IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2020, accepted June 7, 2020, date of publication June 11, 2020, date of current version June 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001730

SecureAS: A Vulnerability Assessment System for Deep Neural Network Based on Adversarial Examples

YAN CHU[1], XIAO YUE[2], QUAN WANG[1], AND ZHENGKUI WANG[3]
[1]College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China
[2]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China
[3]Singapore Institute of Technology, Singapore 138683

Corresponding author: Xiao Yue (yuexiao@zuel.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771155, in part by the China MOE Project of Humanities and Social Sciences under Grant 14YJC630181, in part by the Natural Science Foundation of Hubei Province under Grant 2017CFB592, in part by the Fundamental Research Funds for the Central Universities Zhongnan University of Economics and Law under Grant 2722020JCT032 and Grant 2722020PY047, in part by the Singapore MOE TIF under Grant MOE2017-TIF-1-G018, and in part by the SIT Ignition Grant under Grant R-MOE-E103-D004.

ABSTRACT Deep neural network (DNN) has been recently applied to many safety-critical environments. Unfortunately, recent research has proven that DNN can be vulnerable to well-designed examples, called adversarial examples. Adversarial examples can easily fool a well-performed deep learning model with little perturbations imperceptible to humans. In this paper, to tackle the DNN security issue, we propose a **M**odel **A**dversarial **S**core (MAS) index to evaluate the vulnerability of a deep neural network, and introduce a deep learning vulnerability assessment system (SecureAS) using adversarial samples to assess the vulnerability and risk of a trained DNN in a blackbox way. We also present two adversary algorithms (FGNM and PINM) that provide better adversary images with the similar attack effect compared to existing approaches like FGSM and BIM. Our experimental results confirm the effectiveness of MAS algorithm, SecureAS, FGNM and PINM.

INDEX TERMS Adversarial examples, adversarial attack, deep learning, vulnerability assessment.

I. INTRODUCTION

With the rapid progress and great success in a wide spectrum of applications, deep learning is being applied in many safety-critical environments, such as self-driving cars [1], facial recognition [2]–[4], malware detection [5], medical diagnostics [6], natural language processing [7]–[9] and image generation [10], etc. Recent research has demonstrated that existing DNNs can be vulnerable to maliciously well-designed inputs, called *adversarial examples* [11].

Adversarial examples are a kind of data samples well-designed by attackers to fool a deep learning model. These samples are normally obtained by adding a small interference over the original samples. They cannot be easily identified by humans, but may lead to the misclassification of the model.

Take the self-driving system as an example. The vehicles are normally empowered by deep learning models in their image recognition system to detect and identify all kinds of traffic signs and signal signs. During the model training, a malicious attacker can construct adversarial examples by injecting adversarial perturbation to the original traffic sign image. Though the adversarial examples look very similar to the original image by human eyes, it can lead to a wrong classification. This becomes dangerous while deploying the deep learning model in the self-driving vehicles for image recognition. Once the attacker slightly modifies the actual traffic sign on the road using the same perturbation as the adversarial examples, the vehicles will miss-classify the traffic sign into the wrong classification and incur serious traffic accidents. This attack does not need to change the deep learning model itself but only the samples to mislead the deep neural network classifier. Additionally, the same attack samples are effective for many different deep neural networks, which makes the problem more critical.

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng.

The existence of adversarial examples shows common security problem for deep learning models. The problem of adversarial examples is due to the limited representation accuracy of digital image itself and the fact that there are too many linear operations in the deep neural network. Therefore, the adversarial examples are universal to DNNs, and even can be considered as a basic characteristic of neural network. The existence of adversarial examples will not change essentially because of the structure and type of neural network itself. However, due to the differences in the structure of different neural networks, the preprocessing of the input image samples may be different, and the calculation amount and calculation method of the linear operation may be slightly different. Therefore, the robustness of different neural networks for various kinds of adversarial examples generation algorithm is different, that is, the security of different neural networks may be divergent.

Much research effort has been devoted to proposing the approaches of generating different kinds of adversarial examples. For example, one step gradient approximation methods (such as Fast Gradient Sign Method (FGSM) [12] and fast least possible class approximation method [13]) are proposed to construct the adversarial examples quickly by propagating the target model forward once and calculating the reverse gradient. L-BFGS and Adam have been used to solve the optimization problem of generating the adversarial examples [13].

Existing works need to evaluate the vulnerability of the target neural network according to whether the given neural network can correctly identify the categories of the adversarial examples or not, in a whitebox way. This method is not stable, and has high randomness. It becomes impractical for vulnerability evaluation in many confidentiality scenarios, as it is difficult to understand the inner structure of a deployed DNN. Therefore, this calls for a new approach of evaluating the vulnerability of a deployed neural network.

In this paper, we make the first attempt to address the question: *how to evaluate the vulnerability of a deployed neural network without knowing its inner structure?* Currently, there is no systematic and intuitionistic index to reflect the vulnerability of DNNs, and no standard system to evaluate the vulnerability of DNN remotely in a blackbox way. To address this issue, we propose an index, named Model Adversarial Score (MAS) to evaluate and quantify the vulnerability of DNNs. The output of the MAS index is a score which measures the vulnerability of a model, and can also be used to evaluate the attack effect of an adversarial examples generation algorithm. Furthermore, to evaluate the vulnerability of a deep learning model thoroughly, we present a deep learning security assessment system **SecureAS**, which evaluates the security of a deep neural network for image classification through adversarial examples, in a blackbox way. Using this system, users can quickly generate the adversarial examples of local sample images, use them to test the target model, and analyze the security level of the target model. In addition, we have also investigated the new approaches of generating better adversarial samples, such that the adversarial samples are more close to original samples with less noises.

Our major contributions are as follows:

1) We propose a Model Adversarial Score (MAS) index, a measure to evaluate the vulnerability of given deep neural networks. We also present one security assessment system, SecureAS based on MAS index to assess DNN models in a blackbox way. To the best of our knowledge, this is the first attempt to provide such an index with systematic and intuitionistic system solution in a blackbox way.
2) We provide two adversarial examples generation algorithms, namely Fast Gradient Norm Method (FGNM) and Peak Iteration Norm Method (PINM), which are able to generated higher quality of adversarial examples compared to existing algorithms Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM).
3) We conduct experiments to evaluate the system functions and verify the effectiveness of MAS index, FGNM and PINM.

The rest of the paper is organized as follows. Section II provides the introduction of the MAS index. In Section III, we introduce two improved adversarial example generation algorithms. Section IV presents the SecureAS system design. In Section V, we provide the SecureAS system interface/function testing and experimental evaluations on MAS index and the proposed adversarial generation algorithms. Section VI and Section VII show the related work and conclusion.

