2019 IEEE Symposium on Security and Privacy

DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model

Xiang Ling*,#, Shouling Ji*,†,#(✉), Jiaxu Zou*, Jiannan Wang*, Chunming Wu*, Bo Li‡ and Ting Wang§
*Zhejiang University, †Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies,
‡UIUC, §Lehigh University
{lingxiang, sji, zoujx96, wangjn84, wuchunming}@zju.edu.cn, lxbosky@gmail.com, inbox.ting@gmail.com

***Abstract*—Deep learning (DL) models are inherently vulnerable to adversarial examples – maliciously crafted inputs to trigger target DL models to misbehave – which significantly hinders the application of DL in security-sensitive domains. Intensive research on adversarial learning has led to an arms race between adversaries and defenders. Such plethora of emerging attacks and defenses raise many questions: Which attacks are more evasive, preprocessing-proof, or transferable? Which defenses are more effective, utility-preserving, or general? Are ensembles of multiple defenses more robust than individuals? Yet, due to the lack of platforms for comprehensive evaluation on adversarial attacks and defenses, these critical questions remain largely unsolved.**

In this paper, we present the design, implementation, and evaluation of DEEPSEC, a uniform platform that aims to bridge this gap. In its current implementation, DEEPSEC incorporates 16 state-of-the-art attacks with 10 attack utility metrics, and 13 state-of-the-art defenses with 5 defensive utility metrics. To our best knowledge, DEEPSEC is the first platform that enables researchers and practitioners to (i) measure the vulnerability of DL models, (ii) evaluate the effectiveness of various attacks/defenses, and (iii) conduct comparative studies on attacks/defenses in a comprehensive and informative manner. Leveraging DEEPSEC, we systematically evaluate the existing adversarial attack and defense methods, and draw a set of key findings, which demonstrate DEEPSEC's rich functionality, such as (1) the trade-off between misclassification and imperceptibility is empirically confirmed; (2) most defenses that claim to be universally applicable can only defend against limited types of attacks under restricted settings; (3) it is not necessary that adversarial examples with higher perturbation magnitude are easier to be detected; (4) the ensemble of multiple defenses cannot improve the overall defense capability, but can improve the lower bound of the defense effectiveness of individuals. Extensive analysis on DEEPSEC demonstrates its capabilities and advantages as a benchmark platform which can benefit future adversarial learning research.

I. INTRODUCTION

Recent advances in deep learning (DL) techniques have led to breakthroughs in a number of long-standing artificial intelligence tasks (e.g., image classification, speech recognition, and even playing Go [1]). Unfortunately, it has been demonstrated that existing DL models are inherently vulnerable to *adversarial examples* [2], which are maliciously crafted inputs to trigger target DL models to misbehave. Due to the increasing use of DL models in security-sensitive domains (e.g., self-driving cars [3], face recognition [4], malware detection [5], medical diagnostics [6]), the phenomena of adversarial examples has attracted intensive studies from both academia and industry, with a variety of adversarial attack and defense methods being proposed [2], [7], [8]. At a high level, the attacks attempt to force the target DL models to misclassify using adversarial examples, which are often generated by slightly perturbing legitimate inputs; meanwhile, the defenses attempt to strengthen the resilience of DL models against such adversarial examples, while maximally preserving the performance of DL models on legitimate instances.

The security researchers and practitioners are now facing a myriad of adversarial attacks and defenses; yet, there is still a lack of quantitative understanding about the strengths and limitations of these methods due to incomplete or biased evaluation. First, they are often assessed using simple metrics. For example, misclassification rate is used as the primary metric to evaluate attack methods. However, as shown in our studies, misclassification rate alone is often insufficient to characterize an attack method. Second, they are only evaluated against a small set of attacks/defenses, e.g., many defenses are evaluated using a few "strong" attacks. However, as found in our studies, defenses robust against "stronger" attacks are not necessarily immune to "weaker" ones. Third, the constant arms race between adversarial attacks and defenses invalidates conventional wisdom quickly. For instance, the gradient obfuscation strategy adopted by many defenses is later shown to be ineffective [9]. The compound effects of these factors often result in contradictory and puzzling conclusions about the same attack/defense methods. As an example, defensive distillation (DD) [10] was evaluated against JSMA [11] and claimed to significantly improve the robustness of DL models. Nevertheless, it was soon found to only provide marginal robustness improvement against new attacks (e.g., C&W [12]). Moreover, it was later shown that models trained with DD may perform ever worse than undefended models [13].

We argue that to further advance the research on adversarial examples, it is critical to provide an analysis platform to support comprehensive and informative evaluation of adversarial attacks and defenses. We envision that a set of desiderata are required for such a platform to be practically useful:

- Uniform – It should support to compare different attack/defense methods under the same setting;
- Comprehensive – It should include most representative attack/defense methods;
- Informative – It should include a rich set of metrics to assess different attack/defense methods;
- Extensible – It should be easily extended to incorporate new attack/defense methods.

Unfortunately, none of the existing work (e.g., Cleverhans [14]) meets all the requirements (details in Section VI).

#Xiang Ling and Shouling Ji are the co-first authors. Shouling Ji is the corresponding author.

DOI 10.1109/SP.2019.00023

To bridge this gap, we present DEEPSEC, a first-of-its-kind platform for security analysis of DL models, that satisfies all the aforementioned desiderata. In its current implementation, it incorporates 16 state-of-the-art adversarial attacks with 10 attack utility metrics and 13 representative defenses with 5 defense utility metrics. DEEPSEC enables security researchers and practitioners to (i) assess the vulnerabilities of given DL models to various attacks, (ii) evaluate the effectiveness of various defenses, and (iii) conduct comparative studies on different attacks/defenses in a comprehensive and informative manner. To summarize, we make the following contributions.

1) We present DEEPSEC, the first platform designed specifically to serve as an evaluation platform for adversarial attacks/defenses. Two key features differentiate DEEPSEC from the state-of-the-art adversarial learning libraries: a) to our best knowledge, DEEPSEC includes the largest collection of attack/defense methods (16 attacks and 13 defenses) thus far (e.g., Cleverhans [14] only provides 9 attacks and 1 defenses); b) it treats the evaluation metrics as first-class citizens, thereby supporting the evaluation of attacks/defenses in a uniform and informative manner.
2) Using DEEPSEC, we perform thus far the largest-scale empirical study on adversarial attacks/defenses under different metrics, among which 10 for attack and 5 for defense evaluation are proposed within the paper in addition to the existing ones. Moreover, we perform the largest-scale cross evaluation between different attack and defense methods (16×13) to understand their relative strengths and limitations.
3) Through this systematic study, we obtain a set of interesting and insightful findings that may advance the field of adversarial learning: a) the trade-off between misclassification and imperceptibility performance of adversarial examples is experimentally confirmed; b) most defenses that claim to be universally applicable are only effective for a very limited number of attacks or partially effective for attacks under restricted settings; c) the ensemble of multiple defenses cannot improve the overall defense capability, but can improve the lower bound of the defense effectiveness of individuals.

Acronyms and Notations. For convenient reference, we summarize the acronyms and notations in Tables I and II.

TABLE I
ABBREVIATIONS AND ACRONYMS

Terms		AE	Adversarial Example
		TA	Targeted Attack
		UA	Un-targeted Attack
Attacks	Un-targeted Attacks	FGSM	Fast Gradient Sign Method [15]
		R+FGSM	Random perturbation with FGSM [16]
		BIM	Basic Iterative Method [17]
		PGD	Projected L_∞ Gradient Descent attack [18]
		U-MI-FGSM	Un-targeted Momentum Iterative FGSM [19]
		DF	DeepFool [20]
		UAP	Universal Adversarial Perturbation attack [21]
		OM	OptMargin [22]
	Targeted Attacks	LLC	Least Likely Class attack [17]
		R+LLC	Random perturbation with LLC [16]
		ILLC	Iterative LLC attack [17]
		T-MI-FGSM	Targeted Momentum Iterative FGSM [19]
		BLB	Box-constrained L-BFGS attack [2]
		JSMA	Jacobian-based Saliency Map Attack [11]
		CW	Carlini and Wagner's attack [12]
		EAD	Elastic-net Attacks to DNNs [23]
Defenses	Complete Defenses	NAT	Naive Adversarial Training [24]
		EAT	Ensemble Adversarial Training [16]
		PAT	PGD-based Adversarial Training [18]
		DD	Defensive Distillation [10]
		IGR	Input Gradient Regularization [13]
		EIT	Ensemble Input Transformation [25]
		RT	Random Transformations based defense [26]
		PD	PixelDefense [27]
		TE	Thermometer Encoding defense [28]
		RC	Region-based Classification [29]
	Detection	LID	Local Intrinsic Dimensionality based detector [30]
		FS	Feature Squeezing detector [31]
		MagNet	MagNet detector [32]
Utility Metrics	Attacks	MR	Misclassification Ratio
		ACAC	Average Confidence of Adversarial Class
		ACTC	Average Confidence of True Class
		ALD_p	Average L_p Distortion
		ASS	Average Structural Similarity
		PSD	Perturbation Sensitivity Distance
		NTE	Noise Tolerance Estimation
		RGB	Robustness to Gaussian Blur
		RIC	Robustness to Image Compression
		CC	Computation Cost
	Defenses	CAV	Classification Accuracy Variance
		CRR/CSR	Classification Rectify/Sacrifice Ratio
		CCV	Classification Confidence Variance
		COS	Classification Output Stability

II. ATTACKS & DEFENSES

In this paper, we consider the non-adaptive and white-box attack scenarios, where the adversary has full knowledge of the target DL model but is not aware of defenses that might be deployed. Since most white-box or non-adaptive attacks can be applied to black-box attacks based on transferability or adjustments to specific defenses, considering them would provide general understanding of current attack scenarios [33]–[35]. Further, we focus on classification tasks.

In this section, we summarize the state-of-the-art attack and defense methods and present a rich set of metrics to assess the utility of attack/defense methods.

A. Adversarial Attack Advances

In general, existing attacks can be classified along multiple different dimensions [8]. In this subsection, we classify attacks along two dimensions: *adversarial specificity* (i.e., **UA** and **TA**) and *attack frequency* (i.e., **non-iterative attack** and **iterative attack**). Specifically, UAs aim to generate AEs that can be misclassified into any class which is different from the ground truth class, while TAs aim to generate AEs to be misclassified into a specific target class. For attack frequency, non-iterative attacks take only one single step to generate AEs, while iterative attacks take multiple iterative updates. In fact, those two categorizations are closely integrated, but we describe them separately for clarity.

*1) **Non-iterative UAs**:* In [15], Goodfellow et al. proposed the first and fastest non-iterative UA, called *Fast Gradient Sign Method* (**FGSM**). By linearizing the loss function, FGSM perturbs an image by maximizing the loss subject to a L_∞

TABLE II
NOTATIONS USED IN THIS PAPER

Notations	Description
$\vec{\mathbf{X}} = \{X_1, \cdots, X_N\}$	$\vec{\mathbf{X}}$ is the testing set with N original examples, where $X_i \in R^m$.
$\vec{\mathbf{Y}} = \{y_1, \cdots, y_N\}$	$\vec{\mathbf{Y}}$ is the corresponding ground-truth label set of $\vec{\mathbf{X}}$, where $y_i = 1, \cdots, k$.
$\mathbf{F} : R^m \rightarrow \{1, \cdots, k\}$	$\mathbf{F}$ is a DL classifier on k classes, where $F(\vec{\mathbf{X}}) = \vec{\mathbf{y}}$.
$P : R^m \rightarrow R^k$	P is the softmax layer output of $\mathbf{F}$, where $\mathbf{F}(X) = \arg\max_j P(X)_j$.
$P(X)_j$	$P(X)_j$ represents the j-th probability of $P(X)$, where $j \in \{1, \cdots, k\}$
θ	θ is the parameter of $\mathbf{F}$.
$X^a \in R^m$	X^a is the adversarial example of X.
y^*	The specified target class for TAs.
$\mathbf{J} : R^m \times \{1 \cdots k\} \rightarrow R^+$	$\mathbf{J}$ is the loss function of $\mathbf{F}$.

constraint: $X^a = X + \epsilon \cdot sign(\nabla_X J(X, y^{true}))$, where ϵ is the hyper-parameter of L_∞ constraint. Similarly, Tramèr et al. [16] proposed a non-iterative UA, **R+FGSM**, which applies a small random perturbation before linearizing the loss function.

*2) **Iterative UAs:*** Kurakin et al. [17] introduced an intuitive extension of FGSM - *Basic Iterative Method* (**BIM**) that iteratively takes multiple small steps while adjusting the direction after each step: $X_0^a = X$; $X_{n+1}^a = Clip_{x,\epsilon}(X_n^a + \alpha \cdot sign(\nabla_X J(X_n^a, y^{true})))$, where $Clip_{x,\epsilon}$ is used to restrict the L_∞ of perturbation. Following BIM, Madry et al. [18] introduced a variation of BIM by applying the projected gradient descent algorithm with random starts, named as the **PGD** attack. Similarly, Dong et al. [19] integrated the momentum techniques [36] into BIM for the purpose of stabilizing the updating direction and escaping from poor local maximum during iterations. We refer this UA as **U-MI-FGSM**. **DeepFool** [20] was proposed to generate AEs by searching for the closest distance from the source image to the decision boundary of the target model. Further in [21], Moosavi-Dezfooli et al. developed a *Universal Adversarial Perturbation* (**UAP**) attack, in which an image-agnostic and universal perturbation can be used to misclassify almost all images sampled from the dataset. In [22], He et al. proposed an attack, OptMargin (**OM**), to generate robust AEs that can evade existing region-based classification defense.

*3) **Non-iterative TAs:*** The TA version of FGSM was introduced in [17] to specify the least likely class y^{LL} of an original image X as the target class, where $y^{LL} = \arg\min_y P(y|X)$. We refer this non-iterative TA as the *Least-Likely Class* (**LLC**) attack: $X^a = X - \epsilon \cdot sign(\nabla_X J(X, y^{LL}))$. Similar to R+FGSM, Tramèr et al. [16] introduced **R+LLC**, which also integrates a small random step before linearizing the loss function.

