2023 8th International Conference on Computer and Communication Systems (ICCCS) | 978-1-6654-5612-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCS57501.2023.10151180

# A Convenient Deep Learning Model Attack and Defense Evaluation Analysis Platform

Bingjun He
*30th Research Institute of China Electronic Technology Group Corporation*
Chengdu, China
line 5: hbj_ump45@126.com

Yu Luo
*Sichuan Expressway Construction & Development Group Co.，Ltd*
Chengdu, China

Yifan Wang
*30th Research Institute of China Electronic Technology Group Corporation)*
Chengdu, China

Zhi Sun
*30th Research Institute of China Electronic Technology Group Corporation*
Chengdu, China

Jianfeng Chen
*30th Research Institute of China Electronic Technology Group Corporation*
Chengdu, China

Ye Han
*30th Research Institute of China Electronic Technology Group Corporation*
Chengdu, China

***Abstract*—Researchers have been studying in recent years how to improve the security of deep learning models to resist adversarial attacks. However, attack and defense algorithms are generally targeted, a unified, comprehensive, efficient, and convenient analysis platform is needed for deep learning model security evaluation. Deep learning model security evaluation faces the following challenges:(i) The difference of deep learning frameworks used to generate models causes inconvenience in model security evaluation. (ii) Most of the attack and defense algorithms are not universally applicable. A comprehensive measurement baseline is needed to comprehensively evaluate various attack and defense algorithms. (iii) A set of quantitative metric system needs to be proposed. In this paper, a convenient deep learning model attack and defense evaluation analysis platform is designed to automate the evaluate process. Users need only a small amount of configuration to complete the evaluation of various attack and defense algorithms. The platform integrates a variety of black-box attack algorithms and white-box attack algorithms and 6 defense algorithms. Besides, a system of metrics is constructed, covering imperceptibility, robustness, attack efficiency, etc. Based on this platform, we evaluate the performance of various algorithms on common image classification models. Compared with other platforms, this platform is efficient and convenient for evaluation, and has a wide range of evaluation. It has the ability of comparing and analyzing models generated by various frameworks and is useful for research on the security of deep learning models.**

***Keywords*—*deep learning model security, adversarial attack, adversarial defense, automate evaluation***

## I. Introduction

The progress of science and technology has brought artificial intelligence to new prosperity. For example, deep learning is widely used in various fields, such as image processing, audio recognition, automatic driving, cybersecurity etc. While it has become a booster of social development and scientific and technological progress, deep learning is also facing various threats. The vast majority of existing artificial intelligence systems only consider how to improve their intelligence level, while the security of the systems is often ignored. Because of the vulnerability of neural networks [1], deep learning models used in different fields, such as image recognition [2], text recognition[3] speech recognition [4], are easy to be breached by adversarial attacks. Thus, it is important to study the security of artificial intelligence systematically, to reduce the risk of applying deep learning models in fields that have rigorous security standards. In order to ensure the security of data sharing, some scholars present Calculate data logistic regression[5] and Adaptive Federated Learning algorithm[6].

In order to counter this type of attack that causes neural networks to misjudge by interfering with normal input, security researchers have conducted a lot of relevant research on the subject. Many adversarial libraries do not adopt the most advanced models such as FoolBox, Cleverhans [7], AdverTorch [8], AdvBox [9] when implementing new attack algorithms, which will lead to evaluation errors. With the most advanced attack algorithms and models, these methods need to be quantitatively analyzed. However, the lack of a unified evaluation method and quantitative indicators often leads to misjudgment by safety researchers. DEEPSEC [10] constructs a unified analysis platform, which has a large number of attack and defense algorithms and has various single evaluation metrics for attack and defense. DEEPSEC has evaluated a large number of white-box adversarial attack algorithms. However, DEEPSEC is pointed out that the added disturbance is too large, and the performance of the worst case in the attack is not considered. Francesco Croce built RobustBench [11] platform to solve overestimating the robustness of the model, and it

This work is supported by National Key Research and Development Program (2020AAA0107800), Sichuan Youth Science and Technology Innovation Research Team, (Grant No.2020JDTD0034) and National Key RD Program of China (No.2019YFB2101700).

evaluated the models in more than 30 papers uniformly through Auto-attack. Auto-attack[12] proposed an improved PGD attack algorithm and combined it with existing attacks to form an attack scheme without parameter adjustment. Experiments on more than 50 models have achieved good results.

In the long-term research of attack algorithms and defense algorithms, attack and defense algorithms are constantly upgraded. Therefore, a platform that can comprehensively evaluate attack and defense algorithms is needed in order to promote the related research of adversarial samples and the research of the security of the model itself. The platform needs a variety of attack defense algorithms. It supports different types of model evaluations and has various evaluation metrics, scalability, and merging characteristics.

Contributions:

(1) We automate the whole evaluation process. Security researchers only need to select the items to be evaluated and configure them in a small amount, such as attack methods, defense methods, model robustness, etc. After uploading models and datasets, the platform will automatically generate evaluation scripts and will automatically execute the final output evaluation results. This is different from platforms such as RobustBench, ART[13], DEEPSEC, etc., which need manual coding to complete evaluation.

