial attacks which perturb the original samples and trigger DNN to make invalid responses.

In the recent years, as the deep neural network (DNN) has been widely applied in more and more security-sensitive and trust-sensitive areas, the study of security of DNN is becoming increasingly important.

Scientists have proposed various defense strategies against adversarial attacks, in general, the commonly used defense is divided into the following four categories:(1) Defense via retraining;(2) Defense via detection and rejection;(3) Defense via input pre-processing;(4) Defense via regeneration.
In this paper we will propose an improved defense mechanism, which belongs to defense via input pre-processing, for the DNN classifier proposed by Olga Taran, Shideh Rezaeifar and Slava Voloshynovskiy [1] by pre-processing original samples through direct addition of a secret random matrix key generated by Gaussian noise, aiming at increasing the classifier’s resistance to added attacking perturbations. The size of the secret random key matrix is equal to the size of an individual sample in the training dataset. Our improvement is to generate the pseudo-random matrix key by iterating Logistic chaos instead of by Gaussian noise. This improvement reduces the size of original key space from the size of a training sample to only two elements: the number of iterations \( n \), and the parameter \( \mu \), in Logistic chaos. By the second Kerckhoffs’s cryptographic principle that the less secret key the system contains, the higher the security of the system [2], reduced key’s size contributes to higher safety in our proposed defense mechanism.

We will verify the effectiveness of the new defense strategy on two standard data sets: MNIST and Fashion-MNIST. We will use FGSM, PGD and BIM adversarial attacks to test the reliability of proposed defense strategy.

The main contributions of this paper are as follows:

— Summarize and analyze the existing attack methods and defense strategies
— Propose a new chaotic encryption defense method based on cryptographic principles
— The experiment proves that this defense method that uses chaotic mapping to generate random numbers as a key has the performance to resist adversarial samples

The remainder of this paper consists of the following parts: The second part briefly introduces the principle of chaotic mapping, and summarizes the general classification of existing attacks and defense methods; the third part explains the main defense methods we propose Thought; the fourth part shows the experimental results and analysis results of our defense method; the fifth part summarizes the full text.

2. Background

2.1 Logistic Map
Pseudorandom number generation of piecewise logistic map. The piecewise logistic map (PLM), pseudorandom number generator (PRNG) proposed by Yong Wang et al, has mediocre ergodicity, uneven density probability and high efficiency which makes PLM an ideal PRNG.

The logistic map is a discrete dynamical system defined by

\[
x_{n+1} = \mu x_n (1 - x_n)
\]

where \( x_0 \in (0,1) \) is the state value and \( \mu \) is the control factor. When \( \mu \in [3.57, 4] \) the logistic map is chaotic [3].

2.2 Scenarios of Adversarial Attacks
Based on the attackers’ knowledge, the general cases of adversarial attacks can be grouped into three scenarios:

1) White-box scenario: The structure, parameters and training datasets of the defense system are all transparent and available to attackers.

2) Grey-box scenario: The structure, parameters and training datasets of the defense system are all transparent and available to attackers, while attackers have no access to the defense mechanism parameters [1]

3) Black-box scenario: The attackers do not have any information of the defense system, such as the structure, parameters, and training datasets.

Based on the aims of attacks, we summarize the adversarial attacks into two groups:

1) Targeted adversarial attacks: The attackers intentionally induce the result of DNNs to a targeted result.
2) Non-targeted adversarial attacks: The attackers aim to amplify the prediction error of DNNs without inducing a specific outcome [4]. This group of attacks aim to challenge merely the reliability of DNNs.

Based on the principles lying behind, current existing adversarial attacks on DNN classifiers can be categorized into Gradient based attacks and Non-gradient based attacks.

1) Gradient based attacks

Gradient based attacks generate a perturbation vector by slightly modifying the image to the back-propagation algorithm—which is widely used in DNN training—to cause wrong classification. They consider the classifier parameters as constants and the inputs as variables, therefore able to acquire the corresponding gradient for each element of the input.