*4) **Iterative TAs:*** The first adversarial attack discovered by Szegedy et al. [2] is an iterative TA, which generates AEs by a *Box-constrained L-BFGS* (**BLB**) algorithm. However, BLB has several limitations, e.g., it is time-consuming and impractical to linearly search for the optimal solution at large scale. To facilitate the efficiency of iterative TAs, Kurakin et al. [17] proposed a straightforward iterative version of LLC - **ILLC**. Following the attacks in [19], momentum techniques can also be generalized to ILLC, called *targeted MI-FGSM* (**T-MI-FGSM**). Taking a different perceptive, Papernot et al. [11] proposed the *Jacobian-based Saliency Map Attack* (**JSMA**). Specifically, JSMA first computes the Jacobian matrix of a given sample X, and then perturbs it by finding the input features of X that make the most significant changes to the output. Carlini and Wagner [12] introduced a set of powerful attacks based on different norm measurements on the magnitude of perturbation, termed as **CW**. In particular, CW is formalized as an optimization problem to search for high-confidence AEs with small magnitude of perturbation, and has three variants: CW0, CW2 and CW∞, respectively. In [23], Chen et al. argued that L_1 has not been explored to generate AEs. Therefore, their *Elastic-net Attack to DNNs* (**EAD**) formulates the generation of AE as an elastic-net regularized optimization problem and features L_1-oriented AEs.

B. Utility Metrics of Attacks

From the view of economics, utility is a measure of whether goods or services provide the features that users need [37]. For adversaries who want to attack DL models, utility means to what extent the adversarial attack can provide "successful" AEs. Generally speaking, successful AEs should not only can be misclassified by the model, but also be imperceptible to humans, robust to transformations as well as resilient to existing defenses depending on the adversarial goals.[1]

In this paper, we consider misclassification, imperception, and robustness as utility requirements while taking the resilience as the security requirement. We will first define 10 utility metrics for adversarial attacks below.

1) Misclassification: Firstly, we summarize utility metrics in terms of misclassification as follows.

Misclassification Ratio (MR). Misclassification is the most important property for adversarial attacks. In the case of UAs, MR is defined as the percentage of AEs that are successfully misclassified into an arbitrary class except their ground truth classes. For TAs, MR is defined as the percentage of AEs misclassified into the target classes as specified before. More specifically, $MR_{UA} = \frac{1}{N}\sum_{i=1}^{N} count(\mathbf{F}(X_i^a) \neq y_i)$ and $MR_{TA} = \frac{1}{N}\sum_{i=1}^{N} count(\mathbf{F}(X_i^a) = y_i^*)$.

Average Confidence of Adversarial Class (ACAC). For AEs, ACAC is defined as the average prediction confidence towards the incorrect class, i.e., $ACAC = \frac{1}{n}\sum_{i=1}^{n} P(X_i^a)_{F(X_i^a)}$, where n ($n \leq N$) is the total number of successful AEs.

Average Confidence of True Class (ACTC). By averaging the prediction confidence of true classes for AEs, ACTC is used to further evaluate to what extent the attacks escape from the ground truth: $ACTC = \frac{1}{n}\sum_{i=1}^{n} P(X_i^a)_{y_i}$.

2) Imperceptibility: In essence, imperceptibility implies that the adversarial example would still be correctly classified by human vision, which ensures that the adversarial and benign

[1]In this paper, we distinguish between the robustness and resilience of AEs. Specifically, robustness reflects the misclassification stability after preprocessing by inevitable transformations in physical world, while resilience represents the surveillance of AEs when being defended by well-designed defenses.

examples convey the same semantic meaning. To evaluate the imperceptibility of AEs, we detail the metrics as follows.

Average $\mathbf{L_p}$ Distortion ($\mathbf{ALD_p}$). Almost all existing attacks adopt L_p norm distance (i.e., $p=0,1,\infty$) as distortion metrics for evaluations. Specifically, L_0 counts the number of pixels changed after the perturbation; L_2 computes the Euclidean distance between original and perturbed examples; L_∞ measures the maximum change in all dimensions of AEs. In short, we define ALD_p as the average normalized L_p distortion for all successful AEs, i.e., $ALD_p = \frac{1}{n}\sum_{i=1}^{n}\frac{\|X_i^a - X_i\|_p}{\|X_i\|_p}$. The smaller ALD_p is, the more imperceptible the AEs are.

Average Structural Similarity (ASS). As one of the commonly used metrics to quantify the similarity between two images, SSIM [38] is considered to be more consistent to human visual perception than L_p similarity. To evaluate the imperceptibility of AEs, we define ASS as the average SSIM similarity between all successful AEs and their original examples, i.e., $ASS = \frac{1}{n}\sum_{i=1}^{n} SSIM(X_i^a, X_i)$. Intuitively, the greater the ASS is, the more imperceptible the AEs are.

Perturbation Sensitivity Distance (PSD). Based on the contrast masking theory [39], PSD was proposed in [40] to evaluate human perception of perturbations, where $PSD = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\delta_{i,j}Sen(R(x_{i,j}))$, where m is the total number of pixels, $\delta_{i,j}$ denotes the j-th pixel of the i-th example, $R(x_{i,j})$ represents the surrounding square region of $x_{i,j}$, and $Sen(R(x_{i,j})) = 1/std(R(x_{i,j}))$ with $std(R(x_{i,j}))$ the standard deviation function. The smaller PSD is, the more imperceptible AEs are.

3) Robustness: Normally, images in physical world are inevitably preprocessed before feeding them into production systems (e.g., online image classification systems), which may lead to declines in MR for AEs. Thus, it is essential to evaluate the robustness of AEs under various realistic conditions.

Noise Tolerance Estimation (NTE). In [40], the robustness of AEs is estimated by noise tolerance, which reflects the amount of noises that AEs can tolerate while keeping the misclassified class unchanged. Specifically, NTE calculates the gap between the probability of misclassified class and the max probability of all other classes, i.e., $NTE = \frac{1}{n}\sum_{i=1}^{n}[P(X_i^a)_{F(X_i^a)} - \max\{P(X_i^a)_j\}]$, where $j \in \{1,\cdots,k\}$ and $j \neq F(X_i^a)$. The higher NTE is, the more robust AEs are.

On the other hand, due to the uncertainty of what transformations may be used, we thus sample two most widely and possibly used image preprocessing methods, *Gaussian blur* and *Image compression*, to evaluate the robustness of AEs.

Robustness to Gaussian Blur (RGB). Gaussian blur is widely used as a preprocessing stage in computer vision algorithms to reduce noises in images. Normally, a robust AE should maintain its misclassification effect after Gaussian blur. That is, $RGB_{UA} = \frac{count(\mathbf{F}(\mathbf{GB}(X_i^a)) \neq y_i)}{count(\mathbf{F}(X_i^a) \neq y_i)}$ and $RGB_{TA} = \frac{count(\mathbf{F}(\mathbf{GB}(X_i^a)) = y_i^*)}{count(\mathbf{F}(X_i^a) = y_i^*)}$, where $\mathbf{GB}$ denotes the Gaussian blur function. The higher RGB is, the more robust AEs are.

Robustness to Image Compression (RIC). Similar to RGB, RIC can be formulated as: $RIC_{UA} = \frac{count(\mathbf{F}(\mathbf{IC}(X_i^a)) \neq y_i)}{count(\mathbf{F}(X_i^a) \neq y_i)}$ and $RIC_{TA} = \frac{count(\mathbf{F}(\mathbf{IC}(X_i^a)) = y_i^*)}{count(\mathbf{F}(X_i^a) = y_i^*)}$, where $\mathbf{IC}$ denotes the specific image compression function. Also, the higher RIC is, the more robust AEs are.

4) Computation Cost: We define the **Computation Cost (CC)** as the runtime for attackers to generate an AE on average, and therefore evaluate the attack cost.

C. Defense Advances

In general, existing defense techniques can be classified into 5 categories. We discuss each category as follows.

1) Adversarial Training: Adversarial training has been proposed since the discovery of AEs in [2], with the hope that it can learn robust models via augmenting the training set with newly generated AEs. However, adversarial training with AEs generated by BLB in [2] suffers from high computation cost, which is impractical for large-scale training tasks.

To scale adversarial training to large-scale datasets, Kurakin et al. [24] presented a computationally efficient adversarial training with AEs generated by LLC, which we refer to as *Naive Adversarial Training* (**NAT**). Later, Tramèr et al. [16] proposed the *Ensemble Adversarial Training* (**EAT**) that augments training data with AEs generated by R+FGSM on other pre-trained models instead of the original model. Another variant of adversarial training, referred to as *PGD-based Adversarial Training* (**PAT**), was presented in [18] via retraining the model with AEs generated by PGD iteratively.

2) Gradient Masking/Regularization: A natural idea to defend against adversarial attacks is to reduce the sensitivity of models to AEs and hide the gradients [41], which is referred to as the gradient masking/regularization method.

In [10], Papernot et al. introduced the *Defensive Distillation* (**DD**) defense to reduce or smooth the amplitude of network gradients and make the defended model less sensitive w.r.t perturbations in AEs. However in [13], Ross and Doshi-Velez claimed that DD-enhanced models perform no better than undefended models in general. Aiming at improving robustness of models, they introduced the *Input Gradient Regularization* (**IGR**), which directly optimizes the model for more smooth input gradients w.r.t its predictions during training.

3) Input Transformation: As defenses discussed above either depend on generated AEs or require modifications to the original model, it is particularly important to devise attack/model-agnostic defenses against adversarial attacks. Researchers have attempted to remove the adversarial perturbations of the testing inputs before feeding them into the original model, which we refer to as input transformation defenses.

Using five different image transformation techniques, Guo et al. [25] showed that training the models on corresponding transformed images can effectively defend against existing attacks, which we refer to as *Ensemble Input Transformation* (**EIT**). Another similar work is [26], where Xie et al. introduced a *Random Transformations-based* (**RT**) defense. In RT, the testing images first go through two additional randomization layers, and then the transformed images are passed to the original model. In [27], Song et al. proposed *PixelDefense* (**PD**) to purify adversarial perturbations. More specifically, PD makes use of the PixelCNN [42], a generative model, to

purify the AEs and then passes the purified examples to the original model. Buckman et al. [28] proposed the *Thermometer Encoding* (**TE**) method to retrain the classification model with discretized inputs using thermometer encoding, and discretize the testing inputs before passing them to the retrained model.

4) Region-based Classification: Region-based Classification (**RC**) defense [29] takes the majority prediction on examples that are uniformly sampled from a hypercube around the AE, since they found that the hypercube around an AE greatly intersects with its true class region of the AE.

5) Detection-only Defenses: Given the difficulty in classifying AEs correctly, a number of detection-only defenses have been proposed to merely detect AEs and reject them. In this part, we introduce several latest and representative works and refer interested readers to [8], [43] for more others.

Ma et al. [30] proposed a *Local Intrinsic Dimensionality* based detector (**LID**) to discriminate AEs from normal examples due to the observation that the LID of AEs is significantly higher than that of normal examples. In [31], Xu et al. proposed the *Feature Squeezing* (**FS**) method to detect AEs via comparing the prediction difference between the original input and corresponding squeezed input. In [32], Meng and Chen proposed the **MagNet** defense framework, which is a combination defense of complete defense (i.e., the reformer) and detection-only defense (i.e., the detector).

D. Utility Metrics of Defenses

In general, defenses can be evaluated from two perspectives: *utility preservation* and *resistance to attacks*. Particularly, the utility preservation captures how the defense-enhanced model preserves the functionality of the original model, while the resistance reflects the effectiveness of defense-enhanced model against adversarial attacks. For the utility of defense, it does not make sense for detection-only defenses since they only detect AEs and reject them. Thus, it is important to note that we only evaluate the utility performance of complete defenses.

Suppose we attain the defense-enhanced model $\mathbf{F^D}$ of $\mathbf{F}$, while P^D denotes the corresponding softmax layer output of $\mathbf{F^D}$. Next, we detail 5 utility metrics of defenses.

Classification Accuracy Variance (CAV). The most important metric used to evaluate the performance of a DL model is accuracy. Therefore, a defense-enhanced model should maintain the classification accuracy on normal testing examples as much as possible. In order to evaluate the impact of defenses on accuracy, we define $CAV = Acc(\mathbf{F^D}, T) - Acc(\mathbf{F}, T)$, where $Acc(\mathbf{F}, T)$ denotes model $\mathbf{F}$'s accuracy on dataset T.

Classification Rectify/Sacrifice Ratio (CRR/CSR). To assess how defenses influence the predictions of models on the testing set, we detail the difference of predictions before and after applying defenses. We define the CRR as the percentage of testing examples that are misclassified by $\mathbf{F}$ previously but correctly classified by $\mathbf{F^D}$. Inversely, CSR is the percentage of testing examples that are correctly classified by $\mathbf{F}$ but misclassified by $\mathbf{F^D}$. That is, $CRR = \frac{1}{N}\sum_{i=1}^{N} count(\mathbf{F}(X_i) \neq y_i \& \mathbf{F^D}(X_i) = y_i)$ and $CSR = \frac{1}{N}\sum_{i=1}^{N} count(\mathbf{F}(X_i) = y_i \& \mathbf{F^D}(X_i) \neq y_i)$. In fact, $CAV = CRR - CSR$.

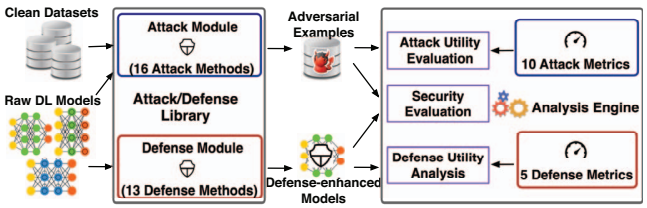

Fig. 1. The System Overview of DEEPSEC

Classification Confidence Variance (CCV). Although defense-enhanced models might not affect the accuracy performance, the prediction confidence of correctly classified examples may significantly decrease. To measure the confidence variance induced by defense-enhanced models, we formulate $CCV = \frac{1}{n}\sum_{i=1}^{n} |P(X_i)_{y_i} - P^D(X_i)_{y_i}|$, where $n < N$ is the number of examples correctly classified by both $\mathbf{F}$ and $\mathbf{F^D}$.