(2) The platform supports the evaluation of black/white box algorithms, target/non-target attack. And the platform integrates 7 advanced black-box attack algorithms and 15 white-box attack algorithms.

(3) The platform employs the containerized design and implementation, which makes it easy for researchers to integrate new algorithms and metrics. The platform supports users to upload deep learning models and datasets. This facilitates learning research for security researchers and facilitates the application and promotion of the platform. Previous platforms such as Cleverhans and Foolbox lack such ability.

(4) To evaluate the effectiveness of attack algorithms and defense algorithms, a set of computable metrics is designed for image scene, which can be comprehensively evaluated from various aspects, and some meaningful conclusions are drawn according to these metrics.

(5) Different from other evaluation platforms that only support a few deep learning frameworks, this platform supports more frameworks by using container virtualization technology and ONNIX model automatic transformation technology Tensorflow, PyTorch, PMML, ONNX and other deep learning frameworks are supported.

## II. System Design and Implementation

### A. Platform Structure

To automate the evaluation process. The platform generates evaluation scripts based on the input configuration. First the platform preprocesses the user's configuration. Then locate specific keyword parameters such as: attack type, learning rate, number of iterations, etc. Finally, the evaluation scripts are generated based on the prepared code templates and rules. The platform will take the default values set in advance for parameters that are not configured by the user. Both rules and code templates are scalable, for easy addition of new rules and maintenance. The scheduling center calls other modules through API to complete the evaluation according to the evaluation script.

In order to perform evaluation tasks efficiently and quickly, the system adopts containerized design. Based on Docker engine, the intelligent algorithm library, attack algorithm library, detection algorithm library and defense algorithm library and database used for testing are all saved in mirror format, which will be pulled up into containers during the task center scheduling. The running environment required by the model to be evaluated is also container environment, platform to provide PMML, ONNX, PyTorch, Tensorflow and other machine learning model running framework and environment. These containers are centrally managed by the task scheduling center. Task scheduling center is the core part of the whole platform, including container management module, task scheduling center module, model security evaluation algorithm module and resource monitoring module. The relationship among modules in the platform is shown in Figure 1.

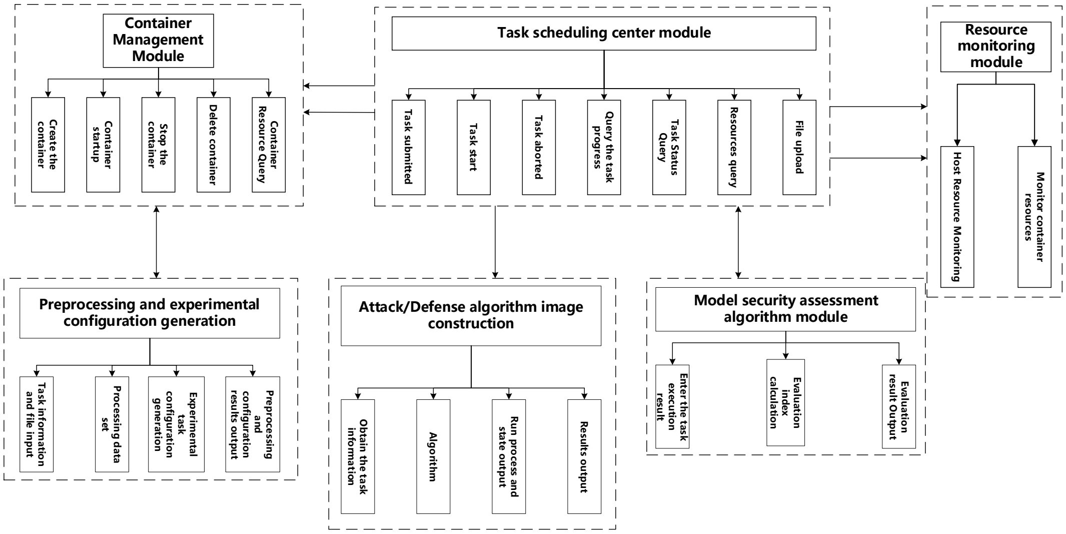
Container Management Module
Create the container
Container startup
Stop the container
Delete container
Container Resource Query
Task scheduling center module
Task submitted
Task start
Task aborted
Query the task progress
Task Status Query
Resources query
File upload
Resource monitoring module
Host Resource Monitoring
Monitor container resources
Preprocessing and experimental configuration generation
Task information and file input
Processing data set
Experimental task configuration generation
Preprocessing and configuration results output
Attack/Defense algorithm image construction
Obtain the task information
Algorithm
Run process and state output
Results output
Model security assessment algorithm module
Enter the task execution result
Evaluation index calculation
Evaluation result Output

Fig. 1. The relationship among modules.