As Ian J. Goodfellow et al. have noted, Fast Sign Gradient Method (FGSM) [5] is the fastest among all the gradient based attacks with relatively little cost; Basic Iterative Method (BIM) [6] makes simple improvements based on FGSM, utilizing the identical step to FGSM repeatedly, with smaller step size, and clipping the pixel value of each intermediate result; Projected Gradient Descent (PGD) [7] attack, as a multi-step variant of FGSM, achieves a higher accuracy. Based on enhanced PGD optimization, Yingpeng Deng et al. proposed UPGD [8], a new generation algorithm for generating universal adversarial attacks, which shows significant advantages in obtaining higher deception rates and lower accuracy. Even in a small training set, the algorithm also has good cross model versatility.

2) Gradient free attacks

One Pixel Attack [9], though not particularly more robust than other attacks, may indirectly increase the system response time; Zeroth Order Optimization (ZOO) [10] based black-box attacks, are capable of causing perturbations to DNNs without training any substitute model as an attack surrogate.

2.3 Defense Strategies

2.3.1 Defense via adversarial retraining. Defense via adversarial retraining is a robust generalization method [11]. In this mechanism, samples modified by adversarial attack method are mixed into the original training datasets. This mechanism is not adaptive to different types of adversarial attacks [11], which means that this defense system can only defense against attack methods that had been mixed into the training datasets. In addition, the accuracy of original model may be reduced after retraining. Adversarial training is a heuristic approach, which has no formal guarantee in convergence and robustness[12,13].

2.3.2 Defense via input monitoring. Input monitoring generally focuses on the classifying the input data as either original data or attacked data. This can be achieved by (a) adding an external augmented subnetwork of binary classification which classifies each input as attacked or un-attacked; in adaptive ML model assurance presented in [14], an external module called robust redundancy is proposed to resist potential hostile attacks and keep the trained ML model intact[11]; (b) feature squeezing which compares the model classification accuracy of outcomes between feature squeezed input and original input [15,16].

2.3.3 Defense via input pre-processing. Modifications exerted on input images by adversarial attacks can be removed by pre-processing defense mechanism. Bling Pre-processing (BP) [17] defense uses a combination of pre-processing layers, which has high robustness;

2.3.4 Defense via regeneration. This method is based recovering modified date to original clean data via regeneration. Research by Tejas Borkar et al.18 proposed a novel selective feature regeneration approach, which can effectively defend against universal perturbations and can significantly improve DNN adversarial robustness by shielding noise in some specific DNN activation [18].
3. Proposed Approach

Based on the fundamental principles of cryptography, we propose an adversarial attack defense mechanism with input images encrypted by Logistic Chaos and output by classifiers.

![Diagram of the proposed defense method](image)

**Fig 1.** The architecture of the proposed defense method: the attacker can only observe the input and output, not the internal state of the secret part.

The figure 1 is the basic scheme of defense against adversarial attacks by Logistic Chaos encryption (DLCE) mechanism. The mechanism begins by inputting image $x \in R^{NxC}$ into the Chaotic Map Encryption module (CME), which is encrypted based on a secret key $k \in K$. After the above encryption, state of input images become unknown. The unknown-state images will then be inputted into a DNN classifier, which is a LeNet-5 neural network model. The structures of both CME and classifier are disclosed to the public. The transformation of CME module will be invertible and undifferentiable. Finally, a classification $\hat{c}$ will be the output of each input $x \in R^{NxC}$, determined by classifier. We will offer a more specified output classification in later section 4.3.

According to various types of defense strategies presented in section 2.3, DLCE is under the type of defense via input pre-processing. However, encrypting by secret key, neither filtration nor elimination of perturbations is required in the CME module of DLCE.