Classification Output Stability (COS). To measure the classification output stability between the original model and the defense-enhanced model, we use JS divergence [44] to measure the similarity of their output probability. We average the JS divergence between the output of original and defense-enhanced model on all correctly classified testing examples, i.e., $COS = \frac{1}{n}\sum_{i=1}^{n} JSD(P(X_i) \| P^D(X_i))$, where $n < N$ is the number of examples classified by both $\mathbf{F}$ and $\mathbf{F^D}$ correctly; JSD is the function of JS divergence.

III. System Design and Implementation

A. System Design

We present the system overview of DEEPSEC in Fig. 1. Basically, it consists of five parts:

1) **Attack Module (AM).** The main function of AM is to exploit vulnerabilities of DL models and attack them via crafting AEs. In this module, we implement 16 state-of-the-art adversarial attacks, including 8 UAs and 8 TAs.
2) **Defense Module (DM).** The main function of DM is to defend DL models and increase their resistance against adversarial attacks. In this module, we implement 13 latest and representative defense methods, which cover all categories of existing defenses.
3) **Attack Utility Evaluation (AUE).** In this module, we implement 10 utility metrics of adversarial attacks (as detailed in Section II-B). With AUE, users can evaluate to what extent the generated AEs satisfy the essential utility requirements of adversarial attacks.
4) **Defense Utility Evaluation (DUE).** Similar to AUE, DUE is mainly used for evaluating the utility of the state-of-the-art defenses in terms of 5 utility metrics, as defined in Section II-D. With this module, users can measure to what extent a defense-enhanced model preserves the fundamental functionality of the original model after applying all the defenses in DM.
5) **Security Evaluation (SE).** Leveraging both AM and DM modules, SE is used to evaluate the vulnerability and resilience of defense-enhanced models against existing attacks. More importantly, users can determine whether the

defense-enhanced models that are planned to deploy/share are resistant to current adversarial attacks.

To the best of our knowledge, DEEPSEC is the first implemented uniform evaluating and securing system for DL models, which comprehensively and systematically integrates the state-of-the-art adversarial attacks, defenses and relative utility metrics of them. The significance of DEEPSEC to research and application lies in the following aspects.

• First, before sharing or deploying the pre-trained DL models publicly, DEEPSEC enables model owners to conveniently and freely choose any existing defenses to secure their models. Model owners can also employ evaluation modules (i.e., DUE and SE) of DEEPSEC to examine whether the defense-enhanced models satisfy their utility/security requirements.

• Second, DEEPSEC is a uniform platform for systematically evaluating different adversarial attacks and defenses. Previously, due to the lack of such a uniform platform, existing attacks and defenses are often implemented and evaluated on different experimental settings (e.g., DL models, parameter settings, evaluation metrics, testing environment, etc.). Consequently, those implementation and evaluation differences make researchers confused about the actual utility and security performance of attacks/defenses. Some researchers even draw contradictory conclusions for the same problems. For instance, the effectiveness of DD against adversarial attacks obtains different observations in different work [10], [12], [13]. However, as a uniform evaluation platform, DEEPSEC can reduce the evaluation bias as much as possible and facilitate fair comparisons among different attacks/defenses. Therefore, DEEPSEC allows model owners to compare the performance of all possible defense-enhanced models that adopt different defenses and thus make the best decision.

• Third, DEEPSEC allows researchers to evaluate the utility and security performance of newly proposed adversarial attacks by attacking state-of-the-art defenses. Also, DEEPSEC enables researchers to compare the performance of newly proposed defenses with existing defenses as well as to examine their defenses' resistance against existing adversarial attacks. Therefore, DEEPSEC is helpful for both attack and defense research to conveniently and fairly apply existing approaches to comprehensively understand the actual performance.

In addition to providing a uniform evaluation system, DEEPSEC takes a fully modular implementation, which makes it easily extendable. First, algorithms in DEEPSEC are implemented using PyTorch [45], which has been widely used in the DL research community. Second, all modules inside DEEPSEC are independent of each other, which means that each module can work individually. Additionally, as shown in Fig. 1, multiple modules can also work together to perform rich evaluation. Third, all algorithms or evaluation tests within each module are also independent, which means that they can be implemented, measured and employed independently.

B. System Implementation

In AM, we implement 16 adversarial attacks as we summarized in Section II-A. Specifically, we cover all categories of existing attacks that include 8 UAs: FGSM [15], R+FGSM [16], BIM [17], PGD [18], U-MI-FGSM [19], DF [20], UAP [21], OM [22]; and 8 TAs: LLC [17], R+LLC [16], ILLC [17], T-MI-FGSM [19], BLB [2], JSMA [11], CW2 [10], EAD [23].

In DM, we implement 13 defense algorithms, which also cover all the categories of state-of-the-art defense algorithms summarized in Section II-C. Specifically, the implemented defense algorithms include 3 adversarial training defenses: NAT [24], EAT [16] and PAT [18]; 2 gradient masking defenses: DD [10] and IGR [13]; 4 input transformation based defenses: EIT [25], RT [26], PD [27] and TE [28]; one region-based classification defense RC [29]; as well as 3 detection-only defenses: LID [30], FS [31] and the detector of MagNet [32].

Note that for both adversarial attacks and defenses, our implementations take representativeness, scalability and practicality into consideration, which leads us to implement the latest, scalable and practical adversarial attacks and defenses.

In addition, for AUE and DUE, we implement 10 attack utility metrics (introduced in Section II-B) and 5 defense utility metrics (introduced in Section II-D), respectively.

IV. EVALUATIONS

In this section, we first evaluate the utility performance of all adversarial attacks and various defense algorithms. Then, we examine the security performance of all defenses against various adversarial attacks. Note that all experiments were conducted on a PC equipped with 2 Intel Xeon 2.2GHz CPU, 256GB system memory and one NVIDIA GTX 1080 GPU.

A. Evaluation of Attacks

1) Experimental Setup: We employ two popular benchmark datasets: MNIST [46] and CIFAR-10 [47], which have been widely used in image classification tasks. To be compatible with existing work on adversarial attacks or defenses, we train a 7-layer CNN [10] and a ResNet-20 model [48] for MNIST and CIFAR-10 (more details are shown in Appendix VIII-A), respectively. We achieve 99.27% testing accuracy on MNIST and 85.95% testing accuracy on CIFAR-10.

We present our evaluation methodology as follows. At first, we randomly sample 1000 examples that are correctly predicted by the corresponding model from each dataset's testing set. Then, for each attack in AM, we generate 1000 AEs on the sampled examples. Finally, leveraging AUE, we examine the utility performance of all attacks. Particularly, the target class for each TA is chosen randomly and uniformly among the labels except the ground truth. [2]

The criteria for parameter setting in evaluating attacks are: (i) the value of common parameters of different attacks are kept the same for unbiased comparisons, e.g., all L_∞ attacks share the same restriction ϵ. (ii) all the other parameters follow the same/similar setting in the original work for all

[2]We do not choose the target class for LLC, R+LLC and ILLC, since they inherently take the least-likely class as the target class.

attacks. The detailed parameter settings can be found in Appendix IX-A.

2) Experimental Results: We only present the evaluation results of CIFAR-10 in Table III, since the results for MNIST are similar and we defer them to Appendix X.

Misclassification. Generally, most of existing attacks, including both UAs and TAs, show strong attacking ability with high MR. More specifically, it can be observed that iterative attacks present noticeably higher MR than non-iterative attacks. Furthermore, we find that all iterative attacks, including iterative UAs and iterative TAs, have nearly 100% MR on CIFAR-10. The reason is intuitive that iterative attacks run multiple complicated iterations to find the optimal perturbation for the target model, while non-iterative attacks only take one step to compute the perturbation.

In spite of 100% MR, some adversarial attacks have low ACAC, which indicates that AEs generated by those attacks are low confident. We suggest that directly comparing the exact ACAC among all kinds of attacks can be misleading since they might have totally different parameters, e.g., it is unfair to compare ACAC between ILLC (L_∞ attack) and CW2 (L_2 attack). On the other hand, via fine-tuning the parameters of attacks, their performance can be significantly changed. For instance, if the κ of CW2 is increased from 0 to 20, ACAC of CW2 increases from 0.393 to 1.000 on CIFAR-10.

Basically, AEs with higher ACAC have lower ACTC, since the sum of probability of each class is 100%. However, if ACAC is lower than 100%, ACTC can be relatively high or low. In that case, for AEs with similar ACAC, we suggest such AEs with lower ACTC would show better resilience to other models as their true classes are less likely to be correctly classified by other models (e.g., defense-enhanced models or other raw models). For instance, both FGSM ($\epsilon = 0.1$) and OM achieve around 0.75 ACAC on CIFAR-10, but the ACTC of FGSM is 6× lower than that of OM. Hence, we conclude that FGSM shows better resilience than OM, which will later be empirically verified in following evaluations (see more in Section IV-C and Section V-A, respectively).

Remark 1. *In most cases, existing attacks show high attack success rate (i.e., MR) in terms of misleading the target model. In addition to MR, it is also important to evaluate the attacks with other metrics. For instance, we observe that AEs with low ACTC show better resilience to other models.*

Imperceptibility. We quantify and analyze the imperceptibility of AEs in terms of ALD_p, ASS and PSD.

In general, most existing attacks explore L_p norm to formulate attack algorithms in their objective functions. From Table III, we observe that attacks that use the same ALD_p metric in their attack objectives tend to perform better in that distance measurement than in other distance measurements. For instance, L_∞ attacks, perform better in L_∞ distortion, but perform poorly in both L_0 and L_2 distortions.

On the other hand, via fine-tuning parameters, L_p distortions of attacks can be easily increased for better misclassification performance. For instance, when we increase κ from 0 to 20, all L_p distortions of CW2 significantly increase. Similar observations are obtained when we increase ϵ for FGSM. The above observations suggest that there exists an objective trade-off between misclassification and imperceptibility. The trade-off stems from the mathematical framework of adversarial attacks, which are usually formulated as optimization problems with two objectives: (i) to misclassify the adversarial sample and (ii) to minimize the perceptual difference of adversarial and benign samples. As these two objectives are not always aligned, there exists a tension between misclassification and imperceptibility, which has been empirically confirmed.

Compared with ALD_p, existing attack techniques perform better at preserving ASS. On CIFAR-10, most attacks achieve nearly 100% similarity between original examples and corresponding AEs. This is because ASS is consistent with L_p norms, and thus balanced L_p norms (i.e., none of L_0, L_2 and L_∞ is extremely high) can result in high ASS. For instance, AEs generated by CW2 and EAD show moderate L_p distortions, which leads to nearly 100% similarity between original examples and AEs.

According to the results, PSD is more sensitive than ASS. Also, we observe that the PSDs of L_2 attacks are much lower than those of other attacks (i.e., L_∞ or L_0 attacks). This implies that AEs generated by L_2 attacks are more visually imperceptible than those generated by other attacks w.r.t PSD. One possible reason is that the formulation of PSD is consistent with L_2 distortion, and thus L_2 attacks outperform others in both L_2 and PSD.

Remark 2. *Among all imperceptibility metrics, PSD is the most sensitive imperceptible metric to the perturbation of AEs, while ASS is the least sensitive, which we suggest is not suitable to quantify AEs. Also, the trade-off between misclassification and imperceptibility is empirically confirmed.*

Robustness. We examine the robustness of existing attacks w.r.t three metrics (i.e., NTE, RGB, RIC). In our evaluation, we use Guetzli [49], an open source compression algorithm that creates high visual quality images. Specifically, the radius of Gaussian blur is set to 0.5 for RGB and the compression quality is set to 90% for RIC.

In general, the evaluation results of NTE, RGB and RIC are positively correlated for the adversarial attack. As shown in Table III, adversarial attacks with high NTE tend to perform better in RGB and RIC in most cases. The underlying reason could be that high NTE implies higher probability of the misclassified class, and therefore it can tolerate more transformations than AEs with smaller NTE. On the other hand, the correlation of NTE, RGB and RIC is non-linear as they measure the robustness of attacks from different perspectives. For instance, we observe that the NTE of CW2 ($\kappa = 20$) is extremely high while its RIC is quite low.