During the evaluation, the task scheduling center created, started, stopped, and deleted the interface management container by calling the container. When multiple tasks are performed, the scheduling center will start multiple containers to evaluate at the same time to improve efficiency of evaluation. The task scheduling center is also responsible for processing the input model and data, preprocessing the input model and data, including unifying the model format, standardizing the size of the input data, etc.

The resource monitoring module is responsible for the resource monitoring of physical hosts and each running container. It periodically reports the resource occupation to the task scheduling center module through the structure so that the scheduling module can allocate containers according to resources. The container management module itself is responsible for the start and stop of containers, mirror management and other functions. The model security evaluation module is responsible for the overall analysis and evaluation of the model according to the results of attack and defense evaluations.

### *B. Attack and Defense Library Construction*

The platform supports various black-box and white-box attack evaluations on models. The characteristic of black box-attack is that the attacker knows nothing about the parameters and network structure of the target model, which brings difficulties in generating adversarial samples. The white-box attack is one in which the attacker knows the structure and parameters of the target model. At present, white-box attack accounts for a majority part of attack algorithms. We implemented the black-box attack by converting the model to the ONNX format and using the ONNX Runtime deployment model. White-box attacks are implemented by designing a multi-framework running deployment container environment.

The platform has 7 black-box attack algorithms:
HopSkipJumpAttack[14], Boundary Attack[15], Zoo Attack[16], SimBA[17], Pixel Attack[18], Hclu[19]and Wasserstein [20].

The platform has 15 white-box attack algorithms:

FGSM[21], RFGSM[22], PGD[23], APGD[23], BIM[25], DEEPFOOL[26], LLC[27], RLLC[27], C&W2[29], C&Wi[30], ATNs[31], JSMA[32], OM[33], UAP[34], EAD[35].

In the development of the attack and defense of the adversarial samples, both attack and defense are making continuous progress. At present, the defense algorithms can be roughly divided into five categories. The platform considers three categories of counter training, gradient shielding and input transformation. Six defense algorithms were integrated, which were EAT[36], NAT[37], DD[38], IGR[39], RT[40] and RC[41].

### *C. Utility Metrics*

The merits of an attack algorithm should be measured not only in terms of its success rate, but also in terms of other aspects of the attack algorithm. For example, an attack algorithm that changes the original sample so much that it is visible to the human eye would be disqualified even if it misclassifies the classifier. The following metrics are proposed to evaluate the attack algorithms in terms of misclassification, imperceptibility and robustness.

Misclassification:

MR misclassification rate: Misclassification rate is a very core evaluation metric for attack algorithms. The level of MR directly reflects the effectiveness of the attack. For a non-target attack, MR means that the sample is wrongly classified into any wrong category, while for a target attack, MR means that the sample is wrongly classified into a specified category. UA is a non-target attack. TA is the target attack. The formulas are as follows:

$$MR_{UA} = \frac{1}{N}\sum_{i=1}^{N} count\left(F(X_i^a \neq y_i)\right) \tag{1}$$

$$MR_{TA} = \frac{1}{N}\sum_{i=1}^{N} count\left(F(X_i^a = y_i^*)\right) \tag{2}$$

ACAC Average Confidence of Adversarial Class: An assessment of the reliability of misclassification. The formula is:

$$\frac{1}{n}\sum_{i=1}^{n} P(X_i^a)_{y_i} \tag{3}$$

Imperceptibility:

ALDp Average Lp Distortion: The p-order norm distance between the generated adversarial sample and the original sample, and the L0 norm counts how many pixels in the adversarial sample are changed. The L2 norm represents the Euclidean distance between the generated adversarial sample and the original sample. The smaller this value is, the more difficult it is for human eyes to detect the difference. L The infinite norm represents the value that changes the most against the elements in the sample. The formulas are as follows:

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\parallel X_i^a - X_i \parallel_p}{\parallel X_i \parallel} \tag{4}$$

Robustness:

NTE Noise Tolerance Estimation[42]:Calculate the difference between the probability of misclassification and the maximum probability of other categories. Noise and deviation cannot be avoided adversarial sample attack, and an excellent attack should not be easily detected by human eyes. The formula is:

$$\frac{1}{n}\sum_{i=1}^{n} \left[P(X_i^a)_{F(X_i^a)} - max\left\{P(X_i^a)_j\right\}\right] \quad j \in \{1, \ldots, k\} \tag{5}$$

AQT Average number of queries: This is an important indicator for the evaluation of black-box attacks and can be used in multiple scenarios. When black-box attacks generate adversarial samples, they need to query the target model to obtain partial information. The level of AQT reflects the complexity and effectiveness of an attack algorithm.

Defense evaluation metric:

The defense algorithm can be measured from two aspects: (1) the resistance of the defense algorithm in the face of adversarial sample attack, which is the stability of the direct counterreaction model. (2) The ability of the defended model to retain the function of the previous model. If the functionality is not fully preserved then there is little point in defending. The platform will measure the defense approach based on several metrics:

AD accuracy difference: The purpose of attack is to produce misclassification, so the purpose of defense is to accurately classify. The model with enhanced defense should be accurate enough to judge the correct example. As a core indicator like MR, AD needs to be considered first. MD indicates the enhanced defense model of the original model M. AD represents the output of softmax layer corresponding to MD. The formula of AD is:

$$AD = Acc(M_D, T) - Acc(M，T) \tag{6}$$

Acc(M,T) represents the accuracy of model M on T datasets.