II. MODEL ADVERSARIAL SCORE

In this section, we present the proposed model adversarial score (MAS) index. We first provide the MAS definition and its calculation algorithm followed by the discussion on its design principle and its applications.

A. DEFINITION AND ALGORITHM

Considering different adversarial samples have been designed, MAS index aims to provide a standard measurement of evaluating the vulnerability of a given DNN.

For a pre-trained deep neural network model M, the sample is represented as (x,y), where $x \in X$, $y \in Y$, and X and Y are the data sample set and label/category set respectively. An adversarial example generation algorithm for M is represented as a_M, where $a_M \in A$ and A is the set of adversarial examples generation algorithms. Based on this, we provide the equation below:

$$\begin{aligned} M(x_c) &= y_c \\ x_a &= a_M(x_c), \\ M(x_a) &= y_a \end{aligned} \tag{1}$$

where x_c is the original sample of model M, x_a is the adversarial example generated on model M by the adversarial example

generation algorithm a_M for x_c, y_c is the prediction category of model M to sample x_c, and y_a is the prediction category of model M to sample x_a. Because the model can achieve high accuracy after pre-training, so in general, $y_c = y_{true}$, but the prediction results of the model for the adversarial examples are not necessarily the same as the real tags. When Equation 2 becomes true, it can be considered that the model M has security problems. The attack based on the adversarial examples can succeed, and the model will be cheated by the adversarial examples.

$$y_a = y_{false}, \tag{2}$$

where the y_{false} represents false category.

For a given dataset, the sample category space of model M is N, i.e., $\|Y\| = N$. Given a data sample x, we define p_n as the probability of category n in the prediction result of model M to sample x, that is, $p_n = P(y = y_n|x)$,where $\sum_{i=1}^{n} p_i = 1$. Meanwhile, $y = argmax(P)$, where P is the probability set of each category predicted by the model for the sample x. A good model M is able to predict the category correctly for one given sample. It is known that the DNN applies the probability to determine the correct category. A higher probability of a category indicates a higher chance of predicting the sample belonging to that category. With the adversarial samples, a secure model can still predict the adversarial samples into the correct category. Therefore, the more prediction results for the adversarial examples deviates from the correct category, the more vulnerable the model itself will be.

The main objective of MAS index is to calculate one vulnerability evaluation score which can be used to evaluate the vulnerability of the model for various adversarial examples. Recall that the output layer of a DNN model decides the final category based on the probability assigned to each category. Therefore, the prediction probability information of each category provides insight about how a model deals with an adversarial sample. For a robust DNN model, it should assign the biggest probability to the correct category. For the given real sample and its adversarial sample, the model may generate two different probabilities of each category. After ranking the categories based on their possibilities in descending order, the position (or index) of each category in the sorted array reflects the distance about how far they are from the predicted results. The ordering variance provide one indicator for the vulnerability of the model, which is also used to derive the MAS index. Algorithm 1 provides the detail procedure about the calculation of the MAS index. The inputs of the algorithm are the target neural network model M, the original sample image, the true classification of the original sample image, a set of the adversarial sample generation algorithms. The output is the MAS score of the model. The algorithm is to find the position or label corresponding to the largest element in the probabilities array. After sorting the array, it finds the sequence number of the element in the value.

Algorithm 1 Compute MAS

Input: Target neural network model M, original sample image $x_c \in X$, the true classification of the original sample image $y_{true} \in Y$, the adversarial sample generation algorithm for the target model $A = \{a_1, \cdots, a_n\}$
Output: Target Model Adversarial Vulnerability Score (MAS).
function Predict(x)
  $P \leftarrow M(x)$
  $y \leftarrow argmax(P)$
  $p_y \leftarrow P[y]$ **return** $y, p_y, sort(P)$
end function

$y_c, p_c, P_c \leftarrow P_{REDICT}(x_c)$
if $y_c \neq y_{true}$ **then return** Error
else
  for $a_i inA$ **do**
    $x_a \leftarrow a_i(x_c)$
    $y_a, p_a, P_a \leftarrow P_{REDICT}(x_a)$
    $I_a \leftarrow \text{Index}(p_a, P_c)$
    $I_c \leftarrow \text{Index}(p_c, P_a)$
    $MAS_i \leftarrow \frac{I_a + I_c}{2\|Y\|}$
  end for
  $MAS \leftarrow \frac{1}{n}\sum_i MAS_i$
  return MAS
end if

B. DISCUSSIONS ON THE USAGE OF MAS INDEX

The MAS index provides a model vulnerability evaluation score. As MAS index is designed by integrating different factors (e.g., samples, adversary generation algorithm, the DNN model), MAS can not only be used to evaluate the vulnerability of the model, but also be used as other indicators, such as a reference index to evaluate and measure the quality of a certain class of samples of the target model, the attack efficiency and the attack effect of a certain adversarial examples generation algorithm.

First, MAS index can be used to measure and evaluate the model security and vulnerability. Traditionally, the machine learning researchers and practioners mainly focus on the DNN model performance ignoring the security and vulnerability. With MAS, while they trying various DNN models, they can also measure the model vulnerability. This will enable the practitioners to identify the most optimal DNN model to use, and even to improve the model with new vulnerability concerns.

Second, MAS index can also be used to quantify the quality of adversarial samples generated by different adversarial sample generation algorithms and the attack effectiveness on the neural networks. With MAS index, after obtaining the characteristics and attack effectiveness of different adversarial sample generation algorithms, practitioner can select the most appropriate and efficient adversarial samples generation algorithm according to the actual situation of neural network model and samples in the attack and defense scene. This

allows practitioners to better attack and test the neural network model.

Lastly, the MAS index can be a reference for the quality of selected training samples. For a target neural network, given the current training samples, if the model can correctly classify the original training samples but not the adversarial samples, this may potentially indicate the model needs to be further trained with more or better quality of training samples to make the model robust enough.

III. IMPROVED ADVERSARIAL EXAMPLES GENERATION ALGORITHM

In this section, we introduce our two new adversary algorithms: Fast Gradient Norm Method (FGNM) and Peak Iteration Norm Method (PINM).

A. IMAGE SIMILARITY

A good adversarial example generation algorithm tries to make minimal changes to the original image, but can still fool a neural network. To determine the similarity and difference between two images (the original and adversarial samples), we adopt two algorithms: Peak Signal to Noise Ratio (PSNR) [14] and Structure Similarity [15]. These two algorithms calculate the similarity of image from the distribution of image noise, so as to evaluate the quality of the algorithm.