Generally, AEs with higher ACAC are shown to be more robust in RGB and RIC. This is because ACAC can influence NTE directly and thus further influence RGB and RIC since these two metrics are consistent with NTE as discussed before. Therefore, increasing the ACAC via fine-tuning parameters in

TABLE III
UTILITY EVALUATION RESULTS OF ALL ADVERSARIAL ATTACKS ON CIFAR-10

Datasets	Attack				Misclassification			Imperceptibility					Robustness			CC
	UA/ TA	Objective	Attacks		MR	ACAC	ACTC	ALD_p L_0	L_2	L_∞	ASS	PSD	NTE	RGB	RIC	
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	89.7%	0.743	0.033	0.993	5.423	0.100	0.710	276.991	0.568	0.942	0.932	0.0017
				$\epsilon=0.2$	89.8%	0.873	0.008	0.994	10.596	0.200	0.433	537.340	0.752	0.939	0.977	0.0016
			R+FGSM		83.7%	0.846	0.018	0.520	3.871	0.100	0.812	142.111	0.733	0.962	0.968	0.0017
			BIM		100.0%	1.000	0.000	0.775	2.003	0.100	0.940	79.507	1.000	1.000	0.998	0.0049
			PGD		100.0%	1.000	0.000	0.979	3.682	0.100	0.827	165.721	1.000	1.000	1.000	0.0227
			U-MI-FGSM		100.0%	1.000	0.000	0.919	3.816	0.100	0.817	171.691	1.000	1.000	1.000	0.0056
			UAP		85.3%	0.723	0.038	1.000	5.335	0.100	0.717	275.353	0.527	0.904	0.907	-*
		L_2	DF		100.0%	0.516	0.458	0.135	0.078	0.010	1.000	2.692	0.058	0.064	0.226	0.0113
			OM		100.0%	0.750	0.182	0.274	0.192	0.022	0.999	5.485	0.541	0.929	0.908	20.4526
	TAs	L_∞ $\epsilon=0.1$	LLC		13.4%	0.768	0.016	0.992	5.400	0.100	0.730	273.682	0.594	0.620	0.630	0.0009
			R+LLC		31.5%	0.876	0.009	0.531	3.897	0.100	0.825	143.450	0.763	0.635	0.548	0.0006
			ILLC		100.0%	1.000	0.000	0.764	1.829	0.100	0.946	73.204	0.909	1.000	0.500	0.0033
			T-MI-FGSM		100.0%	1.000	0.000	0.937	4.063	0.100	0.799	187.717	1.000	1.000	0.993	0.0047
		L_0	JSMA		99.7%	0.508	0.164	0.022	4.304	0.879	0.832	32.445	0.238	0.321	0.224	2.6207
		L_2	BLB		100.0%	0.500	0.349	0.218	0.111	0.013	1.000	3.882	0.141	0.007	0.025	90.6852
			CW2	$\kappa=0$	100.0%	0.393	0.348	0.230	0.112	0.013	1.000	3.986	0.032	0.009	0.023	3.0647
				$\kappa=20$	100.0%	1.000	0.000	0.557	0.279	0.031	0.998	10.157	1.000	0.736	0.049	4.6315
			EAD	EN	100.0%	0.433	0.316	0.106	0.156	0.033	0.999	2.457	0.093	0.015	0.027	4.5115
				L1	100.0%	0.377	0.352	0.041	0.185	0.057	0.999	1.642	0.014	0.014	0.030	4.7438

* Since UAP takes different settings and significantly longer time to generate the universal perturbation, here we do not consider the computation cost for it.

one attack can improve its robustness as well.

Compared to TAs, most UAs are shown to be more robust to regular transformations. For instance, on CIFAR-10, most UAs achieve over 90%, even 100% robustness in both RGB and RIC. This implies that almost all AEs generated by such attacks can maintain its capability of misclassification as before. On the other hand, most TAs, especially BLB, JSMA, CW and EAD, are shown to experience the worst robustness in RGB and RIC. It suggests that regular transformations are effective for mitigating such attacks. The root reason we conjecture is that with specifying the target class, TAs are more difficult to attack than UAs. Therefore, it is difficult for TAs to obtain higher ACAC, which affects their robustness.

Remark 3. *The robustness of AEs is affected by ACAC. Further, most UAs are shown to be more robust than TAs in our evaluation. Even for certain TAs, image transformations can effectively mitigate the added perturbation.*

Computation Cost. To evaluate the computation cost of attacks, we test their runtime that is used to generate one AE on average. It is important to know that comparing the exact runtime of attacks is unfair due to multiple and complex factors (e.g., programming, parallelized computing or not, etc.), which can lead to different runtime performance. Therefore, we keep all attacks' settings unchanged with their original work and only give empirical results in our evaluations.

It is apparent that in the majority of cases, AEs of iterative attacks are much more expensive to generate than those of non-iterative attacks. On average, iterative attacks spend $10\times$ more runtime than non-iterative attacks. Among all iterative attacks, we observe that OM, JSMA, BLB, CW2 and EAD are noticeably slower than other iterative attacks.

B. Evaluation of Defenses

We evaluate the utility performance of defenses as below.

1) Experimental Setup: We use the same benchmark datasets and models as used in Section IV-A. The evaluation methodology is as follows. Firstly, for each complete defense, we obtain the corresponding defense-enhanced model from the original model. Using utility metrics in DUE, we then compare the utility performance of the defense-enhanced model with the original model on the testing set of each dateset. For defense parameter settings in our evaluations, the criteria are: (i) we follow the same/similar setting as in the original work of defenses; (ii) if there are variants for one defense, we choose the one with the best effectiveness performance. The details of defense parameter settings are reported in Appendix IX-B.

2) Results: We present the evaluation results in Table IV.

TABLE IV
UTILITY EVALUATION RESULTS OF ALL COMPLETE DEFENSES

Datasets	Defense-enhanced Models		Accuracy	CAV	CRR	CSR	CCV	COS
	Category	Name						
MNIST	Adversarial Training	NAT	99.51%	0.24%	0.44%	0.20%	0.17%	0.0006
		EAT	99.45%	0.18%	0.44%	0.26%	0.19%	0.0007
		PAT	99.36%	0.09%	0.39%	0.30%	0.33%	0.0012
	Gradient Masking	DD	99.27%	0.00%	0.41%	0.41%	0.14%	0.0005
		IGR	99.09%	-0.18%	0.32%	0.50%	3.03%	0.0111
	Input Transform.	EIT	99.25%	-0.02%	0.42%	0.44%	0.26%	0.0010
		RT	95.65%	-3.62%	0.17%	3.79%	1.24%	0.0048
		PD	99.24%	-0.03%	0.06%	0.09%	0.09%	0.0002
		TE	99.27%	0.00%	0.42%	0.42%	0.55%	0.0020
	RC		99.27%	0.00%	0.07%	0.07%	-	-
CIFAR-10	Adversarial Training	NAT	84.41%	-1.54%	7.14%	8.68%	4.81%	0.0197
		EAT	82.15%	-3.80%	6.50%	10.30%	5.37%	0.0215
		PAT	80.23%	-5.72%	6.60%	12.32%	13.87%	0.0572
	Gradient Masking	DD	87.62%	1.67%	7.34%	5.67%	3.25%	0.0127
		IGR	77.10%	-8.85%	6.56%	15.41%	18.80%	0.0788
	Input Transform.	EIT	83.25%	-2.45%	6.94%	9.39%	5.99%	0.0239
		RT	78.50%	-7.45%	3.22%	10.67%	4.63%	0.0171
		PD	70.66%	-15.29%	3.02%	18.31%	5.82%	0.0221
		TE	88.63%	2.68%	8.13%	5.45%	4.36%	0.0173
	RC		84.87%	-1.08%	1.53%	2.61%	-	-

Although all defenses achieve comparable performances on both MNIST and CIFAR-10, most defenses show that their defense-enhanced models have variances for classification accuracy on the testing set, which can be found in "Accuracy" and "CAV" columns of Table IV. Particularly, the accuracy variances of defense-enhanced models on CIFAR-10 are much higher than those on MNIST. The reason we conjecture is that

the 99.27% accuracy of the original model on MNIST has already been sufficiently high and stabilized, and thus leaves less variation space than that on CIFAR-10 models.

Among all defenses, NAT, DD, TE and RC on both MNIST and CIFAR-10 almost do not sacrifice the classification accuracy on the testing set according to the CAV results. On the other hand, it can also be observed that the classification accuracies of IGR-, RT- and PD-enhanced models have significant drop when they are performed on CIFAR-10. In fact, the performance of CAV is a consequence of the percentage of examples that are rectified and sacrificed by the defended model, which is confirmed by both CRR and CSR results in corresponding columns of Table IV. Consequently, the significant accuracy drop (i.e., CAV) is induced in the defense-enhanced model as long as its CSR is larger than CRR.

According to the CCV results, most of defenses have little impact on the prediction confidence of all correctly classified testing examples before and after applying defenses. However, compared to MNIST, the CCVs of defense-enhanced models on CIFAR-10 are $10\times$ higher in most cases. Even worse, for PAT and IGR, the CCV of their defense-enhanced models on CIFAR-10 are 18.80% and 13.87%, respectively. This is because the defense-enhanced models on CIFAR-10 are less stable, and thus their prediction confidence is more sensitive to examples on the testing set.

As for the classification output stability of all defense-enhanced models, we find that COS has the similar trend to CCV. The reason we conjecture is that if the confidence of the predicted class is fairly high for one example, the confidences of other classes are quite small, accordingly. This implies high variance of the predicted class's probability can lead to considerable adjustments of all other output probabilities. Therefore, the prediction confidence variance greatly impacts its output stability of classification, and thus the values of both COS and CCV follow a similar trend.

Remark 4. *Overall, as long as the defense-enhanced models are trained or adjusted based on the accuracy metric, most of them can also preserve the other utility performances, such as CCV and COS.*

C. Defenses vs. Attacks

Although there have been many sophisticated defenses and strong attacks, it is still an open problem whether or to what extent the state-of-the-art defenses can defend against attacks.

1) Complete Defenses: In this part, we evaluate the effectiveness of all 10 complete defenses against attacks.

Experimental Setup. We use the same benchmark datasets and their corresponding models as that in Section IV-A. The evaluation methodology proceeds as follows. For each attack, we first merely select the successfully misclassified AEs that are generated in Section IV-A, and then we use all defense-enhanced models (as used in Section IV-B) to evaluate the classification performance on such AEs.

Results. We only present the results of CIFAR-10 in Table V, as the results of MNIST are similar (detailed results are reported in Appendix X).

Basically, most defense-enhanced models increase their classification accuracy against existing attacks. When evaluating on CIFAR-10, NAT can successfully defend against more than 80% AEs generated by all attacks on average, and all defense-enhanced models averagely achieve 58.4% accuracy over all kinds of AEs. Thus, we suggest that all state-of-the-art defenses are more or less effective against existing attacks.

In general, most defenses show better defensive performance against TAs than that on UAs. For CIFAR-10, all defense-enhanced models averagely achieve 49.6% and 66.3% accuracy against UAs and TAs, respectively. It implies that AEs generated by UAs show stronger resilience to defense-enhanced models, and thus become more difficult to defend. We conjecture this is because UAs are more likely to generalize to other models including defense-enhanced models, while TAs tend to overfit to the specific target model. Hence, AEs generated by TAs are more easily classified by defenses.

With the increase of attacking ability for one specific attack, fewer AEs can be classified by defense-enhanced models. For instance, when we increase the ϵ of FGSM, we find that the performance of all defenses has a significant drop. Similar results are found when we increase the κ parameter of CW2. The reason is evident. Since larger attacking ability implies that higher magnitude of perturbations are generated by attacks, which make the perturbed AEs more visually dissimilar to the original examples.

Among all defenses, NAT, PAT, EAT, TE, EIT and IGR show better and stable performance in defending against most attacks. RT, PD and RC are observed to have worse performance when defending against each attack on average for both MNIST and CIFAR-10. We conjecture this is mainly because they all retrain their model and obtain totally different model weights. As we will present in Section V-A, without any modification to the original model, merely retraining the model can be a defense. Therefore, a defense that retrains the model usually performs better than other defenses that do not retrain their models, including RC, RT, PD.

According the results, all the defenses have the capability of defending against some attacks, while no defense is universal to all attacks. Taking the RC defense as an example, we find that the RC has superior performance to defend against DF, BLB, EAD and low-confidence CW2 (i.e., $\kappa = 0$), but it achieves much worse performance on the other adversarial attacks. Multiple reasons are responsible for the results such as inherent limitations of defenses against different kinds of attacks (e.g., RC defense is designed to defend against small perturbations), the parameters employed by an algorithm, etc.

Remark 5. *For complete defenses, most of them have capability of defending against some adversarial attacks, but no defense is universal. Particularly, the defenses that retrain their models usually perform better than others without retraining.*

2) Detection: Now, we evaluate the effectiveness of three detection-only defenses against existing adversarial attacks.

Experimental Setup. We use the same benchmark datasets and relative original models as used in Section IV-C1.

TABLE V
CLASSIFICATION ACCURACY OF COMPLETELY DEFENSES AGAINST ADVERSARIAL ATTACKS ON CIFAR-10

Datasets	Attack				# of AEs	Original Model	Defense-enhanced Models										Average
	UA/TA	Objective	Attacks				Adversarial Training			Gradient Mask.		Input Transformation				RC	
							NAT	EAT	PAT	DD	IGR	EIT	RT	PD	TE		
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	897	0.0%	76.4%	57.0%	51.0%	7.9%	69.9%	66.7%	7.6%	6.9%	32.6%	0.1%	37.6%
				$\epsilon=0.2$	898	0.0%	52.7%	28.4%	18.5%	10.7%	51.6%	35.0%	4.8%	2.5%	13.3%	0.1%	21.7%
			R+FGSM		837	0.0%	82.7%	80.4%	75.3%	12.2%	76.9%	79.9%	4.2%	4.5%	68.1%	0.0%	48.4%
			BIM		1000	0.0%	84.7%	81.1%	82.4%	3.8%	77.4%	82.3%	0.0%	2.5%	76.5%	0.0%	49.1%
			PGD		1000	0.0%	81.9%	77.8%	74.3%	0.7%	75.1%	79.7%	0.0%	0.2%	64.7%	0.0%	45.4%
			U-MI-FGSM		1000	0.0%	78.7%	65.5%	69.5%	2.7%	73.0%	69.6%	0.0%	0.0%	47.5%	0.0%	40.7%
			UAP		853	0.0%	80.9%	79.8%	60.8%	5.4%	74.4%	71.4%	3.8%	21.3%	47.8%	1.4%	44.7%
		L_2	DF		1000	0.0%	88.9%	86.6%	83.3%	89.3%	79.2%	87.1%	83.2%	74.9%	92.9%	91.3%	85.7%
			OM		1000	0.0%	88.9%	86.1%	82.3%	81.6%	79.0%	87.4%	52.2%	70.5%	91.1%	14.8%	73.4%
	TAs	L_∞ $\epsilon=0.1$	LLC		134	0.0%	79.9%	65.7%	61.2%	1.5%	76.9%	70.2%	3.0%	6.0%	29.9%	0.0%	39.4%
			R+LLC		315	0.0%	84.4%	86.0%	81.3%	6.7%	81.9%	86.0%	4.1%	5.1%	73.3%	0.0%	50.9%
			ILLC		1000	0.0%	86.6%	85.3%	83.7%	27.6%	78.2%	86.9%	0.9%	49.7%	88.5%	0.0%	58.7%
			T-MI-FGSM		1000	0.0%	83.1%	71.4%	70.2%	11.2%	74.5%	78.5%	0.8%	0.0%	61.4%	0.0%	45.1%
		L_0	JSMA		997	0.0%	68.0%	75.1%	72.7%	50.3%	73.5%	70.0%	37.1%	27.1%	75.5%	16.2%	56.6%
		L_2	BLB		1000	0.0%	89.1%	86.4%	83.0%	89.8%	79.2%	87.4%	83.9%	74.1%	92.8%	91.1%	85.7%
			CW2	$\kappa=0$	1000	0.0%	88.8%	86.5%	83.0%	89.5%	79.2%	88.6%	82.9%	76.7%	92.5%	90.2%	85.8%
				$\kappa=20$	1000	0.0%	88.6%	86.3%	82.3%	82.8%	79.2%	88.0%	26.5%	74.4%	92.2%	14.6%	71.5%
			EAD	EN	1000	0.0%	88.5%	86.5%	82.5%	89.2%	79.1%	88.0%	79.3%	74.8%	92.7%	87.5%	84.8%
				L1	1000	0.0%	88.4%	86.6%	82.6%	88.4%	79.0%	86.3%	81.0%	76.2%	92.6%	88.4%	85.0%
	Average				891.1	0.0%	**82.2%**	76.8%	72.6%	39.5%	75.6%	78.4%	29.2%	34.1%	69.8%	26.1%	**58.4%**

The evaluation methodology proceeds as follows. Firstly, for each attack, we select all successfully misclassified AEs that are generated in Section IV-A. Then, to make the dataset balanced for detection, we randomly select the same number of normal examples from the testing set to build a mixed set for each attack. To eliminate biases in our evaluation, all selected normal examples can be correctly recognized by the original model. Finally, we examine the effectiveness of the three detection-only defenses against all kinds of attacks. For the parameter settings of detection, we mainly follow the same or similar settings as in their original papers. The details of their parameter settings are deferred to Appendix IX-B.