COS Classification Output Stability : JS divergence was used to measure the similarity of M and MD in output probability, and the divergence of M and MD outputs was averaged over the correctly classified examples. COS reflects the stability of the classification output between M and MD. The formula is:

$$COS = \frac{1}{n}\sum_{i=1}^{n} JSD\big(P(X_i) \parallel P_D(X_i)\big) \tag{7}$$

The n<N is the number of correct classification, and JSD is the function of JS divergence.

### D. Evaluation Process

First, users upload models and datasets and perform some configuration. Then, the platform generates evaluation scripts based on the models and items to be evaluated, and selects attack/defense algorithms from a library of attack/defense algorithms. At the same time, the scheduling center starts the runtime environment container for the evaluation model. And according to the evaluation script to select the mirror from the attack algorithm library to generate adversarial samples. When all the attack processes are finished, the evaluation module collects and analyzes the attack results. Then, the model is strengthened by applying defense algorithms selected from the defense algorithm library. After the reinforcement, the model is saved and compared with the original model. Finally, various attack algorithms are called again to generate adversarial samples on the enhanced model and carry out comprehensive evaluation. The model evaluation flow is shown in Figure 2.

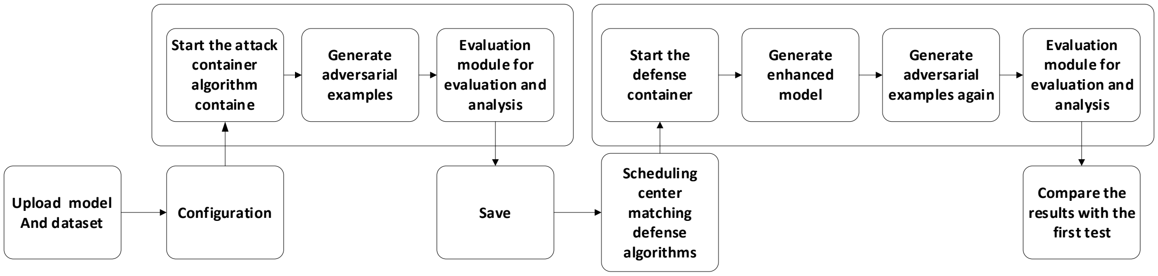
Start the attack container algorithm containe
Generate adversarial examples
Evaluation module for evaluation and analysis
Start the defense container
Generate enhanced model
Generate adversarial examples again
Evaluation module for evaluation and analysis
Upload model And dataset
Configuration
Save
Scheduling center matching defense algorithms
Compare the results with the first test

Fig. 2. Evaluation process chart.

This evaluation process is done automatically. The platform will generate evaluation scripts according to the uploaded model and the configuration of the user, automatically complete the evaluation of the model according to the evaluation scripts, and output evaluation results. The system automatically matches scenarios based on the configurations entered by users. After evaluating a sufficient number of models, the system prioritizes and recommends more effective attack and defense algorithms based on the previous evaluations.

The platform supports multiple machine learning framework such as Tensorflow, PyTorch, ONNX, PMML, and so on. The corresponding container environment can be launched when needed. At present, many platforms only support a specific type of input samples. Our platform is designed to support multiple types of input samples. Currently, image samples is supported. The modular and containerized design of the whole platform is also conducive to the expansion of functions in the future.

The platform provides a comprehensive and unified system. In order to better evaluate the performance of various attack defense algorithms, a comprehensive metric system is established for both image scene. The system objectively and comprehensively evaluates various methods to avoid misjudgment of the same method or model caused by different evaluation methods. This is helpful for security researchers to objectively evaluate existing methods and models, to fully understand their safety performance, and to facilitate subsequent research.

## III. Evaluations

The platform supports the evaluation of black-box and white-box attack algorithms. And it supports the evaluation of deep learning models with different frameworks.

In this section, we completed evaluation of attack, evaluation of defense and evaluation of defense against attack in the image scene in turn. Finally, we completed the analysis of the experimental results.

The hardware configuration: CPU Intel(R) Xeon(R) CPU E5-2609v4@1.70GHz, 4*GPU GTX1080Ti(11GB).

### A. Evaluation of Attack

To test the evaluation function of the platform, first we upload a model and configured the parameters, the platform took 7 seconds to generate the evaluation script, then selected 5 black-box attacks and 5 white-box attacks from attack library, task scheduling stared 11 containers of the center at the same time. The platform took 174 seconds to generate adversarial samples, finally evaluation module took 27 seconds to complete the evaluation.

Datasets: MNIST of the platform was used for the datasets, and 1260 images were randomly selected as the test set, and each figure was handwritten numbers of 28*28.