PSNR is a common algorithm to calculate image similarity, which is generally used to measure the quality of compressed image. Its calculation is based on the mean square error of the pixel value of the original image and the processed image, and the calculation equation is shown in Equation 3.

$$\begin{aligned} MSE &= \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|I_{i,j}-K_{i,j}\right\|_2, \\ PSNR &= 10lg\left(\frac{MAX_I^2}{MSE}\right) = 20lg\left(\frac{MAX_I}{\sqrt{MSE}}\right), \end{aligned} \tag{3}$$

where MAX_I is the maximum possible pixel value of the image. Generally, when samples are represented with B bits per sample, MAX_I is $2^B - 1$.

B. ALGORITHM IMPROVEMENT PRINCIPLE

In the process of generating adversarial examples, the generation algorithm based on gradient often needs to calculate the adversarial perturbation according to the gradient. The adversarial perturbation is a series of special image noise points. The adversarial perturbation and the original sample image are superposed to get the adversarial examples. Therefore, there is no difference between the adversarial image and the original sample image in the main content, image structure and other aspects. Comparing with the original sample image, the adversarial example has some more noise points. In this paper, the improvement goal of the algorithm is to make the adversarial image more similar to the original image, so the noise added to the adversarial image should be minimized and difficult to find.

The principle of adversarial examples generation algorithm is very similar to the training process of neural network, and also uses the back propagation algorithm. However, different from the neural network training process, in the process of generating adversarial examples, it is necessary to keep the parameters of the model itself unchanged and optimize the input samples of the model. While the neural network training process will not change the input samples, it is to optimize the parameters of the model. The generation algorithm aims to reduce the difference between the generated image and the original image as much as possible on the premise of ensuring the attack effect. In other words, the added adversarial noise on the original image is as small as possible. In the process of neural network training, in order to avoid overtraining and overfitting, regular terms will be added to the loss function to constrain the parameters. Similar methods, i.e., L1 regularization, L2 regularization, can be used to constrain the scale of adversarial noise in the adversarial examples generation algorithm.

C. IMPROVED GENERATION ALGORITHM

Recent works have proposed two different ways of generating the adversaries: Fast Gradient Sign Method (FGSM) [12] and Basic Iterative Method (BIM) [13]. FGSM aims to craft adversarial perturbations using the derivative of the model's loss function with respect to the input feature vector. BIM is the iterative version of FGSM by applying the adversarial noise many time iteratively. In this paper, we further improve them by adopting a new adversarial perturbation calculation method.

In the adversarial examples generation algorithm based on gradient, the adversarial perturbation ρ is calculated according to gradient. To improve the quality of the adversarial samples, we use L2 regulization to constrain the gradient and reduce the scale of adversarial noise. The new adversarial perturbation calculation method is provided in Equation 4.

$$\rho = \varepsilon\frac{\frac{\partial}{\partial x}J(\theta;x,y)}{\left\|\frac{\partial}{\partial x}J(\theta;x,y)\right\|_2}, \tag{4}$$

where x is the input image sample; $J()$ is the loss function; y is the real category of the sample; θ and ε are the model parameter and the scaling vector respectively.

Based on this improved adversarial perturbation calculation method, we design two adversarial generation algorithms: Fast Gradient Norm Method (FGNM) based on the fast gradient truncation method as in FGSM and Peak Iteration Norm Method (PINM) based on iterative gradient truncation method as in BIM.

The calculation equation of FGNM is shown in Equation 5

$$\begin{aligned} \rho &= \varepsilon\frac{\frac{\partial}{\partial x_c}J(\theta;x_c,y_{true})}{\left\|\frac{\partial}{\partial x_c}J(\theta;x_c,y_{true})\right\|_2}, \\ x_a &= clip(x_c+\rho). \end{aligned} \tag{5}$$

Algorithm 2 Adversarial Example Generation

Input: Target neural network model M, loss function of $MJ(\theta; x, y)$, original sample image $x_c \in X$, image size m*n, category of original image $y_{true} \in Y$, scaling vector ε, maximum iterations N, ultimate image similarity K, image pixel level MAX_x
Output: adversarial example
$n \leftarrow 0$
$x^0 \leftarrow x_c$
while $n < N$ **do**

$$\rho^n = \varepsilon \frac{\frac{\partial}{\partial x^n} J(\theta; x^n, y_{true})}{\left\| \frac{\partial}{\partial x^n} J(\theta; x^n, y_{true}) \right\|_2}$$
$$x^{n+1} = clip(x^n + \rho^n)$$
$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\| I_{i,j} - K_{i,j} \right\|_2$$
$$PSNR = 10lg\left(\frac{MAX_I^2}{MSE}\right) = 20lg\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

if $M(x^{n+1}) \neq M(x) \& PSNR < K$ **then**
$x_a \leftarrow x^{n+1}$
return x_a
end if
$n \leftarrow n + 1$
end while
return None

The calculation equation of PINM is designed by using the new adversarial perturbation calculation method and introducing the PSNR. The algorithm can generate the adversarial examples by Equation 6

$$\rho^n = \varepsilon \frac{\frac{\partial}{\partial x^n} J(\theta; x^n, y_{true})}{\left\| \frac{\partial}{\partial x^n} J(\theta; x^n, y_{true}) \right\|_2}, \quad x^{n+1} = clip(x^n + \rho^n). \tag{6}$$

After introducing the algorithm of PSNR to calculate image similarity, Algorithm 2 introduces the main algorithm flow. The input of the algorithm includes the target model, the loss function of the target neural network, the original sample image, the real category and size of the original image, the scaling vector and iteration times, etc. The output of the algorithm is the adversarial sample. The algorithm first calculates the adversarial perturbation according to the gradient of loss function, then generates the adversarial examples according to the adversarial perturbation. After which, it calculates the PSNR of the adversarial image and the original sample image, and the prediction category of the model for the adversarial examples. If the prediction category is not the same as the original image sample, and the PSNR is less than the threshold value, it outputs the adversarial image. And the attack is successful. Otherwise, it enters the next iteration. If the number of iterations reaches the preset number and still fails to generate the required adversarial examples, the attack fails.

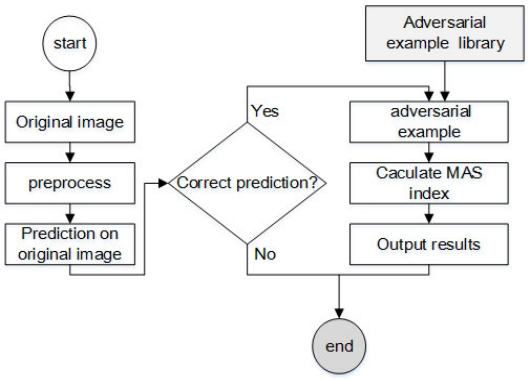

FIGURE 1. Components and work flow overview of SecureAS system.