Results. Due to the space limitation, we only present the results of CIFAR-10 in Table VI (detailed results of MNIST are reported in Appendix X) and analyze them as follows.

To measure the overall detection performance, we calculate their AUC scores as AUC is independent with the manually selected threshold. According to the results, all detection methods can yield fairly high AUC scores against most attacks. The average AUCs of the three detection methods are all higher than 70%, i.e., they show comparable discriminative power against existing attacks. Specifically, LID has the best performance in terms of AUC than others in most cases. However, even for the best detection method LID on CIFAR-10, it almost fails to detect AEs generated by DF and OM, with AUC about 65%, which is lower than that of FS or MagNet (over 80% on average).

In addition to AUC, we also evaluate the true positive rate (TPR) and false positive rate (FPR) of different detection methods on the mixed testing set. In order to fairly compare the detection rate (i.e., TPR), we try our best to adjust the FPR values of all detection methods to the same level via fine-tuning the parameters. In our evaluations, we first set the FPR of all the detection methods to around 4%, and then compare their TPRs.

According to the results, LID has the highest average TPR against all kinds of AEs. Although FS and MagNet have higher average TPRs on MNIST (i.e., more than 90%, see Appendix X for details), their average TPRs on CIFAR-10 are much lower. One possible reason we conjecture is that we only choose one threshold for each detection method to discriminate diverse AEs generated by all attacks. We suggest that we can improve the TPR performance within an acceptable FPR of the detection method via fine-tuning the parameters or adjusting the threshold. For instance, we believe the TPRs of FS against DF, BLB, CW2 and EAD can be significantly increased since their corresponding AUC scores are much higher (all over 86%).

For detection-only defense, it is hypothesized that AEs with higher magnitude of perturbation are easier to be detected since most detection methods are based on the difference between normal and adversarial examples. To better understand the influence of the perturbation of AEs on detection, we conduct a simple test on FGSM with different ϵ, and the results are shown in Table VII. According to the results, we observe that there is no clear relationship between the magnitude of perturbation of AEs and detection AUC. Thus, we argue that we cannot conclude AEs with higher magnitude of perturbation are easier to be detected.

Remark 6. *All detection methods show comparable discriminative ability against existing attacks. Different detection methods have their own strengths and limitations facing various kinds of AEs. It is not the case that AEs with high magnitude of perturbation are easier to be detected.*

V. CASE STUDIES

To further demonstrate the functionality of DEEPSEC as a uniform and comprehensive analysis platform, we present two case studies in this section.

A. Case Study 1: Transferability of Adversarial Attacks

The transferability is an intriguing property that AEs generated against one target model can also be misclassified by other models. Although there has been several literature

TABLE VI
EVALUATION RESULTS OF DETECTION-ONLY DEFENSES AGAINST ALL ADVERSARIAL ATTACKS

Dataset	Attack			# of Examples	Detection-only Defenses								
					LID			FS			MagNet		
	UA/TA	Objective	Attacks		TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR	AUC
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	1794	100.0%	5.1%	100.0%	9.5%	2.9%	82.6%	99.1%	4.7%	93.5%
			R+FGSM	1674	100.0%	2.9%	100.0%	6.0%	4.8%	70.7%	33.3%	3.2%	83.2%
			BIM	2000	94.6%	2.9%	99.1%	1.6%	4.5%	25.5%	1.8%	4.2%	53.0%
			PGD	2000	99.9%	3.5%	100.0%	0.4%	3.8%	16.5%	3.2%	4.3%	59.2%
			U-MI-FGSM	2000	100.0%	3.0%	100.0%	1.8%	4.1%	23.8%	6.3%	4.1%	57.1%
			UAP	1706	100.0%	5.3%	100.0%	2.9%	3.8%	76.3%	99.5%	5.9%	94.9%
		L_2	DF	2000	9.2%	5.7%	64.0%	1.5%	3.9%	86.3%	21.5%	2.8%	81.0%
			OM	2000	8.8%	4.9%	65.1%	25.0%	3.8%	89.0%	46.4%	3.9%	78.7%
	TAs	L_∞ $\epsilon=0.1$	LLC	268	100.0%	1.5%	100.0%	3.7%	9.0%	73.5%	100.0%	6.7%	91.8%
			R+LLC	630	99.0%	5.7%	99.2%	11.7%	5.1%	71.0%	31.4%	3.8%	81.2%
			ILLC	2000	79.2%	5.3%	96.1%	51.7%	3.3%	83.9%	2.6%	4.7%	61.2%
			T-MI-FGSM	2000	100.0%	5.8%	100.0%	10.0%	3.8%	45.0%	10.4%	3.8%	57.9%
		L_0	JSMA	1994	71.5%	3.4%	94.4%	20.6%	3.7%	91.7%	53.2%	5.3%	92.3%
		L_2	BLB	2000	13.0%	3.1%	72.3%	1.7%	4.1%	89.3%	52.5%	4.3%	81.6%
			CW2	2000	19.9%	3.8%	77.6%	0.9%	3.7%	88.1%	38.4%	4.4%	81.8%
			EAD (EN)	2000	17.2%	4.0%	73.8%	1.9%	3.5%	89.8%	54.2%	5.0%	82.1%
			EAD (L1)	2000	23.0%	5.7%	76.3%	1.1%	3.8%	86.4%	35.6%	4.3%	81.4%
	AVERAGE			1768.6	66.8%	**4.2%**	89.3%	8.9%	**4.2%**	70.0%	40.6%	**4.4%**	77.2%

TABLE VII
DETECTION-ONLY DEFENSES AGAINST FGSM WITH DIFFERENT ϵ

Datasets	FGSM		AUC		
	ϵ	# of examples	LID	FS	MagNet
MNIST	0.1	138	90.4%	99.8%	100.0%
	0.2	432	85.0%	99.4%	100.0%
	0.3	608	93.7%	99.1%	100.0%
	0.4	734	97.7%	99.0%	100.0%
	0.5	896	98.2%	99.0%	100.0%
	0.6	1032	98.7%	99.0%	100.0%
CIFAR-10	0.1	1794	100.0%	82.6%	93.5%
	0.2	1796	95.2%	89.7%	98.8%
	0.3	1820	39.6%	58.7%	99.0%
	0.4	1820	15.7%	31.5%	96.6%
	0.5	1820	6.4%	21.6%	91.8%
	0.6	1820	6.6%	17.8%	87.1%

discussing the transferability of AEs [2], [18], [50], [51], no work comprehensively evaluates the transferability of AEs generated by existing attacks. In this case study, we conduct a series of experiments on existing attacks and compare their transferability performance on different target models.

1) Experimental Setup: We use the same benchmark datasets and corresponding original models as that in Section IV-A. Besides, we prepare three additional DL models:

- **Model 1**: We train **model 1** that is identical to the original model, but with different random initializations.
- **Model 2**: We train **model 2** that keeps the same configurations as the original DL model, except the network architecture is slightly different.
- **Model 3**: We train **model 3** as a totally different model.

The evaluation methodology is that we first independently train the three above models on each dataset with comparable accuracy. Then, we employ the three trained models to classify the misclassified AEs generated in Section IV-A. Finally, to compare the transferability of adversarial attacks, we evaluate the MR and ACAC of each model for each dataset.

2) Results: We present the results of CIFAR-10 in Table VIII, and the conclusions on MNIST are similar (detailed results are reported in Appendix XI).

Apparently, all adversarial attacks show more or less transferability on other models. As shown in Table VIII, the transferability rates of most attacks on the three models are over 10%. Moreover, the average transferability rate of all attacks on the three models is 42.4%. This empirically confirms the existence of transferability of all adversarial attacks.

In particular, the confidence (ACAC) of AEs that successfully transfer to other models is higher than that of AEs that are misclassified on the original model. For instance, on CIFAR-10 the average confidence of AEs that transfer to the three models is 0.812, while the ACAC of AEs misclassified by the original model is 0.751. This may be explained as since successfully transferred AEs are part of AEs misclassified by the original model and low-confidence AEs usually fail to transfer, the confidence of transferable AEs is selectively higher.

For the impact of model diversity, we observe that the attack transferability differs marginally across different target models (i.e., **model 1**, **model 2** and **model 3**). On CIFAR-10, all three models averagely achieve approximately 42% transferability rate. It indicates that the transferability of AEs is independent of the model architecture, which confirms the finding in [50].

In general, different kinds of attacks tend to have different transferability performance, which implies different attack abilities under black-box scenarios. To be specific, the transferability differences of different attacks have two facets. First, AEs generated by UAs are more transferable than those of TAs. For CIFAR-10, the average transferability rate of UAs is 74.6%, which is much higher than that of TAs (i.e., 10.0%). This confirms the conclusion in [51]. Secondly, for both UAs and TAs, L_∞ attacks are much more transferable than others (i.e., L_2 and L_0 attacks). In particular, we observe that the average transferability rate of all L_∞ UAs (i.e., more than 90%) is several times higher than other UAs on CIFAR-10. Similar results are observed in TAs. We conjecture that one possible reason is that L_∞ attacks tend to perturb every pixel of the original image with the L_∞ constraint, and thus the AEs generated by them are more perceptible than others, which can be observed and confirmed in Table III.

Remark 7. *Different attacks have different transferability: (i) we confirm that UAs are more transferable than TAs; (ii) we find that L_∞ attacks are more transferable than other attacks (i.e., L_2 and L_0 attacks). Furthermore, the confidence of AEs that can transfer to other models is higher than that of AEs that can only be misclassified by the original model.*

B. Case Study 2: Is Ensemble of Defenses More Robust?

For classification tasks, ensemble methods are widely used in research and competitions to improve the performance [52], [53]. Recently, the idea of ensemble has been used to defend against adversarial attacks [16], [31], [54]–[56]. However, the effectiveness of ensemble against adversarial attacks is still chaotic: some believes that ensemble of multiple diverse classifiers (e.g., clean or defense-enhanced models) can increase

TABLE VIII
TRANSFERABILITY RATE OF ALL ADVERSARIAL ATTACKS ON CIFAR-10

Datasets	Attack				# of AEs	Original Model		Model 1		Model 2		Model 3		Average MR of 3 Models	Average ACAC of 3 Models
	UA/ TA	Objective	Attacks			MR	ACAC	MR	ACAC	MR	ACAC	MR	ACAC		
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	897	100.0%	0.743	88.2%	0.922	88.4%	0.841	92.4%	0.674	89.7%	0.812
				$\epsilon=0.2$	898	100.0%	0.873	86.3%	0.878	91.0%	0.804	93.3%	0.731	90.2%	0.804
			R+FGSM		837	100.0%	0.846	89.1%	0.863	88.8%	0.830	90.0%	0.699	89.3%	0.797
			BIM		1000	100.0%	1.000	96.2%	0.940	96.0%	0.949	93.5%	0.893	95.2%	0.927
			PGD		1000	100.0%	1.000	98.2%	0.933	99.4%	0.952	98.8%	0.914	98.8%	0.933
			U-MI-FGSM		1000	100.0%	1.000	97.0%	0.934	97.9%	0.926	97.0%	0.841	97.3%	0.900
			UAP		853	100.0%	0.723	87.0%	0.868	88.5%	0.744	90.4%	0.682	88.6%	0.764
		L_2	DF		1000	100.0%	0.516	13.1%	0.769	10.7%	0.788	11.5%	0.705	11.8%	0.754
			OM		1000	100.0%	0.750	22.7%	0.788	20.9%	0.805	19.2%	0.721	20.9%	0.771
	Average of UAs				942.8	100.0%	0.828	74.7%	0.876	74.3%	0.848	74.9%	0.762	**74.6%**	0.829
	TAs	L_∞ $\epsilon=0.1$	LLC		134	100.0%	0.768	3.7%	0.871	14.2%	0.770	17.9%	0.632	11.9%	0.758
			R+LLC		315	100.0%	0.876	18.1%	0.826	24.1%	0.828	35.2%	0.774	25.8%	0.809
			ILLC		1000	100.0%	1.000	25.3%	0.875	27.0%	0.863	23.0%	0.823	25.1%	0.854
			T-MI-FGSM		1000	100.0%	1.000	42.7%	0.893	43.0%	0.916	34.0%	0.870	39.9%	0.893
		L_0	JSMA		997	100.0%	0.508	7.7%	0.833	9.0%	0.769	7.6%	0.755	8.1%	0.786
		L_2	BLB		1000	100.0%	0.500	1.8%	0.778	1.6%	0.783	1.5%	0.737	1.6%	0.766
			CW2	$\kappa=0$	1000	100.0%	0.393	1.8%	0.751	1.5%	0.815	1.4%	0.759	1.6%	0.775
				$\kappa=20$	1000	100.0%	1.000	6.2%	0.821	7.8%	0.860	5.6%	0.722	6.5%	0.801
			EAD	EN	1000	100.0%	0.433	1.9%	0.768	2.1%	0.764	1.6%	0.741	1.9%	0.758
				L1	1000	100.0%	0.377	1.9%	0.781	2.1%	0.749	1.4%	0.781	1.8%	0.770
	Average of UAs				844.6	100.0%	0.685	9.7%	0.811	10.6%	0.809	9.5%	0.760	**10.0%**	0.794
	Totally Average				891.1	100.0%	0.753	41.5%	0.847	42.8%	0.829	42.9%	0.761	**42.4%**	0.812

its resistance against attacks [31], [54], while others hold the negative opinion [55]. To figure out the effectiveness of different ensemble defenses against attacks, in this case study, we evaluate the classification performance (i.e., accuracy and confidence) of three different ensemble methods.