Parameter setting: Keep the general parameters consistent, and other parameters consistent with the original work.

Evaluation model: The experimental model adopts an RNN image classification model of Tensorflow framework.

The experimental results are shown in Table 2. Most of the attacks were successful in producing adversarial samples. In terms of misclassification, the MR of white-box attacks are higher than that of black-box on the whole. Although the MR of HSJA and BA is 100%, the MR of other black-box attacks is very low, and the ACAC of black-box attack is very low, indicating that the attack samples generated by black-box attack are mostly of low confidence, which also reflects that black-box attack is more difficult than white-box attack.

The ACAC and ACTC of black-box and white-box attacks against the sample will not be high at the same time, because the sum of the class probabilities is 1. In HSJA and BA, the ACAC and ACTC of DF and CWI are 0.74, 0.01, 0.82 and 0.37, respectively. In FGSM and APGD, the ACTC is 0 when the ACAC reaches 1. This indicates that when the ACAC is less than 1, the ACTC will be high or low.

In terms of imperceptibility. It can be seen that the ASS of black-box attack is lower than that of white-box attack, and the adversarial sample generated by black-box attack is more deviated from the original sample. Black-box attack needs to add more disturbance to successfully generate adversarial sample.

In terms of robustness. Generated from the white-box attack adversarial sample is better than the black-box in NTE generated attacked samples, this attack algorithm has much to do, and the black-box attacks can only according to the input and output information to calculate the gradient, this leads to generate the adversarial samples of tolerance to noise is very small, namely these generated robustness adversarial samples is very small, When defending against black-box attack, special preprocessing module can be added to the input of the model to resist black-box attack.

TABLE I. EFFECTIVENESS EVALUATION RESULTS OF ADVERSARIAL SAMPLE ATTACKS

| **Attack** | | | **MR** | **ACAC** | **ACTC** | **ALDP** | | | **ASS** | **NTE** | **AQT** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **BA/WA** | **UA/TA** | **Attacks** | | | | **L0** | **L2** | **L∞** | | | |
| BA | UA | HSJA | 100% | 0.23 | 0.21 | 4.73 | 0.16 | 0.3 | 0.21 | 0.02 | 2171 |
| | | BA | 100% | 0.23 | 0.23 | 6.6 | 0.14 | 0.24 | 0.19 | 0.01 | 3548 |
| | | Zoo | 0.00% | 0.00 | 0.00 | - | - | - | - | - | - |
| | | SimBA | 12.76% | 0.17 | 0.51 | 0.53 | 2.81 | 0.10 | 0.13 | 0.04 | 5741 |
| | | Square | 29.73% | 0.37 | 0.35 | 4.11 | 0.27 | 0.13 | 0.14 | 0.02 | 3844 |
| WA | | FGSM | 79.82% | 1.00 | 0.00 | 0.66 | 3.09 | 0.30 | 0.33 | 1.00 | - |
| | | APGD | 99.99% | 1.00 | 0.00 | 4.09 | 0.62 | 0.30 | 0.42 | 1.00 | |
| | | DF | 94.11% | 0.74 | 0.01 | 3.94 | 0.06 | 0.14 | 0.37 | 0.05 | |
| | TA | CW2 | 83.23% | 0.47 | 0.36 | 5.754 | 0.19 | 0.57 | 1.00 | 1.00 | |
| | | CWi | 75.58% | 0.82 | 0.37 | 0.22 | 5.43 | 0.28 | 0.31 | 0.69 | |

## B. *Evaluation of Defense*

Firstly, the model to be evaluated was uploaded to the platform for defense configuration. The platform took 11 seconds to generate the evaluation script. Meanwhile, the task scheduling center started 7 containers, and it took 21 minutes to generate the reinforcement model.

Datasets: The data set adopts the CIFAR10 data set of the platform, whose images are 32*32 color images. We randomly selected 1200 images as a test set. To verify the effect of the platform on the color data set.

Parameter setting: Keep the general parameters consistent, and other parameters consistent with the original work.

Evaluation model: The evaluation model adopts a ResNET-20 image classification model based on PyTorch framework.

The evaluation results are shown in Table 3. By comparing the original work of these defense methods, we can find that the defense algorithm of the platform is effective. Each algorithm can successfully reinforce the model. Although the accuracy of some models is lost, such as DD, EAT and IGR, the loss caused by the defense algorithm may not be worth mentioning in the face of adversarial sample attack. To verify this, we are going to do defenses against attacks evaluations. It can be seen from the COS results of each algorithm that the COS has a similar trend, which may be related to the prediction confidence variance.