IV. SecureAS

In this section, we will introduce one deep learning vulnerability assessment system, SecureAS to analyze the security of deep learning model for image classification. Meanwhile, SecureAS is used to analyze and verify the effect of the improved algorithm.

A. ARCHITECTURE OF SecureAS

The deep learning vulnerability assessment system includes the following functions:

(1) The system can call a variety of local pre-trained DNN models to recognise and classify any picture and visualize the output.

(2) The system can test the remote cloud image recognition server and use the predefined interface to call the remote cloud image recognition server to recognize and classify the local image.

(3) The system can generate the local model's adversarial examples based on the original image, and attack and cheat the target model with the generated adversarial image.

(4) The system can evaluate and analyze the model's vulnerability and risk, and output the analysis results visually.

SecureAS consists of five modules: image preprocessing module, local image recognition engine, remote identification calling engine, adversarial examples generation module and vulnerability assessment visualization module. We will introduce the details of each module in the rest of this section.

To be intuitive, Figure 1 indicates a simple workflow about how the assessment is done. The image preprocessing is first carried out after which it is fed to the model to generate the predicted result. In addition, the adversarial samples are generated by using the adversarial examples generation algorithm after which they are fed to the model to generate the predicted result. In the end, we evaluate the model security and risk by using the MAS index algorithm followed by intuitive visualizations for users.

B. IMAGE PREPROCESSING

The current secureAS system is designed to support the deep learning models for image recognition and image

classification in the field of computer vision. The data samples are all digital images.

The neural network model needs unified standard input. But in the actual use of the model, the input image is often inconsistent with the format required by the model. The aim of the image preprocessing is to transfer the original input data into the required data format. The preprocessing process of digital images mainly includes three aspects: image color channel splitting, image size scaling and image pixel value standardization.

When the input image size does not fulfill the requirement of the model, the image needs to be preprocessed. If the original image is larger than the model required size, the image need to be cropped. For image recognition and classification tasks, the main content of the default recognition and classification is located in the middle of the image. So, normally, the new image is cut out in the middle of the original image. When the original image is smaller, bilinear filtering is needed to interpolate the image pixels and expand the image.

The samples processed in this paper are color images. For the color image, we need to consider its color space. The color space is a description and representation of the color of pixels in a digital image. In essence, the color model is the elaboration of coordinate system and subspace. Each color in the system is represented by a single point. Each color space is represented by three or four channels. RGB color space is the most commonly used color space in video and image. It has three channels, namely red, green and blue. For a color image, each pixel has three values of R, G and B. But for the deep learning model such as convolutional neural network, the color channels of image need to be split, and each channel needs to be processed separately.

For RGB color space digital image, the value range of pixel value of each channel is between [0, 255]. But for MXNet deep learning model [16], the value range of each pixel of the input image matrix is between [0, 1], so it is necessary to standardize the pixel value of the input digital image. In this paper, the pixel standardization method is provided in Equation 7.

$$x_{i,j} = \frac{x_{i,j}}{127.5} - 1. \tag{7}$$

Based on Equation 7, we process each pixel of the image channel into a multi-dimentional matrix with a specified requirement for the model, which can be directly fed into the DNN model. This is also the basic sample template for generating adversarial examples.

C. LOCAL IMAGE RECOGNITION ENGINE

The local image recognition engine consists of various DNNs that are under attack testing. We aim to build SecureAS as an independent platform to evaluate the vulnerability of a DNN model directly, which means we allows users to directly input their well-trained DNN models instead of making any changes in the systems. In this way, we may also avoid unnecessary interference on the attack process, and obtain a more objective and fair attack effect on the target model.

TABLE 1. Target model list.

Model name	Pre-trained	Dataset
VGG16	Yes	ImageNet
VGG19	Yes	ImageNet
Inception-v3	Yes	ImageNet
SqueezeNet	Yes	ImageNet
ResNet152-v2	Yes	ImageNet

TABLE 2. Information of deep neural network.

Neural network	Input image format	Class number	Top-1 accuracy	Top-5 accuracy
VGG16	224*224	1000	0.6986	0.8945
VGG19	224*224	1000	0.7072	0.8988
Inception-V3	299*299	1000	0.7755	0.9364
SqueezeNet V1.1	224*224	1000	0.5496	0.7817
ResNet152-v2	152*152	1000	0.7833	0.9409

In our experiment, we adopt various open source and well pre-trained models such as VGG16, VGG19, Inception-v3, SqueezeNet and ResNet152-v2 as shown in Table 1. In the experiment, we do not change and adjust the model itself, but use MXNet framework to build model calculation chart according to the structure of neural network. The effect in the system after the model is built is shown in Figure 2.

After the construction of these five models based on Gluon of MXNet framework, we download the pre-trained model weight file from MXNet Model Zoo [17], use these pre-trained model weights to build a deep neural network model server for image classification. Combined with the preprocessing method, the original uploaded image can be classified and predicted directly, and the prediction probability distribution can be obtained according to probability distribution and the final prediction results. The system realizes a complete end-to-end deep learning image classification system by using these five pre-trained deep learning models.

Table 2 shows the accuracy and other relevant information of the five neural network models used in this paper after loading the pre-trained weights of MXNet model zoo.

D. REMOTE IDENTIFICATION CALLING ENGINE

In addition to using local model to predict and classify image samples, SecureAS can also customize the interface to use MXNET-MODEL-SERVER to serve the pre-trained model. It uses a remote deep learning image recognition server to classify and predict images. In the process of model service, we can use the trained model file in the cloud to build the model service directly, or download the pre-trained model weight file from the cloud and start the model service locally.

The deep learning image classification server built does not know the internal situation of the model completely in the process of attack. It can only be predicted by uploading samples and obtaining the return results, which meets the requirements of blackbox attack on the target model. For this kind of remote model, we can directly define the model

name and model address in the security evaluation system. Then, we can use the preprocessing script to upload the image to the cloud image recognition engine through the local preprocessing after uploading the image. After that, we use the remote model to pre-test and analyze its security and prediction accuracy.

E. ADVERSARIAL EXAMPLES GENERATION MODULE

The adversarial examples generation module aims to generate the adversarial examples. According to Equation 5 and Equation 6, the generation of the adversarial images first needs to calculate the adversarial noise. The computation of the adversarial noise needs to get the partial derivative of the loss function of the target model to the original sample that is $\nabla_{x^n} J(\theta; x^n, y_{true})$. Therefore, it is necessary for the target neural network to carry out a feed-forward calculation and get the loss function value. Then, it calculates its gradient about the original sample, and finally updates the sample. In the whole calculation process, the weight of the neural network itself is constant, and what changes is the input sample image.