1) Experiment Setup: We use the same benchmark datasets and corresponding original models as used in Section IV-A. In addition, we use three ensemble methods as follows.

Completely-random Ensemble: randomly select 3 defenses from all 9 complete defenses.[3]

Interclass-random Ensemble: randomly select 1 defense separately from 3 categories of complete defenses and thus a total of 3 defenses are selected.

Best-defense Ensemble: select the best three defenses that outperform others in defending against various adversarial attacks. As analyzed in Section IV-C1, PAT, TE and NAT are the best three defenses for MINST. For CIFAR-10, NAT, EIT and EAT are the best on average.

The experimental methodology proceeds as follows. First, we prepare successful AEs that are generated by each attack in Section IV-A. For each ensemble method, we get 3 selected defense-enhanced models from Section IV-B. Finally, for each testing AE, we predict it by letting each defense-enhanced model votes for a label, i.e., $y^{en} = \arg\max_k \sum_{i=1}^{3} P_i(x)_k$. Additionally, to avoid accidental phenomena in random-based ensemble methods, we independently repeat the first two ensemble methods 3 times and calculate the average.

2) Results: Table IX shows the results of CIFAR-10, and similar results on MNIST are reported in Appendix XI.

Generally, different ensemble methods show different defensive performance. Among the three ensembles, the completely-random ensemble performs the worst while the best-defense ensemble performs the best w.r.t accuracy and confidence. The reason is that the performance of ensemble mainly depends on the individual defense, and thus the best-defense ensemble outperforms others.

[3]We exclude RC in our ensemble methods as it does not provide confidence information for the testing examples.

TABLE IX
PERFORMANCE OF DIFFERENT ENSEMBLE METHODS ON CIFAR-10

Dataset	Attack				# of AEs	Original model	Ensemble Methods					
	UA/ TA	Objective	Attacks				Completely-random		Interclass-random		Best-defense	
							Acc.	Conf.	Acc.	Conf.	Acc.	Conf.
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	897	0%	35%	0.56	57%	0.63	73%	0.80
				$\epsilon=0.2$	898	0%	26%	0.56	28%	0.63	42%	0.72
			R+FGSM		837	0%	40%	0.57	80%	0.67	87%	0.87
			BIM		1000	0%	29%	0.58	78%	0.68	88%	0.88
			PGD		1000	0%	24%	0.62	73%	0.66	85%	0.86
			U-MI-FGSM		1000	0%	24%	0.58	62%	0.64	78%	0.83
			UAP		853	0%	41%	0.56	76%	0.66	83%	0.85
		L_2	DF		1000	0%	93%	0.85	91%	0.83	91%	0.91
			OM		1000	0%	88%	0.79	90%	0.84	92%	0.90
	Average of UAs				943	0%	44%	0.63	71%	0.70	80%	0.85
	TAs	L_∞ $\epsilon=0.1$	LLC		134	0%	30%	0.54	66%	0.63	78%	0.83
			R+LLC		315	0%	39%	0.55	86%	0.68	91%	0.89
			ILLC		1000	0%	63%	0.61	87%	0.76	91%	0.90
			T-MI-FGSM		1000	0%	34%	0.58	71%	0.66	84%	0.85
		L_0	JSMA		997	0%	6%	0.70	77%	0.75	76%	0.85
		L_2	BLB		1000	0%	93%	0.85	92%	0.85	91%	0.91
			CW2	$\kappa=0$	1000	0%	93%	0.85	92%	0.86	92%	0.91
				$\kappa=20$	1000	0%	87%	0.77	91%	0.85	91%	0.91
			EAD	EN	1000	0%	93%	0.84	92%	0.85	92%	0.91
				L1	1000	0%	93%	0.8	92%	0.85	91%	0.91
	Average of TAs				845	0%	69%	0.71	84%	0.77	88%	0.89
	Average				891	0%	**57%**	0.67	78%	0.74	84%	0.87

We observe that ensemble of different defenses does not perform better than each individual defense on average. According to the results, the completely-random ensemble averagely achieves 57% accuracy on all attacks, which is comparable to 58.4% of all defenses as previously shown in Table V. Even for the best-defense ensemble, it does not significantly improve the accuracy than the most successful defense. Particularly, the average accuracy of best-defense ensemble against all attacks is 84%, which is marginally greater than 82.2% achieved by the most successful defense NAT on CIFAR-10. This partially confirms the conclusion that ensemble of multiple defenses does not guarantee to perform better [55].

On the other hand, ensemble of multiple defenses can improve the lower bound of defense ability for individuals

against certain adversarial attacks. According to the results, there is no extremely low classification accuracy for ensemble models. As we analyzed in Section IV-C1, most individual defenses have the capability of defending against some specific adversarial attacks but not all attacks. For some individual defenses, their classification accuracy on some attacks even drops significantly below 10%, but not the case with ensemble. Even in the worst ensemble of completely-random, their classification accuracy on CIFAR-10 are above 20%.

Remark 8. *For ensemble methods, we confirm that ensemble of different defenses cannot significantly improve the defensive capabilities as a whole [55], but it can improve the lower bound of defense ability for individuals.*

VI. Discussion

Limitations and Future Work. Below, we discuss the limitations of this work along with the future work.

Firstly, we only integrate the most representative 16 adversarial attacks and 13 defenses. Even though we do cover all the categories of the state-of-the-art attacks and defenses, DeepSec does not enumerate and implement all strategies due to the fact some strategies have similar methodology. However, DeepSec employs a modular design and implementation, which makes it easy for users to integrate new attacks, defenses and corresponding utility metrics. Hence, we open source DeepSec and encourage the public to contribute.

Secondly, due to space limitations, we employ one setting for each individual attack and defense. To be specific, the exclusive parameters among different attacks or defenses are kept with the same or similar with the original settings in the papers, while the common parameters are kept the same for fair comparison. However, based on DeepSec, it is easy to extend the evaluations to different settings.

Finally, in the current implementation, we mainly focus on non-adaptive and white-box attacks. Nevertheless, we emphasize that the modular and generic design of DeepSec enables it to be readily extendable to support many other adversarial attacks via controlling the information available to the attacks/defenses. For instance, adaptive attacks [9], [35] are easily incorporated into DeepSec if the adversary is allowed to access the deployed defense-enhanced model when generating AEs; to support black-box attacks [33], [34], we may restrict the attacks' access to only the input and output of DL models; for unsupervised learning (e.g., generative models) [57], [58], we may disable the attacks' access to the label information. As the modular implementation of DeepSec provides standard interfaces for accessing data and models, such extensions can be readily implemented.

Additional Related Work. Currently, several attack/defense platforms have been proposed, like Cleverhans [14], Foolbox [59], AdvBox [60], ART [61], etc. Cleverhans is the first open-source library that mainly uses Tensorflow [62] and currently provides implementations of 9 attacks and 1 simple adversarial training based defense. Foolbox improves upon Cleverhans by interfacing it with other popular DL frameworks such as PyTorch [45], Theano [63], and MXNet [64]. Advbox is implemented on the PaddlePaddle [65] and includes 7 attacks. ART also provides a library that integrates 7 attacks and 5 defenses. However, DeepSec differs from the exiting work in several major aspects:

1) Existing platforms provide a fairly limited number of adversarial attacks and only few of them implement defense methods. However, DeepSec incorporates 16 attacks and 13 defenses, covering all the categories of the state-of-the-art attacks and defenses.
2) In addition to a rich implementation of attacks/defenses. DeepSec treats evaluation metrics as the first-class citizens and implements 10 attack and 5 defense utility metrics, which help assess given attacks/defenses.
3) Rather than solely providing reference implementation of attacks/defenses, DeepSec provides a unique analysis platform, which enables researchers and practitioners to conduct comprehensive and informative evaluation on given attacks, defences, and DL models.

VII. Conclusion

We design, implement and evaluate DeepSec, a uniform security analysis platform for deep learning models. In its current implementation, DeepSec incorporates 16 state-of-the-art adversarial attacks with 10 attack utility metrics and 13 representative defenses with 5 defense utility metrics. To our best knowledge, DeepSec is the first-of-its-kind platform that supports uniform, comprehensive, informative, and extensible evaluation of adversarial attacks and defenses. Leveraging DeepSec, we conduct extensive evaluation on existing attacks and defenses, which help answer a set of long-standing questions. We envision that DeepSec is able to serve as a useful benchmark to facilitate adversarial deep learning research.

Acknowledgment

We would like to thank our shepherd Christopher Kruegel and the anonymous reviewers for their valuable suggestions for improving this paper. We are also grateful to Xiaoyu Cao, Jacob Buckman and Yang Song for sharing their code, and to Yuan Chen and and Saizhuo Wang for helping open source DeepSec. This work was partly supported by the National Key Research and Development Program of China under Nos. 2016YFB0800102 and 2016YFB0800201, the NSFC program under No. 61772466, the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under No. R19F020013, the Provincial Key Research and Development Program of Zhejiang, China under Nos. 2017C01055, 2017C01064, and 2018C03052, the Alibaba-ZJU Joint Research Institute of Frontier Technologies, the CCF-NSFOCUS Research Fund under No. CCF-NSFOCUS2017011, the CCF-Venustech Research Fund under No. CCF-VenustechRP2017009, and the Fundamental Research Funds for the Central Universities under No. 2016XZZX001-04. Ting Wang is partly supported by the National Science Foundation under Grant No. 1566526 and 1718787.

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv:1604.07316*, 2016.

[4] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, 2015.

[5] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *SIGCOMM*, 2014.

[6] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball *et al.*, "Mura dataset: Towards radiologist-level abnormality detection in musculoskeletal radiographs," *arXiv:1712.06957*, 2017.

[7] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Sok: Towards the science of security and privacy in machine learning," *arXiv:1611.03814*, 2016.

[8] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv:1712.07107*, 2017.

[9] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," in *ICML*, 2018.

[10] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *S&P*, 2016.

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *EuroS&P*, 2016.

[12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*, 2017.

[13] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *AAAI*, 2018.

[14] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "cleverhans v2.1.0: an adversarial machine learning library," *arXiv:1610.00768*, 2016.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[16] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *ICLR*, 2018.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[19] Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, and J. Zhu, "Boosting adversarial attacks with momentum," *arXiv:1710.06081*, 2017.

[20] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *CVPR*, 2016.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *CVPR*, 2017.

[22] W. He, B. Li, and D. Song, "Decision boundary analysis of adversarial examples," in *ICLR*, 2018.

[23] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh, "EAD: elastic-net attacks to deep neural networks via adversarial examples," in *AAAI*, 2018.

[24] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *ICLR*, 2017.

[25] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *ICLR*, 2018.

[26] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *ICLR*, 2018.

[27] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *ICLR*, 2018.

[28] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *ICLR*, 2018.

[29] X. Cao and N. Z. Gong, "Mitigating evasion attacks to deep neural networks via region-based classification," in *ACSAC*, 2017.

[30] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, M. E. Houle, G. Schoenebeck, D. Song, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," in *ICLR*, 2018.

[31] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *NDSS*, 2018.

[32] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *CCS*, 2017.

[33] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *AsiaCCS*, 2017.

[34] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv:1707.08945*, 2017.

[35] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *AISec*, 2017.

[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, 1986.

[37] Wikipedia, "Utility," https://en.wikipedia.org/wiki/Utility.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, 2004.

[39] A. Liu, W. Lin, M. Paul, C. Deng, and F. Zhang, "Just noticeable difference for images with decomposition model for separating edge and textured regions," *IEEE Transactions on Circuits and Systems for Video Technology*, 2010.

[40] B. Luo, Y. Liu, L. Wei, and Q. Xu, "Towards imperceptible and robust adversarial example attacks against neural networks," in *AAAI*, 2018.

[41] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *AISec*, 2017.

[42] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv:1601.06759*, 2016.

[43] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *arXiv:1801.00553*, 2018.

[44] Wikipedia, "Jensen–shannon divergence," https://en.wikipedia.org/wiki/Jensen-Shannon_divergence.

[45] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[47] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Citeseer*, 2009.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[49] J. Alakuijala, R. Obryk, O. Stoliarchuk, Z. Szabadka, L. Vandevenne, and J. Wassenberg, "Guetzli: Perceptually guided jpeg encoder," *arXiv:1703.04421*, 2017.

[50] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv:1605.07277*, 2016.

[51] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv:1611.02770*, 2016.

[52] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000.

[53] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics & Data Analysis*, 2009.

[54] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, "Ensemble methods as a defense to adversarial perturbations against deep neural networks," *arXiv:1709.03423*, 2017.