It is supposed that the attacker knows information other than the key, including the structure of the classifier and the security module used. similarly, our approach follows the principle of key sharing in both the training and prediction phases, exposing all algorithmic details except the key, which is known only by the defender, confidential to the attacker, who can not access the internal variables of the defense structure. In other words, the input and output of CMA can not be accessed, and the output of CMA is the input of classifier. Furthermore, the attacker can not access the input of classifier, but can only observe the input and output of the whole architecture, as well as the structure of the classifier and CMA, thus a secret part is formed, lest the features of the training data be learned, or the gradient information of the system is obtained by BPDA[19] techniques.

We use a key-based security module CME. In general, CME be integrated with various kinds of transformations, such as simple permutations. However, unlike the previous encryption form, this module is to transform the input matrix into data uncorrelated by generating keys by Iterating Logistic mapping. This secret key is unknown to the attacker. Therefore, it creates the information advantage of the defender over the attacker. The size of the key space we choose is no longer equal to the size of the input signal, and chaotic map encryption uses less key space to ensure higher security. This improvement reduces the size of the original key space from the size of the training sample to the two elements of iteration number $n$ and parameter $\mu$ in Logistic chaos. logistic mapping is a discrete dynamic system, with pseudo-random number generated by the $x_{n+1} = \mu x_n (1 - x_n)$ piecewise logistic map, where $x_0 \in (0, 1)$ is the state value and $\mu$ is the control parameter. Moreover, when $\mu \in [3.57, 4]$, the logistic mapping is chaotic [3]. The number of iterations is chosen to be $n = 500$ with initial value $x_0 = 0.2, \mu = 4$. In the fourth part, we will explain the experimental effect of using chaotic encryption.

4. Evaluation
4.1 Datasets
In order to test the generalization ability of the proposed method, we have tested the effectiveness of the proposed method on three different data sets. We used a simple MNIST handwritten digit recognition data [20] set, which contains ten categories, including 60,000 training images and 10,000 test images, each of which is a 28*28 grayscale image. We also use the Fashion-MNIST dataset, which contains ten categories, including 60,000 training images and 10,000 test images, each of which is a 28*28 grayscale image. Examples of images in each data set are shown in Figure 2.

![Image of MNIST and Fashion-MNIST](image)

**Fig 2.** The example of raw images of each category from MNIST(first row) and Fashion MNIST(second row)

To clarify, in our experiment, 55,000 images in the training set of the MNIST and Fashion-MNIST datasets are used for training, and 5000 images are used for verification. Since the selected adversarial attack is generated very slowly, both data sets are tested with the first 1000 images in the testing set only.

4.2 Adversarial Attacks Details
We use FGSM, BIM, and PGD as adversarial attacks to test the capability of the proposed method. The details of employed adversarial attacks in the virtual experiment are explained as below.

4.2.1 FGSM. FGSM[5], namely Fast Gradient Sign Attack, is a typical white-box adversarial attack. It adds perturbation to the gradient direction of the input vector along the error function of the output category and the target category to obtain an adversarial perturbation, and then the adversarial perturbation is added to the original sample to generate an adversarial sample. This method can generate adversarial samples quickly and at low cost through one-step iteration:

$$\eta = \epsilon \text{sign}(\nabla_x \mathcal{J}(\theta, x, y))$$  \hspace{1cm} (2)

$$x^{adv} = x + \eta \text{sign}(\nabla_x \mathcal{J}(\theta, x, y))$$  \hspace{1cm} (3)

In the formula above, $\eta$ is the added disturbance; $x^{adv}$ is the adversarial sample; $\mathcal{J}(\cdot)$ is the loss function of the DNN classifier; $\theta$ is the parameter of the model, and $\epsilon$ is used to represent the magnitude of the disturbance.