F. VULNERABILITY ASSESSMENT VISUALISATION MODULE

After generating the adversarial examples, the system adopts the DNN models to perform predictions for them. According to the algorithm of the MAS index, the MAS index is calculated by the prediction results of the adversarial examples and the original samples as well as the specific prediction probability distribution.

Evaluation results of a model include the attack algorithm, the original samples, the adversarial examples, the risk index of the current model, the security status of the model, the prediction categories of the model for the original samples and the adversarial examples. The calculation of the risk index of a model is based on the proposed MAS algorithm. A higher MAS index value indicates more vulnerability of the model. The vulnerability state of the model is obtained by the MAS index analysis with four levels: security, relatively security, risk and serious risk. Researchers can make subsequent improvements and adjustments to the model according to the model state, or use the hazard index as a reference index when making model selection and evaluating model performance.

V. SYSTEM FUNCTION EVALUATIONS

In this section, we will first present the SecureAS system interfaces/function test, and then provide all the experimental evaluation for the MAS index and proposed adversarial generation algorithms.

A. SecureAS - LOCAL MODEL PREDICTION TEST

SecureAS provides a user-friendly interface to allow users to select the images and DNN models performing the prediction tasks. Figure 2 shows the prediction results using the picture of cats. The prediction results indicate that, based on the pre-trained VGG16 model, the probability of spotted cats is much higher than that of other categories. The system predicts the spotted cats correctly.

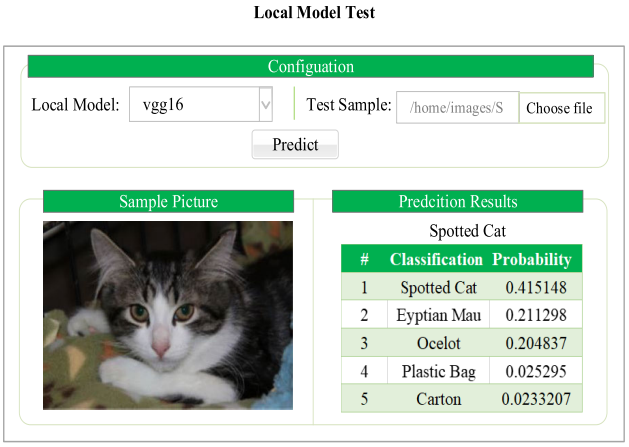

FIGURE 2. Illustration of local model prediction.

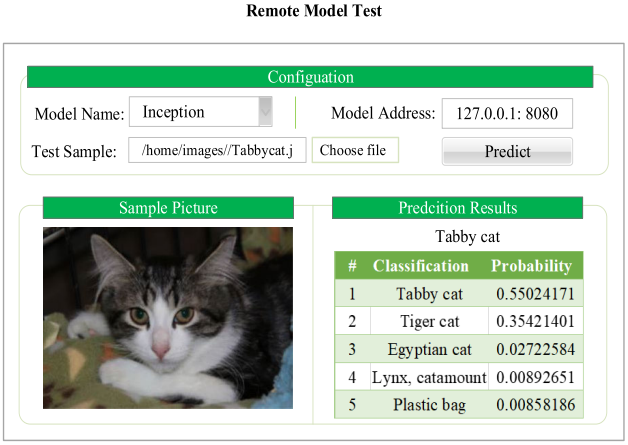

FIGURE 3. Illustration of remote model prediction.

B. SecureAS - REMOTE MODEL PREDICTION TEST

For remote model prediction, the system provides a user-friendly interface to allow users to specify the DNN model, the model address and the images to use. Figure 3 provides one example of how the user can use the Inception model with the address of 127.0.0.1: 8080 to evaluate one cat image. After uploading the image, the image recognition server calls the corresponding DNN model to perform the prediction and provides details information about the result.

C. SecureAS - VULNERABILITY ASSESSMENT ANALYSIS TEST

SecureAS also provides an interface to show users the vulnerability assessment results. Figure 4 is the final vulnerability evaluation result of the system for the target model. As shown in the Figure 4, the target model is VGG16 using the adversarial example generated by the attacking algo-

TABLE 3. Results of model vulnerability assessment.

Model	FGSM	FGNM	BIM(20 epochs)	PINM(20 epochs)	Original sample forecast	MAS
Pre-trained-VGG16	Carpet	Spotted cat	Carpet	Hound	Spotted cat	0.83
Pre-trained-VGG19	Persian cat	Spotted cat	Hound	Hound	Spotted cat	0.70
Pre-trained-SqueezeNet_v1.1	Doormat	Egyptian cat	Carpet	Persian cat	Egyptian cat	0.72
Pre-trained-Inception-v3	Doormat	Egyptian cat	Carpet	Egyptian cat	Egyptian cat	0.47
Pre-trained-ResNet-152	Tiger cat	Spotted cat	Carpet	Tiger cat	Tiger cat	0.24

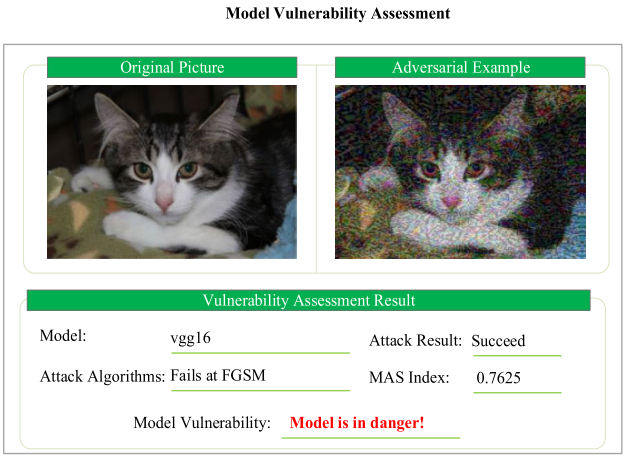

FIGURE 4. Illustration of model vulnerability assessment.

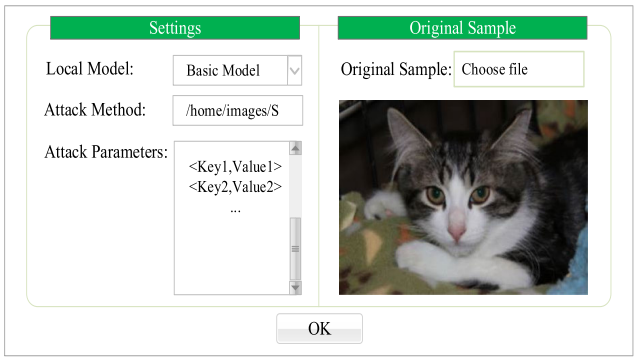

FIGURE 5. Parameters setting of adversarial examples generation.

rithm FGSM. The system also provides the security index (i.e. vulnerability index) calculated by the MAS index algorithm, which is 0.7625. Meanwhile, it evaluates the DNN model's risk level (i.e., high-risk in the example).