TABLE II. EFFECTIVENESS EVALUATION RESULTS OF DEFENSE METHODS

| Datasets | Defense | Accuracy | AD | COS |
|---|---|---|---|---|
| CIFAR10 | Nan | 73.81% | - | - |
| | DD | 63.26% | -10.55% | 0.046 |
| | NAT | 74.46% | 0.65% | 0.038 |
| | EAT | 72.74% | -1.07% | 0.087 |
| | IGR | 69.10% | -4.71% | 0.183 |
| | RC | 18.93% | -54.88% | - |
| | RT | 70.47% | -3.34% | 0.015 |

## C. *Against Attacks*

In order to accurately understand how well the hardened model can resist various attack algorithms, we uploaded the model and configured the parameters. The platform generated evaluation scripts in 14 seconds, selected 5 white-box attack algorithms from the attack library and 6 defense algorithms from the defense library for evaluation of defenses against attacks., and started 12 containers in the task scheduling center, taking 26 minutes to complete all evaluations. The returned evaluation results can analyze the effectiveness of various defense algorithms against different attacks.

Datasets: 1200 32*32 color images randomly selected from CIFAR10 data set.

Parameter setting: Keep the general parameters consistent, and other parameters consistent with the original work.

Evaluation model: The evaluation model adopts a ResNET-20 image classification model based on PyTorch framework.

The evaluation results are shown in the following table. Most of the defense algorithms are still able to defend the attack to some extent. Through the joint evaluation of several attack and defense algorithms, we find that gradient shielding (regularization) defense algorithms such as DD and IGR are very effective against white-box attack based on gradient class. DD defense has obvious effect in the face of FGSM and BIM, and its MR decreases from 93.81% and 100% to 57.24% and 61.3%. The defense performance of EAT, NAT and other adversarial samples training is very different, and EAT cannot effectively reduce MR under these attack algorithms. We analyze that EAT uses the adversarial samples generated by RFGSM for pre-training, and this defense algorithm may not be universal. Since RT and RC are input-conversion defense, we input the adversarial samples generated by various attacks into the strengthened model. These attack samples are 100% misclassified by the original model, so the higher ACC represents the better defense effect. RT has obvious defensive effect on BIM and RLLC, ACC increased from 0% to 41.38% and 42.12%, respectively.

TABLE III. EFFECTIVENESS EVALUATION OF DEFENSE METHODS AGAINST FGSM

| Defenses | ACC | MR | ACAC | ACTC | ALDP | | | ASS | NTE |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | L0 | L2 | L∞ | | |
| Nan | - | 100% | 0.99 | 0.00 | 0.99 | 10.06 | 0.30 | 0.41 | 0.99 |
| DD | - | 87.4% | 1.00 | 0.00 | 0.99 | 10.35 | 0.30 | 0.38 | 1.00 |
| EAT | - | 100% | 0.95 | 0.00 | 0.99 | 10.83 | 0.30 | 0.38 | 0.91 |
| NAT | - | 100% | 0.98 | 0.00 | 0.99 | 10.31 | 0.30 | 0.40 | 0.99 |
| IGR | - | 78.27% | 0.55 | 0.09 | 0.98 | 10.34 | 0.30 | 0.39 | 0.36 |
| RC | 0.11% | - | - | - | - | - | - | - | - |
| RT | 19.82% | - | - | - | - | - | - | | - |

TABLE IV. EFFECTIVENESS EVALUATION OF DEFENSE METHODS AGAINST PGD

| Defenses | ACC | MR | ACAC | ACTC | ALDP | | | ASS | NTE |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | L0 | L2 | L∞ | | |
| Nan | - | 100% | 0.99 | 0.00 | 0.99 | 10.06 | 0.30 | 0.41 | 0.99 |
| DD | - | 87.4% | 1.00 | 0.00 | 0.99 | 10.35 | 0.30 | 0.38 | 1.00 |
| EAT | - | 100% | 0.95 | 0.00 | 0.99 | 10.83 | 0.30 | 0.38 | 0.91 |
| NAT | - | 100% | 0.98 | 0.00 | 0.99 | 10.31 | 0.30 | 0.40 | 0.99 |
| IGR | - | 78.27% | 0.55 | 0.09 | 0.98 | 10.34 | 0.30 | 0.39 | 0.36 |
| RC | 0.11% | - | - | - | - | - | - | - | - |
| RT | 19.82% | - | - | - | - | - | - | | - |

TABLE V. EFFECTIVENESS EVALUATION OF DEFENSE METHODS AGAINST BIM

| Defenses | ACC | MR | ACAC | ACTC | ALDP | | | ASS | NTE |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | L0 | L2 | L∞ | | |
| Nan | - | 100% | 0.99 | 0.00 | 0.99 | 2.93 | 0.10 | 0.82 | 0.99 |
| DD | - | 61.3% | 1.00 | 0.00 | 0.99 | 3.28 | 0.10 | 0.80 | 1.00 |
| EAT | - | 100% | 0.83 | 0.00 | 0.99 | 3.93 | 0.10 | 0.76 | 0.72 |
| NAT | - | 100% | 0.99 | 0.00 | 0.99 | 3.04 | 0.10 | 0.82 | 0.98 |
| IGR | - | 55.83% | 0.45 | 0.17 | 0.99 | 3.03 | 0.10 | 0.82 | 0.22 |
| RC | 0.00% | - | - | - | - | - | - | - | - |
| RT | 41.38% | - | - | - | - | - | - | | - |