[55] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defense: Ensembles of weak defenses are not strong," in *WOOT*, 2017.

[56] A. Bagnall, R. Bunescu, and G. Stewart, "Training ensembles to detect adversarial examples," *arXiv:1712.04006*, 2017.

[57] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Generative adversarial examples," *arXiv:1805.07894*, 2018.

[58] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv:1801.02610*, 2018.

[59] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models," *arXiv:1707.04131*, 2017.
[60] P. Developers, "Advbox: A toolbox to generate adversarial examples," *https://github.com/PaddlePaddle/models/tree/develop/fluid/adversarial*.
[61] IBM, "Adversarial robustness toolbox (art v0.1)," *https://github.com/IBM/adversarial-robustness-toolbox*, 2018.
[62] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, 2016.
[63] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv:1605.02688*, 2016.
[64] 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," *arXiv:1512.01274*, 2015.
[65] P. Paddle, "Parallel distributed deep learning: An easy-to-use, efficient, flexible and scalable deep learning platform," *http://www.paddlepaddle.org/*.
[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.
[67] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *CVPR*, 2017.

VIII. Model Architectures

A. The Original Model Architectures

Table X and Table XI show the originally DL model architectures (Model 1) of MNIST and CIFAR-10, respectively. For simplicity, the details of Layer 1, Layer 2 and Layer 3 layers's details are summarized in Table XII.

TABLE X
Original Model (Model 1) for MNIST

Layer Type	MNIST Architecture
Relu Conv.	32 filters (3*3)
Relu Conv.	32 filters (3*3)
Max Pooling (2*2)	
Relu Conv.	64 filters (3*3)
Relu Conv.	64 filters (3*3)
Max Pooling (2*2)	
Flatten	
Relu FC	200 units
Dropout	0.5
Relu FC	200 units
Softmax FC (10)	

TABLE XI
Original Model (Model 1) for CIFAR-10

Layer Type	CIFAR-10 Architecture (ResNet-20)
Layer 1:	filters=16, strides=1
Layer 2:	filters=16, strides=1
Layer 2:	filters=16, strides=1
Layer 2:	filters=16, strides=1
Layer 3:	filters=32, strides=2
Layer 2:	filters=32, strides=1
Layer 2:	filters=32, strides=1
Layer 3:	filters=64, strides=2
Layer 2:	filters=64, strides=1
Layer 2:	filters=64, strides=1
Average Pooling (8*8)	
Flatten	
Softmax FC (10)	

TABLE XII
Details of L1, L2 and L3 for ResNet

Layer 1:	filters, strides		
Conv2D	filters, kernel_size=3, strides, kernel_init='he_normal', kernel_regularizer=l2(1e-4)		
BN			
Activation	relu		
Layer 2:	filters, strides		
Conv2D	filters, kernel_size=3, strides, kernel_init='he_normal', kernel_regularizer=l2(1e-4)		
BN			
Activation	relu		
Conv2D	filters, kernel_size=3, strides=1, kernel_init='he_normal', kernel_regularizer=l2(1e-4)		
BN			
Activation	relu		
Layer 3:	filters, strides		
Conv2D	filters, kernel_size=3, strides, kernel_init='he_normal', kernel_regularizer=l2(1e-4)	Conv2D	filters, kernel_size=1, strides, kernel_init='he_normal', kernel_regularizer=l2(1e-4)
BN			
Activation	relu		
Conv2D	filters, kernel_size=3, strides=1, kernel_init='he_normal', kernel_regularizer=l2(1e-4)		
BN			
Activation	relu		

B. Other Models for the Transferability Case Study

For **Model 2**, we add one convolution block to the original model for MNIST and choose a similar ResNet-56 for CIFAR-10. For **Model 3**, we use AlexNet [66] and DenseNet [67] in MNIST and CIFAR-10, respectively.

IX. Parameter Settings

A. Attacks Settings

The detailed parameter settings of all the attacks are summarized in Table XIII.

B. Defense Settings

• **NAT:** The loss function of NAT is weighted with 100% normal examples and 30% AEs generated by LLC. For MNIST, ϵ is randomly chosen from a normal distribution $N(\mu = 0, \sigma = 50)$ and then clipped into interval $[0, 0.3]$. For CIFAR-10, ϵ is randomly chosen from a normal distribution $N(\mu = 0, \sigma = 15)$ and clipped into interval $[0, 0.1]$.

• **EAT:** EAT augments training data with AEs generated by R+FGSM on 4 different pre-trained models. For MINST, $\epsilon = 0.3$ and $\alpha = 0.05$ are set for R+FGSM. For CIFAR-10, $\epsilon = 0.0625$ and $\alpha = 0.03125$ are set for R+FGSM.

• **PAT:** The PAT method retrains the model with only AEs generated by the PGD attack. For MNIST: $attack\_steps = 40$, $step\_size = 0.01$ and $\epsilon = 0.3$; For CIFAR-10: $attack\_steps = 7$, $step\_size = 0.007843$, $\epsilon = 0.03137$.

• **DD:** For both MNIST and CIFAR-10, T is set to be 50.

• **IGR:** The λ regularization terms of MNIST and CIFAR-10 are set to 316 and 10, respectively.

• **EIT:** In our evaluation, we orderly employ the following four image transformation techniques.

1) Image Crop and Rescaling. For MINST, images are cropped from 28*28 to 26*26, and then rescaled back to

TABLE XIII
PARAMETER SETTING FOR ALL ADVERSARIAL ATTACKS IN EVALUATIONS

Dataset	Attacks				Configurations	
	UA/TA	Objective	Attacks			
MNIST	UA	L_∞ $\epsilon=0.3$	FGSM	$\epsilon=0.3$	$\epsilon=0.3$	
				$\epsilon=0.5$	$\epsilon=0.5$	
			R+FGSM		$\epsilon=0.15$	$\alpha=0.15$
			BIM		$\epsilon=0.3$	eps_iter=0.05
			PGD		$\epsilon=0.3$	eps_iter=0.05
			U-MI-FGSM		$\epsilon=0.3$	eps_iter=0.05
			UAP		$\epsilon=0.3$	fool rate=30%
		L_2	DF		overshoot=0.02	max_iter=50
			OM		batch_size=1000; initial_const=0.02; bin_search_steps=4	learning_rate=0.2; noise_count=20; noise_mag=0.3
	TA	L_∞ $\epsilon=0.3$	LLC		$\epsilon=0.3$	
			R+LLC		ϵ=0.15	$\alpha=0.15$
			ILLC		$\epsilon=0.3$	eps_iter=0.05
			T-MI-FGSM		$\epsilon=0.3$	eps_iter=0.05
		L_0	JSMA		$\theta=1$	$\gamma=0.1$
		L_2	BLB		None	
			CW2	$\kappa=0$	batch_size=10; learning_rate=0.02	box=-0.5, 0.5; init const=0.001
				$\kappa=20$	batch_size=10; learning_rate=0.02	box=-0.5, 0.5; init const=0.001
			EAD	EN	$\kappa=0$	batch size=10; $\beta=1e-3$
				L1	$\kappa=0$	batch size=10; $\beta=1e-3$
CIFAR-10	UA	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	$\epsilon=0.1$	
				$\epsilon=0.2$	$\epsilon=0.2$	
			R+FGSM		$\epsilon=0.05$	$\alpha=0.05$
			BIM		$\epsilon=0.1$	eps_iter=0.01
			PGD		$\epsilon=0.1$	eps_iter=0.01
			U-MI-FGSM		$\epsilon=0.1$	eps_iter=0.01
			UAP		$\epsilon=0.1$	fool rate=80%
		L2	DF		overshoot=0.02	max_iter=50
			OM		batch_size=1; bin_search_steps=4; initial_const=1	learning_rate=0.02; noise_count=20; noise_mag=8/255
	TA	L_∞ $\epsilon=0.1$	LLC		$\epsilon=0.1$	
			R+LLC		ϵ=0.05	$\alpha=0.05$
			ILLC		$\epsilon=0.1$	eps_iter=0.01
			T-MI-FGSM		$\epsilon=0.1$	eps_iter=0.01
		L_0	JSMA		$\theta=1$	$\gamma=0.1$
		L_2	BLB		None	
			CW2	$\kappa=0$	batch_size=10; learning_rate=0.02	box=-0.5, 0.5; init const=0.001
				$\kappa=20$	batch_size=10; learning_rate=0.02	box=-0.5, 0.5; init const=0.001
			EAD	EN	$\kappa=0$	batch_size=10; $\beta=1e-3$
				L1	$\kappa=0$	batch_size=10; $\beta=1e-3$

28*28; For CIFAR-10, images are cropped from 32*32 to 30*30, and then rescaled back to 32*32.

2) Total Variance Minimization. We employ a special-purpose solver based on the split Bregman method with $p=2$ and $\lambda_{TV}=0.03$.
3) JPEG Compression. We perform JPEG compression at quality level 85 (out of 100).
4) Bit-depth Reduction. We set $depth = 4$.

• **RT:** For MNIST, random resizing layer: $(28 * 28) \rightarrow (rnd{*}rnd)$ and random padding layer: $(rnd{*}rnd) \rightarrow (31{*}31)$ where rnd is a random integer between 28 and 31 from a uniform distribution; For CIFAR-10, random resizing layer: $(32 * 32) \rightarrow (rnd * rnd)$ and random padding layer: $(rnd * rnd) \rightarrow (36 * 36)$ where rnd is a random integer between 32 and 36 from a uniform distribution.

• **PD:** For MNIST, the BPD of its PixelCNN model is set to 0.8836 and the defense parameter ϵ_{defend} is 0.3; For CIFAR-10, the BPD of its PixelCNN model is set to 3.0847 and the defense parameter ϵ_{defend} is 0.0627.

• **TE:** For MNIST, the TE-based model use 16 level discretization and adversarially trained with the LS-PGA attack ($\epsilon = 0.3$, $\xi = 0.01$ and 40 steps); For CIFAR-10, the TE-based model use 16 level discretization and adversarially trained with the LS-PGA attack ($\epsilon = 0.031$, $\xi = 0.01$ and 7 steps).

• **RC:** We sample 1000 data points from the hypercube, i.e., $m = 1000$. For MNIST, the length r of hypercube is set to be 0.3; For CIFAR-10, r is set to be 0.02.

• **LID:** For MNIST, we train a logistic regression classifier where the training set consists of a positive set and a negative set. Particularly, the positive set is the set of AEs generated by FGSM with $\epsilon = 0.3$; the negative set consists of normal testing examples and their corresponding noisy examples with L_2 Gaussian noise. For CIFAR-10, we also train a logistic regression classifier with the similar training set, but the ϵ of FGSM is set to 0.1. As for both classifiers, we set $k = 20$ and $minibatch = 100$ in the LID algorithm.

• **FS:** For MNIST, we first employ two squeezers: Bit Depth (1bit) and Median Smoothing (2×2), and then train the classifier whose FPR is controlled at around 4%; For CIFAR-10, we employ three squeezers: Bit Depth (5bit), Median Smoothing (2×2) and Non-local Means (13-3-2), and then train the classifier with about 4% FPR.

• **MagNet:** For MNIST, we employ two detectors: reconstruction error-based detectors that use the L_1 and L_2 norm, and then train the classifier whose FPR is controlled at around 4%; For CIFAR-10, we employ three detectors: reconstruction L_1 error-based detector and two probability divergence-based detectors with temperature T of 10 and 40, respectively; Also, we control the FPR of classifier at around 4%.

X. SUPPLEMENTARY EVALUATION RESULTS

In this part, we provide more evaluation results of MNIST for Section IV. We provide the utility evaluation results of existing attacks in Table XIV. In Table XV, we report the classification performance of all complete defenses against existing attacks. In Table XVI, we also report the detection performance of detection-only defenses against existing attacks.

XI. SUPPLEMENTARY RESULTS OF CASE STUDIES

In this part, we provide more results of MNIST for Section V. In Table XVII, we report the transferability performance of all attacks. We also provide the classification performance of different ensembles in Table XVIII.