D. SecureAS - ADVERSARIAL EXAMPLES GENERATION TEST

When users want to generate the adversarial examples in the system, they can simply specify the adversarial example generation algorithm/parameters to use and the original samples. Figure 5 shows the user interface with one sample image and algorithm. After the user submit the image, the system provides results with details. To be intuitive, it shows the original image, the disturbance used by the algorithm and the generated adversarial example as shown in Figure 6. In addition, the system also indicates whether the adversarial example can attack the model successfully or not.

Furthermore, in addition to evaluate the local models, the system is also used to evaluate those models in the industrial image recognition engines like Google's large-scale image recognition engine. Figure 7 shows the recognition results of Google image recognition engine for the original image samples. Figure 8 shows the recognition results of Google image recognition engine for the adversarial image generated by SecureAS. The Google image recognition engine recognizes the original sample as cat and the adversarial image as art, but the main contents of the original sample and the adversarial image are cats. It can be seen that the adversarial examples generated by the system successfully deceive the industrial image recognition engine.

E. MAS INDEX EVALUATION

The aim of vulnerability evaluation experiments is to test the vulnerability of the target model without knowing the internal structure and weight of the target model for adversarial examples attack. For VGG19 model, four kinds of adversarial example generation algorithms are used to generate the adversarial image of the same original sample. In the whole process, the parameters and internal structure of VGG19 model are known, and the gradient of VGG19 neural network can be used to generate adversarial examples. But the internal parameters and gradients of the other four kinds of neural networks, including VGG16, SequeezeNet, Inception-V3 and ResNet-152, are unknown. These four kinds of neural networks are completely black boxes for attackers. During the attack, only their input and output can be obtained, and the loss function gradients cannot be obtained. Table 3 shows the prediction results of multiple target neural networks adversarial examples and the self vulnerability calculated by the MAS index. To some extent, these five kinds of DNNs will be deceived and misled by the adversarial examples generated by different attack algorithms. With a higher MAS index, the greater risk the model has, and the easier it is to be attacked and cheated by the adversarial examples. Table 3 shows that VGG16 has the biggest risk among the five neural networks.

F. ADVERSARIAL EXAMPLES GENERATION ALGORITHMS

The aim of this experiment is to evaluate the effectiveness of our proposed adversarial examples generation algorithms Fast Gradient Norm Method (FGNM) and Peak Iteration

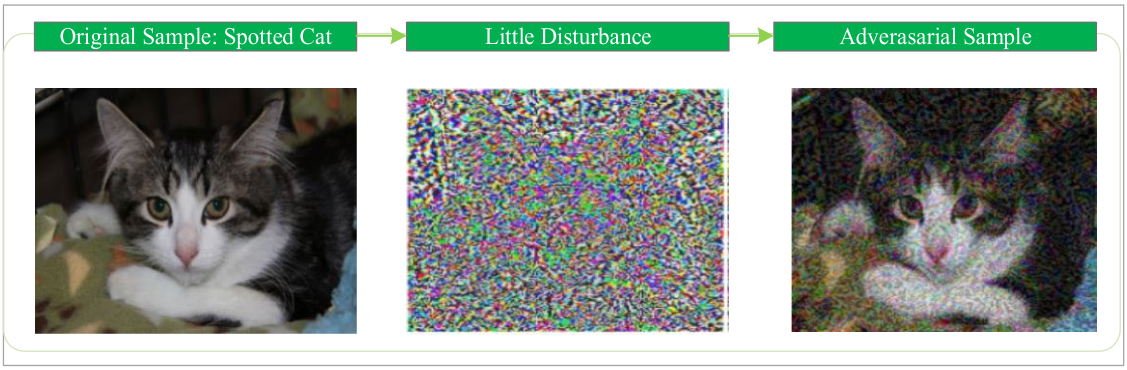

FIGURE 6. One possible result of adversarial example generation.

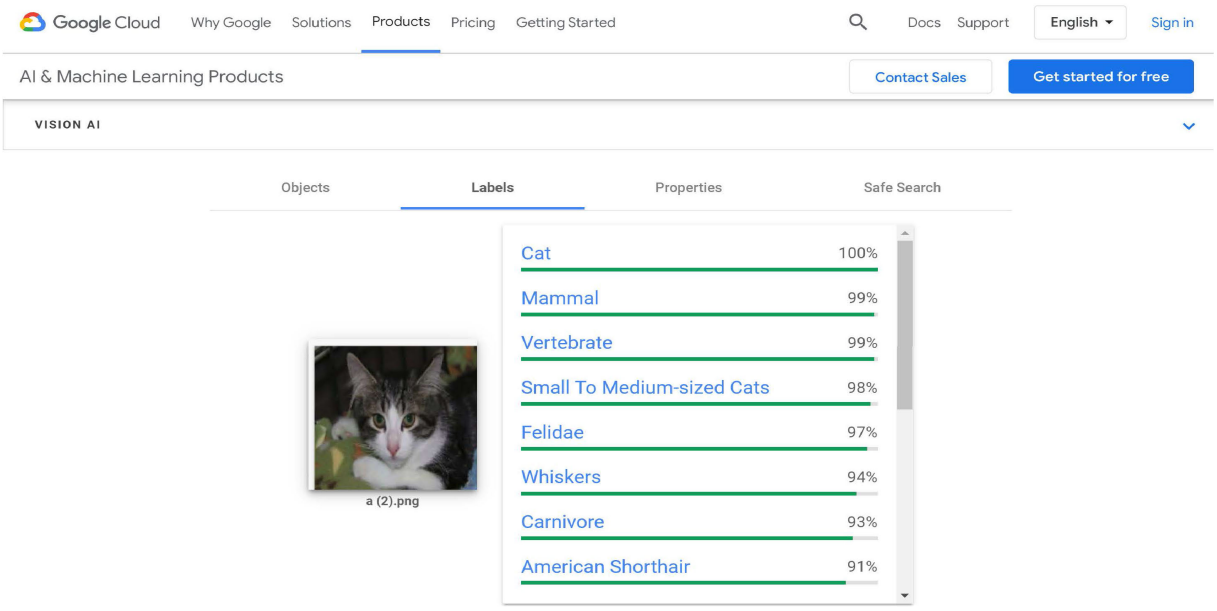

FIGURE 7. Recognition result of original image by Google image recognition engine.