4.2.2 BIM. BIM (Basic Iterative Methods) [6] uses multiple smaller input change parameters along the gradient direction to perform iterative attacks instead of generating adversarial disturbances in one step like FGSM:

$$x^{adv}_0 = X$$  \hspace{1cm} (4)

$$x^{adv}_N + 1 = \text{Clip}_{X}(x^{adv}_N + \alpha \text{sign}(\nabla_x \mathcal{J}(x^{adv}_N, y_{true}))$$  \hspace{1cm} (5)

In the formulas above, $\text{Clip}_{X}(x', (x, y, z)) = \min(255, x(x', y, z) + \epsilon, \max(0, x(x, y, z) - \epsilon, x'))$ means that $x$, $y$, $z$, and $X$ are in the 3-D space, which means that the image’s width, height, number of pathways.

4.2.3 PGD. Raised by Madry and her colleagues, PGD(Project Gradient Descent) is a typical first-order iterative attack, which can also be called K-FGSM, where K represents the number of iterations. The PGD algorithm first performs a random initialization search within the allowable range (spherical
noise area) near the original image, and then performs multiple iterations to generate adversarial samples:

\[ X_{0}^{\text{adv}} = X \]  
\[ X_{t+1}^{\text{adv}} = \text{Clip}(X + \alpha \text{sign}(\nabla x J(X_{t}^{\text{adv}}, y)) \]  

4.3 Result and Discussion

In order to get the best experimental results, we have chosen the three existing attack codes, which are all from CleverHans based on Tensorflow. On the two data sets, the parameters used by the attack method and the corresponding examples of the generated adversarial samples are shown in Table 1 and Table 2.

| Attack | MNIST | FASHION-MNIST |
|--------|-------|---------------|
|        | Step_size | Iterations | Step_size | Iterations |
| FGSM   | 0.25   | 0.09         |           |             |
| BIM    | 0.3    | 0.01         | 100       | 0.1         | 0.01        | 100         |
| PGD    | 0.3    | 2/255        | 40        | 0.1         | 2/255       | 40          |

Table 1. Parameters of adversarial attacks

| Attack | MNIST | Fashion-MNIST |
|--------|-------|---------------|
| FGSM   | ![Image](mnist_fgsample.png) | ![Image](fashion_fgsample.png) |
| BIM    | ![Image](mnist_bimsample.png) | ![Image](fashion_bimsample.png) |
| PGD    | ![Image](mnist_pgsample.png) | ![Image](fashion_pgsample.png) |

Table 2. Adversarial Examples generated by FGSM, BIM and PGD attacks on MNIST and Fashion-MNIST

Under the premise of the attack and the data set, we used the LeNet-5 based network structure to classify all attacks. The classifier structure is shown in Table 3; the classifier training parameters are shown in Table 4.

| Layer | MNIST Model | Fashion-MNIST Model |
|-------|-------------|---------------------|
| Convolution | 3x3x6 | 5x5x6 |
| Max Pooling | 2x2 | 2x2 |
The results are shown in Table 4, where "Original classifier" represents the original DNN classifier without any defense measures, and its parameters are shown in the table; "Classifier with Logistic chaos" represents the DNN classifier on which the Logistic chaos defense module is used.

According to the experimental results shown in Table 5, on the MNIST data set, the classification error rate of the original classifier without any defense measures is about 81%-98% for the adversarial samples generated by the three adversarial attack methods--FGSM, BIM, and PGD; the classification error of the classifier using the Logistic chaos defense module is about 7%-20%. Similarly, on the Fashion-MNIST dataset, the classification error rate of the original classifier without any defense measures is about 58%-74% higher than that of the classifier using the Logistic chaos defense module. Therefore, the defense method of using Logistic chaos to generate the key greatly reduces the classification error rate of the classifier. The experimental results obtained are sufficient to prove the effectiveness of the defense method based on chaotic map encryption, which is a highly potential method to resist adversarial attacks.