TABLE VI. EFFECTIVENESS EVALUATION OF DEFENSE METHODS AGAINST LLC

| Defenses | ACC | MR | ACAC | ACTC | ALDP | | | ASS | NTE |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | L0 | L2 | L∞ | | |
| Nan | - | 19.70% | 0.82 | 0.01 | 0.99 | 15.44 | 0.30 | 0.22 | 0.73 |
| DD | - | 11.36% | 1.00 | 0.00 | 0.99 | 15.68 | 0.30 | 0.22 | 1.00 |
| EAT | - | 12.84% | 0.45 | 0.04 | 0.99 | 15.47 | 0.30 | 0.26 | 0.18 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| NAT | - | 7.4% | 0.48 | 0.08 | 0.99 | 15.35 | 0.30 | 0.26 | 0.30 |
| IGR | - | 11.11% | 0.47 | 0.04 | 0.99 | 15.63 | 0.30 | 0.22 | 0.23 |
| RC | 0.00% | - | - | - | - | - | - | - | - |
| RT | 29.74% | - | - | - | - | - | - | | - |

TABLE VII. Effectiveness Evaluation of Defense Methods Against RLLC

| **Defenses** | **ACC** | **MR** | **ACAC** | **ACTC** | **ALDP** | | | **ASS** | **NTE** |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | L0 | L2 | L∞ | | |
| Nan | - | 37.07% | 0.72 | 0.01 | 0.53 | 3.91 | 0.10 | 0.77 | 0.58 |
| DD | - | 19.41% | 0.99 | 0.00 | 0.53 | 3.93 | 0.10 | 0.74 | 0.99 |
| EAT | - | 16.74% | 0.45 | 0.05 | 0.51 | 3.87 | 0.10 | 0.78 | 0.24 |
| NAT | - | 7.63% | 0.52 | 0.06 | 0.53 | 3.93 | 0.10 | 0.78 | 0.34 |
| IGR | - | 1.5% | 0.37 | 0.08 | 0.52 | 3.94 | 0.10 | 0.78 | 0.17 |
| RC | 0.00% | - | - | - | - | - | - | - | - |
| RT | 42.12% | - | - | - | - | - | - | | - |

## IV. Future Work and Conclusion

Firstly, the most advanced 7 black-box and 15 white-box attack methods and 6 defense methods are integrated into the image scenes evaluation, but some advanced algorithms are still not integrated. More scenes such as audio and video will be considered in future work. In the future, attack and defense algorithms will continue to be expanded and updated. The modular design and implementation of the platform are beneficial to integrate new attack and defense algorithms.

Secondly, the attack and defense in image scene is supported. It is evaluated in terms of error rate, imperceptibility, and robustness, which is an evaluation of attack and defense algorithms and lacks an evaluation of model stability. In the future, we will consider other test algorithms to evaluate the model itself, such as the metamorphosis test, split fuzzy test and so on.

In addition, the models generated using TensorFlow and PyTorch are evaluated on the Mnist and CIFAR10 datasets, respectively. Based on the experiments, we prove that the platform support various frameworks. The reason why the evaluation results of individual algorithms are not ideal may be that this algorithm needs skills to set parameters that are not suitable for platform automated evaluation. This kind of problem will be fully studied in the future.

Finally, the platform requires users to manually select models and various algorithms and set parameters. In the next work, the platform will be optimized to help users configure fewer parameters to complete the evaluation.

In summary, a convenient analysis platform for the evaluation of deep learning models for attack and defense is established. The platform is also evaluated. The platform can automate the evaluation task after a small amount of configuration. The platform supports image data types, supports the evaluation of black-box and white-box attack algorithms, has a variety of attack and defense algorithms, and has a complete evaluation system. Users can evaluate deep learning models under various frameworks. Experiments show that the platform is convenient and efficient. The platform is useful for security research of deep learning models.

## References

[1] Christian Szegedy,Wojciech Zaremba,Ilya Sutskever,Joan Bruna,Dumitru Erhan,Ian J. Goodfellow,Rob Fergus. “Intriguing properties of neural networks,” Computer Science.2013.

[2] Anh Mai Nguyen,Jason Yosinski,Jeff Clune, “Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images,”[J]. CoRR,2014,abs/1412.1897.

[3] Jinfeng Li,Shouling Ji,Tianyu Du,Bo Li,Ting Wang, “TextBugger: Generating Adversarial Text Against Real-world Applications.” 2018.

[4] Xiaolei Liu,Kun Wan,Yufei Ding,Xiaosong Zhang,Qingxin Zhu, “Weighted-Sampling Audio Adversarial Example Attack.” Proceedings of the AAAI Conference on Artificial Intelligence,2020,34(04).

[5] Saeed Samet, "Privacy-Preserving Logistic Regression," Vol.6,No.3, pp. 88-95, August, 2015. doi: 10.12720/jait.6.3.88-95.