TABLE 4. Configuration of adversarial examples generation algorithm.

Algorithm	Is iteration	Basic configuration
FGSM	No	ε=0.3
FGNM	No	ε=0.3
BIM	Yes	ε=0.1, epochs=20
PINM	Yes	ε=0.1, epochs=20

Norm Method (PINM) compared to the existing algorithms Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM). Table 4 shows the configurations of the four algorithms used. The four algorithms are divided into two types, iterative method and non iterative method (i.e. single step method). The configuration of each type of algorithm is the same.

We evaluate the four algorithms using the same original sample images. Figure 9 shows the generated adversarial examples images. In Figure 9, four different rows (from row one to row four) are the data for four different algorithms (FGSM, FGNM, BIM and PINM) respectively. In each row, the first image is the original sample used. The second image is the adversarial noise calculated by the adversarial example generation algorithm and the third column is the adversarial image generated. Table 5 shows the effect of these four adversarial example generation algorithms on VGG16 under the same basic parameter configuration.

According to Figure 9 and Table 5, under the same parameter configuration, the Peak Signal Noise Ratio (PSNR) of the adversarial image generated by our proposed algorithms (FGNM and PINM), is much higher than that of the image generated by FGSM and BIM. This indicates the quality of the image generated by the improved algorithms is much better than FGSM and BIM. Meanwhile, FGNM has achieved the highest quality. However, the adversarial image generated by FGNM under the default parameter configuration fails

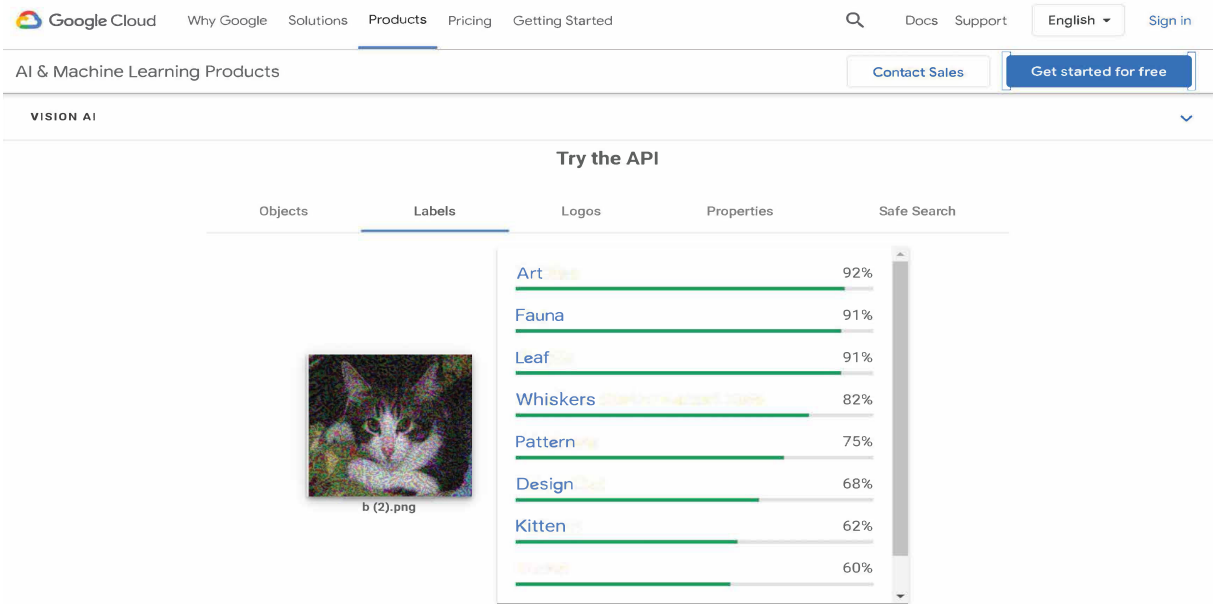

FIGURE 8. Recognition result of Google image recognition engine for adversarial examples.

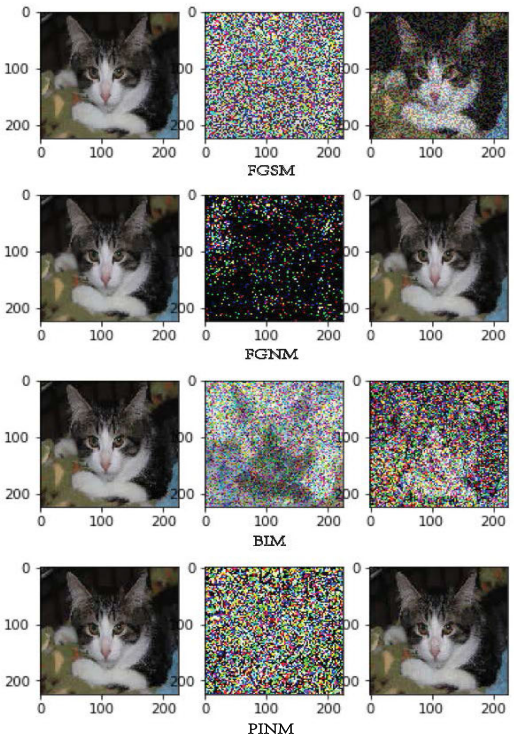

FIGURE 9. Comparison of adversarial examples generation algorithms.

to attack successfully. Additionally, PINM is able to generate higher quality of image than FGSM and BIM, and still achieve the same attack effect.

VI. RELATED WORK

Currently, several similar platforms have been proposed, such as Cleverhans [18], Foolbox [19], AdvBox [20], ART [21],

TABLE 5. Effectiveness comparison of adversarial examples generation algorithm.

Algorithm	Prediction category	Attack successfully	PSNR
FGSM	Carpet	Yes	37.01
FGNM	Spotted cat	No	53.17
BIM	Hound	Yes	38.33
PINM	Hound	Yes	44.14

DeepSec [22], etc. Cleverhans is the first open-source python library in Tensorflow to benchmark machine learning systems' vulnerability to adversarial example. Foolbox is an upgrade of Cleverhans working natively with other popular DNN frameworks such as PyTorch, Theano, and MXNet. Advbox is implemented on the PaddlePaddle to benchmark the robustness of machine learning models with providing 7 attacks, and ART integrates 7 attacks and 5 defenses. DeepSec provides a uniform platform for defenses/attacks testing in a white box way. Different from existing work, SecureAS may evaluate the vulnerability of a DNN both locally and remotely in a blackbox way, without knowing the inner structure of the target DNN. Meanwhile, it treats security evaluation as the first class citizen and implements all categories of adversary methods, which enables data scientists to conduct comprehensive and effective test on given attacks and DNNs.