Table 4. Training parameters of DNN classifiers

| Training parameters | MNIST Model | Fashion-MNIST Model |
|---------------------|-------------|---------------------|
| Optimizer           | Adam        | Adam                |
| Learn rate          | 1e-3        | 1e-3                |
| Epoch               | 50          | 50                  |
| Batch size          | 128         | 128                 |

Table 5. Classification error (%) on the first 1000 test examples

| Attack | Original classifier | Classifier with Logistic chaos |
|--------|---------------------|--------------------------------|
|        | MNIST               |                                |
| FGSM   | 0.813               | 0.208                          |
| BIM    | 0.988               | 0.128                          |
| PGD    | 0.926               | 0.078                          |
|        | Fashion-MNIST       |                                |
| FGSM   | 0.745               | 0.195                          |
| BIM    | 0.933               | 0.169                          |
| PGD    | 0.936               | 0.173                          |

In the experiment, in order to verify the effectiveness of Logistic chaotic map encryption against adversarial sample attacks, we assume that the length of the key is equal to the length of the input image, making the key long enough to deal with brute force attacks. Taking into account security issues, the attacker may attack the model by brute force to unlock the key. Therefore, we ensure that the internal variables are unknown to the attacker, and only the input and output can be observed. This avoids the attacker from obtaining key information, thereby ensuring system security.

5. Conclusions
In the paper, in view of the less efficiency of deep neural networks to adversarial examples, we studied the existing adversarial attack methods and defense methods, and made a general classification and
summary. On this basis, we propose a new defense mechanism based on chaotic map encryption to resist adversarial samples. This defense mechanism is mainly aimed at the existing white-box attack scenarios and two data sets. Experiments have proved that this method can obtain high classification accuracy and has advanced performance in defense against adversarial samples.

Our defense method improves the classification accuracy. In future work, we will start to study the adaptability of chaotic map encryption defense in black box scenarios, and try our best to expand its application to other data sets.

References
[1] Massey, J.L.: Cryptography: Fundamentals and applications. In: Copies of transparencies, Advanced Technology Seminars. Volume 109. (1993) 119
[2] A pseudorandom number generator based on piecewise logistic map by Yong Wang. Zhaolong Liu. Jianbin Ma. Haiyuan He
[3] The Threat of Adversarial Attacks Against Machine Learning in Network Security: A Survey Olakunle Ibitoye, Rana Abou-Khamis, Ashraf Matrawy and M. Omair Shafiq School of Information Technology, Carleton University, Ottawa, Canada
[4] Towards a Robust and Trustworthy Machine Learning System Development
[5] Goodfellow I J, Shlens J, Szegedy C. Explaining and Harnessing Adversarial Examples. Computer Science, 2014.
[6] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. 2016.
[7] Madry A, Makelov A, Schmidt L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks. 2017.
[8] Moosavi-Dezfooli S M, Fawzi A, Fawzi O, et al. Universal adversarial perturbations. IEEE, 2017.
[9] Su J, Vargas D V, Kouichi S. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 2017.
[10] Chen P Y, Zhang H, Sharma Y, et al. ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models. ACM, 2017.
[11] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition 84 (2018), 317 – 331.
[12] Dasgupta P, Collins J B. A Survey of Game Theoretic Approaches for Adversarial Machine Learning in Cybersecurity Tasks. AI Magazine, 2019, 40(2);31-43.
[13] Wang X, Li J, Kuang X, et al. The security of machine learning in an adversarial setting: A survey. Journal of Parallel and Distributed Computing, 2019.
[14] in adaptive ML model assurance presented in
[15] Xu W, D Evans, Qi Y. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks[C]// Network and Distributed System Security Symposium. 2017.
[16] Rouani B D, Samragh M, Javidi T, et al. Safe Machine Learning and Defeating Adversarial Attacks. IEEE Security & Privacy, 2019, 17(2):31-38.
[17] Rakin A S, He Z, Gong B, et al. Blind Pre-Processing: A Robust Defense Method Against Adversarial Examples. 2018.
[18] Borkar T S, Karam L J. DeepCorrect: Correcting DNN Models Against Image Distortions. IEEE Transactions on Image Processing, 2017, PP(99).
[19] Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
[20] Lecun Y, Cortes C. The mnist database of handwritten digits. http://www.research.att.com/~yann/ocr/mnist/, 2010.