[6] Alessandro Giuseppi, Lucrezia Della Torre, Danilo Menegatti, Francesco Delli Priscoli, Antonio Pietrabissa, and Cecilia Poli, "An Adaptive Model Averaging Procedure for Federated Learning (AdaFed)," Journal of Advances in Information Technology, Vol. 13, No. 6, pp. 539-548, December 2022.

[7] Ian J. Goodfellow,Nicolas Papernot,Patrick D, “McDaniel. cleverhans v0.1: an adversarial machine learning library.” CoRR,2016, abs/1610.00768.

[8] Gavin Weiguang Ding,Luyu Wang,Xiaomeng Jin, “advertorch v0.1: An Adversarial Robustness Toolbox based on PyTorch.” CoRR,2019, abs/1902.07623.

[9] Dou Goodman and Hao Xin and Wang Yang and Wu Yuesheng and Xiong Junfeng and Zhang Huan, “Advbox: a toolbox to generate adversarial examples that fool neural networks.”2020 arXiv 2001.05574.

[10] Ling X, Ji S , Zou J , et al, “DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model.” 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.

[11] Croce F, Andriushchenko M, Sehwag V, et al. “RobustBench: a standardized adversarial robustness benchmark.” 2020.

[12] Croce F , Hein M , “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks.” 2020.

[13] Nicolae M I , Sinn M , Tran M N , et al. “Adversarial Robustness Toolbox v1.0.0.” 2018.

[14] Chen J , Jordan M I , Wainwright M J , “HopSkipJumpAttack: A Query-Efficient Decision-Based Attack.” 2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020.

[15] Brendel W , Rauber J , Bethge M . “Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models” 2017.

[16] 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.

[17] Guo C , Gardner J R , You Y , et al. “Simple Black-box Adversarial Attacks.” 2019.

[18] Kotyan S , Vargas D V , “Adversarial Robustness Assessment: Why both $L_0$ and $L_\\infty$ Attacks Are Necessary.” 2019.

[19] Grosse K , Pfaff D , Smith M T , et al, “The Limitations of Model Uncertainty in Adversarial Settings.” 2018.

[20] Wong E , Schmidt F R , Kolter J Z , “Wasserstein Adversarial Examples via Projected Sinkhorn Iterations.” 2019.

[21] Goodfellow I J , Shlens J , Szegedy C , “Explaining and Harnessing Adversarial Examples.” Computer Science, 2014.

[22] F Tramèr, Kurakin A , Papernot N , et al, “Ensemble Adversarial Training: Attacks and Defenses.” 2017.

[23] Madry A , Makelov A , Schmidt L , et al. “Towards Deep Learning Models Resistant to Adversarial Attacks.” 2017.

[24] Croce F , Hein M , “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks.” 2020.

[25] Kurakin A , Goodfellow I , Bengio S . “Adversarial examples in the physical world.” 2016.

[26] Moosavi-Dezfooli S M , Fawzi A , Frossard P , “DeepFool: a simple and accurate method to fool deep neural networks.” Computer Vision & Pattern Recognition. IEEE, 2016.

[27] Kurakin A , Goodfellow I , Bengio S , “Adversarial examples in the physical world.” 2016.

[28] F Tramèr, Kurakin A , Papernot N , et al, “Ensemble Adversarial Training: Attacks and Defenses.” 2017.

[29] Carlini N , Wagner D , “Towards Evaluating the Robustness of Neural Networks.”2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017.

[30] Carlini N , Wagner D , “Towards Evaluating the Robustness of Neural Networks.” 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017.

[31] Baluja S , Fischer I , “Adversarial Transformation Networks: Learning to Generate Adversarial Examples.” 2017.

[32] Papernot N , Mcdaniel P , Jha S , et al, “The Limitations of Deep Learning in Adversarial Settings.” 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2015.

[33] He W , Bo L , Song D , “Decision Boundary Analysis of Adversarial Examples.” 2018.

[34] Moosavi-Dezfooli S M , Fawzi A , Fawzi O , et al, “Universal adversarial perturbations.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[35] Chen P Y , Sharma Y , Zhang H , et al, “EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples.” 2017.

[36] F Tramèr, Kurakin A , Papernot N , et al, “Ensemble Adversarial Training: Attacks and Defenses.” 2017.

[37] Kurakin A , Goodfellow I , Bengio S , “Adversarial Machine Learning at Scale.” arXiv, 2016.

[38] Papernot N , Mcdaniel P , Wu X , et al, “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks.” 2016 IEEE Symposium on Security and Privacy (SP). IEEE, 2016.

[39] Ross A S , Doshi-Velez F , “Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients.” 2017.

[40] Xie C , Wang J , Zhang Z , et al, “Mitigating adversarial effects through randomization.” 2017.

[41] Cao X , Gong N Z , “Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification.” 2017.

[42] Zhou W , Bovik A C , Sheikh H R , et al, “Image quality assessment: from error visibility to structural similarity.” IEEE Trans Image Process, 2004, 13(4).