TABLE XIV
UTILITY EVALUATION RESULTS OF ALL ATTACKS ON MNIST

Datasets	Attack				Misclassification			Imperceptibility					Robustness			CC
	UA/TA	Objective	Attacks		MR	ACAC	ACTC	ALD_p L_0	ALD_p L_2	ALD_p L_∞	ASS	PSD	NTE	RGB	RIC	
MNIST	UAs	L_∞ $\epsilon=0.3$	FGSM	$\epsilon=0.3$	30.4%	0.762	0.065	0.398	4.887	0.300	0.554	67.589	0.643	0.908	1.000	0.0012
				$\epsilon=0.5$	44.8%	0.705	0.071	0.382	7.795	0.500	0.350	105.763	0.566	0.933	1.000	0.0012
			R+FGSM		34.2%	0.766	0.073	0.540	4.866	0.300	0.553	84.001	0.648	0.874	0.991	0.0013
			BIM		75.6%	0.995	0.004	0.740	4.527	0.300	0.572	83.466	0.990	0.979	0.894	0.0014
			PGD		82.4%	0.996	0.004	0.825	4.918	0.300	0.564	94.842	0.992	0.977	1.000	0.0034
			U-MI-FGSM		70.4%	0.989	0.008	0.658	5.343	0.300	0.534	97.801	0.979	0.972	1.000	0.0014
			UAP		30.3%	0.757	0.089	0.745	7.220	0.300	0.456	155.794	0.607	0.855	0.884	0.0000
		L_2	DF		100.0%	0.543	0.405	0.359	1.808	0.492	0.927	13.994	0.138	0.066	0.999	0.0070
			OM		100.0%	0.834	0.048	0.214	4.390	0.693	0.795	27.422	0.735	0.877	1.000	1.4738
	TAs	L_∞ $\epsilon=0.3$	LLC		5.6%	0.683	0.062	0.320	4.459	0.300	0.519	56.790	0.624	0.782	0.982	0.0001
			R+LLC		4.0%	0.651	0.090	0.534	4.843	0.300	0.437	85.311	0.547	0.763	0.974	0.0001
			ILLC		59.4%	0.865	0.012	0.757	4.510	0.300	0.572	83.904	0.769	0.786	1.000	0.0004
			T-MI-FGSM		86.4%	0.851	0.028	0.672	5.534	0.300	0.524	102.145	0.798	0.914	0.999	0.0014
		L_0	JSMA		76.4%	0.605	0.169	0.052	4.944	1.000	0.768	22.045	0.376	0.402	0.996	0.3756
		L_2	BLB		100.0%	0.677	0.200	0.373	1.778	0.536	0.910	14.145	0.460	0.026	0.981	21.6217
			CW2	$\kappa=0$	99.7%	0.326	0.318	0.342	1.746	0.528	0.925	14.086	0.003	0.004	0.388	2.6302
				$\kappa=20$	97.4%	0.995	0.001	0.393	3.570	0.842	0.772	39.359	0.995	1.000	1.000	2.5650
			EAD	EN	100%	0.361	0.322	0.200	1.717	0.620	0.933	9.928	0.025	0.036	0.766	1.3443
				L1	100%	0.371	0.312	0.189	1.746	0.657	0.934	9.558	0.039	0.038	0.836	1.3622

TABLE XV
CLASSIFICATION ACCURACY OF COMPLETELY DEFENSES AGAINST ADVERSARIAL ATTACKS ON MNIST

Datasets	Attack				# of AEs	Original Model	Defense-enhanced Models									Average
	UA/TA	Objective	Attacks				Adversarial Training			Gradient Masking		Input Transformation			RC	
							NAT	EAT	PAT	DD	IGR	EIT	RT	PD	TE	
MNIST	UAs	L_∞ $\epsilon=0.3$	FGSM	$\epsilon=0.3$	304	0.0%	88.2%	88.8%	94.1%	60.2%	76.6%	61.5%	26.0%	11.2%	93.8%	8.6%	60.9%
				$\epsilon=0.5$	448	0.0%	28.8%	25.9%	26.3%	29.5%	42.6%	27.9%	15.2%	2.9%	20.5%	1.6%	22.1%
			R+FGSM		342	0.0%	95.9%	95.6%	98.3%	78.4%	88.3%	77.8%	30.1%	17.3%	97.1%	19.3%	69.8%
			BIM		756	0.0%	93.0%	92.5%	97.9%	71.6%	83.6%	65.7%	21.2%	7.0%	97.6%	9.4%	63.9%
			PGD		824	0.0%	95.4%	93.8%	98.2%	74.5%	85.9%	67.7%	18.3%	10.6%	98.4%	11.5%	65.4%
			U-MI-FGSM		704	0.0%	90.9%	90.8%	97.3%	64.4%	80.0%	57.0%	20.2%	8.7%	97.0%	9.8%	61.6%
			UAP		303	0.0%	96.7%	95.4%	98.7%	71.3%	15.2%	12.9%	10.2%	18.2%	97.7%	42.2%	55.8%
		L_2	DF		1000	0.0%	99.6%	99.3%	99.3%	96.8%	98.9%	99.0%	68.9%	97.5%	99.3%	98.6%	95.7%
			OM		1000	0.0%	88.6%	87.7%	93.7%	70.8%	90.5%	77.9%	26.3%	4.8%	94.8%	1.0%	63.6%
	TAs	L_∞ $\epsilon=0.3$	LLC		56	0.0%	96.4%	98.2%	100.0%	73.2%	96.4%	75.0%	8.9%	23.2%	100.0%	7.1%	67.9%
			R+LLC		40	0.0%	97.5%	95.0%	97.5%	92.5%	97.5%	92.5%	25.0%	30.0%	95.0%	32.5%	75.5%
			ILLC		594	0.0%	98.7%	98.8%	99.0%	87.0%	95.1%	86.7%	25.4%	20.7%	98.5%	30.8%	74.1%
			T-MI-FGSM		864	0.0%	98.4%	97.9%	99.2%	81.9%	90.5%	72.1%	26.6%	18.8%	99.1%	22.9%	70.7%
		L_0	JSMA		764	0.0%	78.5%	74.1%	79.7%	78.5%	86.0%	73.7%	38.6%	13.7%	73.0%	35.2%	63.1%
		L_2	BLB		1000	0.0%	99.7%	99.0%	99.3%	98.7%	99.1%	99.0%	66.3%	95.2%	98.6%	98.0%	95.3%
			CW2	$\kappa=0$	997	0.0%	99.6%	99.0%	99.3%	98.2%	99.1%	98.2%	68.3%	96.4%	98.4%	98.4%	95.5%
				$\kappa=20$	963	0.0%	79.7%	78.7%	85.2%	79.2%	90.7%	73.5%	19.8%	2.9%	81.1%	0.5%	59.1%
			EAD	EN	1000	0.0%	99.3%	98.4%	99.0%	98.4%	99.1%	98.5%	64.5%	96.6%	98.0%	98.3%	95.0%
				L1	1000	0.0%	99.0%	97.8%	98.3%	98.1%	99.1%	97.9%	62.0%	94.9%	96.8%	98.3%	94.2%
	Average				682.1	0.0%	90.7%	89.8%	92.6%	79.1%	85.0%	74.4%	33.8%	35.3%	91.3%	38.1%	**71.0%**

TABLE XVI
EVALUATION RESULTS OF DETECTION-ONLY DEFENSES AGAINST ALL ADVERSARIAL ATTACKS ON MNIST

Dataset	Attack			# of Examples	Detection-only Defenses								
	UA/TA	Objective	Attacks		LID			FS			MagNet		
					TPR	FPR	AUC	TPR	FPR	AUC	TPR	FPR	AUC
MNIST	UAs	L_∞ $\epsilon=0.3$	FGSM	608	73.0%	3.6%	93.7%	96.1%	4.9%	99.1%	100.0%	6.6%	100.0%
			R+FGSM	684	70.2%	4.1%	94.5%	97.7%	3.5%	99.5%	100.0%	3.5%	100.0%
			BIM	1512	10.4%	4.2%	60.2%	92.7%	3.7%	98.7%	100.0%	3.7%	100.0%
			PGD	1648	10.3%	4.1%	54.8%	96.1%	3.4%	99.5%	100.0%	3.6%	100.0%
			U-MI-FGSM	1408	22.7%	4.1%	67.6%	90.5%	3.6%	98.4%	100.0%	3.7%	100.0%
			UAP-i	606	87.8%	4.6%	97.5%	99.7%	5.0%	99.6%	100.0%	4.0%	100.0%
		L_2	DF	2000	84.1%	2.9%	98.0%	99.9%	4.0%	99.6%	80.5%	3.6%	94.8%
			OM	2000	60.7%	3.0%	90.0%	94.0%	3.7%	99.1%	91.3%	3.7%	97.0%
	TAs	L_∞ $\epsilon=0.3$	LLC	112	87.5%	3.6%	91.1%	100.0%	7.1%	99.7%	100.0%	1.8%	100.0%
			R+LLC	80	95.0%	5.0%	85.3%	100.0%	2.5%	100.0%	100.0%	2.5%	100.0%
			ILLC	1188	64.8%	5.9%	89.2%	99.7%	3.9%	100.0%	100.0%	5.2%	100.0%
			T-MI-FGSM	1728	52.7%	3.5%	89.8%	99.3%	3.0%	99.9%	100.0%	4.5%	100.0%
		L_0	JSMA	2000	69.1%	5.6%	92.8%	100.0%	3.2%	99.6%	84.0%	5.0%	95.3%
		L_2	BLB	1528	77.5%	5.9%	94.7%	99.7%	4.8%	99.5%	98.2%	3.7%	99.1%
			CW2	1994	93.9%	3.4%	99.2%	100.0%	3.0%	99.6%	80.5%	3.7%	94.5%
			EAD (EN)	2000	92.0%	3.5%	98.5%	100.0%	3.5%	99.4%	75.8%	4.4%	92.3%
			EAD (L1)	2000	93.1%	3.8%	98.9%	100.0%	4.9%	99.5%	73.7%	3.4%	92.1%
	AVERAGE			1358.6	67.3%	**4.2%**	88.0%	98.0%	**4.0%**	99.4%	93.2%	**3.9%**	97.9%

TABLE XVII
TRANSFERABILITY RATE OF ALL ADVERSARIAL ATTACKS ON MNIST

Datasets	Attack					Original Model		Model 1		Model 2		Model 3		Average	Average
	UA/ TA	Objec- tive	Attacks		# of AEs	MR	ACAC	MR	ACAC	MR	ACAC	MR	ACAC	MR of 3 Models	ACAC of 3 Models
MNIST	UAs	L_∞ $\epsilon=0.3$	FGSM	$\epsilon=0.3$	304	100.0%	0.762	53.0%	0.692	46.4%	0.725	46.7%	0.725	48.7%	0.714
				$\epsilon=0.5$	448	100.0%	0.705	75.7%	0.584	65.2%	0.618	56.0%	0.592	65.6%	0.598
			R+FGSM		342	100.0%	0.766	38.3%	0.703	28.7%	0.750	27.8%	0.716	31.6%	0.723
			BIM		756	100.0%	0.995	66.4%	0.923	50.7%	0.899	39.3%	0.898	52.1%	0.907
			PGD		824	100.0%	0.996	62.9%	0.914	48.8%	0.868	32.5%	0.877	48.1%	0.887
			U-MI-FGSM		704	100.0%	0.989	69.6%	0.911	56.7%	0.861	46.3%	0.880	57.5%	0.884
			UAP		303	100.0%	0.757	43.6%	0.590	38.3%	0.696	16.8%	0.613	32.9%	0.633
		L_2	DF		1000	100.0%	0.543	4.7%	0.780	2.3%	0.776	6.9%	0.765	4.6%	0.774
			OM		1000	100.0%	0.834	44.8%	0.762	29.2%	0.769	32.2%	0.713	35.4%	0.748
	TAs	L_∞ $\epsilon=0.3$	LLC		56	100.0%	0.683	10.7%	0.684	7.1%	0.665	1.8%	0.889	6.5%	0.746
			R+LLC		40	100.0%	0.651	0.0%	-	0.0%	-	0.0%	-	0.0%	-
			ILLC		594	100.0%	0.865	14.5%	0.774	7.4%	0.752	5.7%	0.729	9.2%	0.751
			T-MI-FGSM		864	100.0%	0.851	27.8%	0.764	16.7%	0.745	14.5%	0.739	19.6%	0.749
		L_0	JSMA		764	100.0%	0.605	11.8%	0.824	8.1%	0.798	10.2%	0.770	10.0%	0.798
		L_2	BLB		1000	100.0%	0.677	1.2%	0.754	0.6%	0.728	1.0%	0.712	0.9%	0.731
			CW2	$\kappa=0$	997	100.0%	0.326	0.5%	0.702	0.3%	0.661	0.8%	0.745	0.5%	0.703
				$\kappa=20$	963	100.0%	0.995	57.2%	0.884	33.0%	0.874	26.9%	0.804	39.0%	0.854
			EAD	EN	1000	100.0%	0.361	1.4%	0.806	0.6%	0.785	0.7%	0.886	0.9%	0.826
				L1	1000	100.0%	0.371	1.9%	0.753	0.6%	0.842	1.0%	0.850	1.2%	0.815
	Average				682.1	100.0%	0.723	30.8%	0.767	23.2%	0.767	19.3%	0.772	24.4%	0.769

TABLE XVIII
CLASSIFICATION PERFORMANCE OF DIFFERENT ENSEMBLE METHODS ON MNIST

Dataset	Attack					Original Model	Ensemble Methods					
							Completely-random		Interclass-random		Best-defense	
	UA/TA	Objective	Attacks		# of AEs		Accuracy	Confidence	Accuracy	Confidence	Accuracy	Confidence
MNIST	UAs	L_∞ $\epsilon=0.3$	FGSM	$\epsilon=0.3$	304	0.0%	62.6%	0.606	81.5%	0.803	93.8%	0.951
				$\epsilon=0.5$	448	0.0%	21.3%	0.544	26.0%	0.651	24.8%	0.709
			R+FGSM		342	0.0%	75.7%	0.639	91.0%	0.844	97.1%	0.979
			BIM		756	0.0%	54.5%	0.672	85.7%	0.840	97.0%	0.978
			PGD		824	0.0%	56.7%	0.682	87.9%	0.847	97.7%	0.985
			U-MI-FGSM		704	0.0%	50.4%	0.670	82.4%	0.828	96.5%	0.971
			UAP		303	0.0%	76.7%	0.459	89.9%	0.764	98.7%	0.981
		L_2	DF		1000	0.0%	99.0%	0.864	99.4%	0.973	99.3%	0.994
			OM		1000	0.0%	67.8%	0.618	84.5%	0.813	94.3%	0.932
	TAs	L_∞ $\epsilon=0.3$	LLC		56	0.0%	78.0%	0.571	92.3%	0.834	100.0%	0.983
			R+LLC		40	0.0%	87.5%	0.656	95.8%	0.892	100.0%	0.974
			ILLC		594	0.0%	78.7%	0.626	95.4%	0.866	99.2%	0.989
			T-MI-FGSM		864	0.0%	75.6%	0.610	93.6%	0.849	99.2%	0.987
		L_0	JSMA		764	0.0%	71.5%	0.635	79.1%	0.801	79.6%	0.874
		L_2	BLB		1000	0.0%	98.9%	0.855	97.2%	0.956	99.3%	0.990
			CW2	$\kappa=0$	997	0.0%	99.1%	0.852	99.3%	0.972	99.5%	0.990
				$\kappa=20$	963	0.0%	57.6%	0.640	79.6%	0.814	85.7%	0.900
			EAD	EN	1000	0.0%	99.0%	0.843	99.0%	0.967	99.2%	0.982
				L1	1000	0.0%	98.9%	0.832	98.8%	0.961	98.6%	0.979
	Average				682.1	0.0%	**74.2%**	0.678	**87.3%**	0.857	**92.6%**	0.954