Existing methods of generating adversarial examples can be divided into three major categories: (1) The first category is to generate the adversarial examples directly using L-BFGS or Adam [23] to solve the optimization problem of adversarial examples. This optimization-based generation method is often slower than other methods, but it is more powerful. (2) The second category is to generate the examples based on one-step gradient approximation methods, such as Fast

Gradient Sign Method (FGSM) [12] or fast least possible class approximation method [13]. These methods can quickly construct the adversarial examples, only need to propagate the target model forward once and calculate the reverse gradient, and add the interference calculated by the reverse gradient to the original clean samples to generate the adversarial examples. (3) The third category is to obtain the examples by using the improved iterative methods based on the reverse gradient. These methods carry out forward propagation and calculate the reverse gradient for the target deep neural network many times. They construct the interference based on the results of multiple iterations, and finally generate the adversarial examples.

VII. CONCLUSION

In this paper, we presented SecureAS, a DNN security evaluation system. Compared with existing attack/defense platforms, SecureAS assesses the vulnerability of existing DNN in a blackbox way, both locally and remotely. The core of SecureAS is the MAS index, which is used to evaluate the model security risk quantitatively. The MAS index can directly quantify the security of image recognition deep learning models. We expect MAS index to be used as an indicator for other aspects like accuracy and performance as well. In addition, we introduce two other improved adversarial example generation algorithms: Fast Gradient Norm Method (FGNM) and Peak Iteration Norm Method (PINM) which are able to generate higher quality of examples. We conduct extensive experiments and verify that the system can effectively analyze and evaluate the vulnerability and risks of DNN models.

REFERENCES

[1] G. Hee Lee, F. Fraundorfer, and M. Pollefeys, "Structureless pose-graph loop-closure with a multi-camera system on a self-driving car," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 564–571.

[2] C. Middlehurst. *China Unveils World'S First Facial Recognition ATM*. [EB/OL]. Accessed: Apr. 4, 2020. [Online]. Available: http://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009, doi: 10.1109/TPAMI.2008.79.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[5] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: Deep learning in Android malware detection," in *Proc. ACM Conf. SIGCOMM (SIGCOMM)*, F. E. Bustamante, Y. C. Hu, A. Krishnamurthy, and S. Ratnasamy, Eds. Chicago, IL, USA: ACM, Aug. 2014, pp. 371–372, doi: 10.1145/2619239.2631434.

[6] P. Rajpurkar, J. Irvin, A. Bagul, D. Y. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, "MURA dataset: Towards radiologist-level abnormality detection in musculoskeletal radiographs," *CoRR*, vol. abs/1712.06957, Dec. 2017. [Online]. Available: http://arxiv.org/abs/1712.06957

[7] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: Successful approaches and future challenges," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 5, no. 6, pp. 292–303, Nov. 2015.

[8] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, Sep. 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[9] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, L. Getoor and T. Scheffer, Eds. Washington, DC, USA: Omnipress, Jun./Jul. 2011, pp. 1017–1024. [Online]. Available: https://icml.cc/2011/papers/524_icmlpaper.pdf

[10] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. 32nd Int. Conf. Mach. Learn. Res.*, vol. 37, F. R. Bach and D. M. Blei, Eds. Lille, France: JMLR.org, Jul. 2015, pp. 1462–1471. [Online]. Available: http://proceedings.mlr.press/v37/gregor15.html

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Represent.*, Jan. 2014, pp. 1–10.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015, pp. 1–11. [Online]. Available: http://arxiv.org/abs/1412.6572

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=HJGU3Rodl

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2008. [Online]. Available: http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[16] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *CoRR*, vol. abs/1512.01274, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.01274

[17] Apache. *Mxnet Model Zoo*. [EB/OL]. Accessed: May 4, 2020. [Online]. Available: https://mxnet.apache.org/versions/1.4.1/model_zoo/index.html

[18] I. J. Goodfellow, N. Papernot, and P. D. McDaniel, "Cleverhans v0.1: An adversarial machine learning library," *CoRR*, vol. abs/1610.00768, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1610.00768

[19] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models," *CoRR*, vol. abs/1707.04131, Jul. 2017. [Online]. Available: http://arxiv.org/abs/1707.04131

[20] D. Goodman, X. Hao, Y. Wang, Y. Wu, J. Xiong, and H. Zhang, "Advbox: A toolbox to generate adversarial examples that fool neural networks," *CoRR*, vol. abs/2001.05574, Jan. 2020. [Online]. Available: https://arxiv.org/abs/2001.05574

[21] M. Nicolae, M. Sinn, T. N. Minh, A. Rawat, M. Wistuba, V. Zantedeschi, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v0.2.2," *CoRR*, vol. abs/1807.01069, Jul. 2018. [Online]. Available: http://arxiv.org/abs/1807.01069

[22] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang, "DEEPSEC: A uniform platform for security analysis of deep learning model," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 673–690.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA: arXiv.org, May 2015, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980

YAN CHU was born in Harbin, China. She received the B.S., M.S., and Ph.D. degrees from the College of Computer Science and Technology, Harbin Engineering University, in 2002, 2005, and 2008, respectively. She has been an Assistant Professor with the College of Computer Science and Technology, Harbin Engineering University. She is the author of one book, five inventions, and more than 20 articles. Her interests include machine learning, data analysis, and recommendation system.

XIAO YUE received the Ph.D. degree from the Zhongnan University of Economics and Law, in 2010. From 2013 to 2014, she was a Postdoctoral Researcher with the University of Bern. She is currently working as an Assistant Professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law. She has published more than 20 articles, host one foundation of National Social Sciences, Social Science Youth Foundation of Ministry of Education of China. Her interests include public safety and emergence big data.

QUAN WANG was born in 1996. He received the B.S. degree from the College of National Security, Harbin Engineering University, in 2018. He had an Internship at Hangzhou Moresec Technology Company Limited, from 2016 to 2017, working on the application of machine learning in cyber security. Since 2018, he has been working Tencent Technology Company Limited, and has also been involved in intrusion detection and situation awareness (SA). He has four invention patents. His interests include machine learning, cybersecurity, and adversarial samples.

ZHENGKUI WANG received the M.S. degree from the Department of Computer Science, Harbin Institute of Technology, China, in 2008, and the Ph.D. degree from National University of Singapore, in 2013. He is currently working with the Singapore Institute of Technology, as an Assistant Professor. His research interests include large-scale data analysis, data warehousing, cloud computing, and scientific data processing.