Security and Privacy Issues in Deep Learning

Ho Bae[†], Seoul National University, Republic of Korea
Jaehee Jang[†], Seoul National University, Republic of Korea
Dahuin Jung, Seoul National University, Republic of Korea
Hyemi Jang, Seoul National University, Republic of Korea
Heonseok Ha, Seoul National University, Republic of Korea
Sungroh Yoon[*], Seoul National University, Republic of Korea
[*]E-mail: sryoon@snu.ac.kr
[*]: To whom correspondence should be addressed, [†]: These authors contributed equally to this work.

With the development of machine learning (ML), expectations for artificial intelligence (AI) technology have been increasing daily. In particular, deep neural networks have shown outstanding performance results in many fields. Many applications are deeply involved in our daily life, such as making significant decisions in application areas based on predictions or classifications, in which a DL model could be relevant. Hence, if a DL model causes mispredictions or misclassifications due to malicious external influences, then it can cause very large difficulties in real life. Moreover, training DL models involve an enormous amount of data and the training data often include sensitive information. Therefore, DL models should not expose the privacy of such data. In this paper, we review the vulnerabilities and the developed defense methods on the security of the models and data privacy under the notion of secure and private AI (SPAI). We also discuss current challenges and open issues.

Additional Key Words and Phrases: Private AI, Secure AI, Machine Learning, Deep Learning, Homomorphic Encryption, Differential Privacy, Adversarial Example, White-box Attack, Black-box Attack

1. INTRODUCTION

Development of deep learning (DL) algorithms have transformed the solution of data-driven problems in various real-life applications, including the use of large amounts of patient data for health prediction services [Shickel et al. 2017], autonomous security audits from system logs [Buczak and Guven 2016], and unmanned car driving powered by visual object detection [Ren et al. 2015]. However, the vulnerabilities of DL systems have been recently uncovered within a vast amount of literature. Since these applications are based on a limited understanding of the security and privacy of DL systems, these applications may be unsafe.

Although many research studies have been published on both attacks and defense with DL security and privacy, they are still fragmented. Hence, we review recent attempts concerning secure and private AI (SPAI). Addressing the need for robust ar-

tificial intelligence (AI) systems for security and privacy, we develop a perspective on SPAI. Secure AI aims for AI systems with high security guarantees, and private AI aims for AI systems that preserve data privacy. Additionally, as a part of the effort to build the SPAI system, we review the fragmented findings and attempts to address the attacks and defense in DL.

Given prior knowledge of the model's parameters and structure, attacks on DL models usually attempt to subvert the learning process or induce false predictions, to inject adversarial samples. This type of attack, which can include gradient-based techniques [Biggio et al. 2013; Goodfellow et al. 2014b], is often called a white-box attack. In contrast, black-box attacks lead the target system to make false predictions, without any information about the underlying model. We observe that most of the attacks exploit the prediction confidence given by the targeted model without knowing the model's structure and parameters.

To defend from these attacks, methods such as adversarial training [Goodfellow et al. 2014b; Sun et al. 2018; Gu et al. 2018], gradient masking [Buckman et al. 2018; Dhillon et al. 2018; Song et al. 2017], GAN [Samangouei et al. 2018; Song et al. 2017] and statistical approaches [Steinhardt et al. 2017; Paudice et al. 2018b,a] have been proposed. Table III lists recent research on attacks on various DL models, their structures and parameters, and the defense against these attacks.

We review recent research on privacy and security issues associated with DL in several domains. Additionally, we taxonomize possible attacks and state-of-the-art defense methods on Secure AI and Private AI. Our work is the first attempt to taxonomize approaches to privacy in DL.

2. BACKGROUND

Behind the success of DL lies advancements in deep neural networks (DNNs) trained with an extensive amount of data. In the following, we detail the components and the training algorithm of a DNN. Furthermore, we describe recent widely used DNN architectures.

DNNs consist of layers of *artificial neurons*, or node in Fig 1, which compute the affine transformation of input and the weights followed by an activation function:

$$y = \sigma(\sum_{i=1}^{n} w_i x_i).$$ (1)

where x, y are input and output, σ is an activation function, and w is the weight. For activation functions we generally use nonlinear functions, including sigmoid ($\frac{1}{1+e^{-x}}$), tanh ($\frac{e^x-e^{-x}}{e^x-e^{-x}}$) and ReLU ($max(0,x)$). The combinations of linear and nonlinear functions enables DNNs to uncover the patterns in data.

2.1. Artificial Intelligence Powered by Deep Learning

*2.1.1. **Deep Learning Workflow—Training and Inference**.* The workflow of DL contains two phases: training and inference. DNNs learn new capabilities through the training phase from the existing data, and the learned capabilities are applied to unseen data at the inference phase. The training process of DNNs are generally done by stochastic gradient descent (SGD), as described in Fig. 1. SGD is an iterative gradient-based optimization method. In SGD, weights are updated in a way that minimizes error by using the gradients:
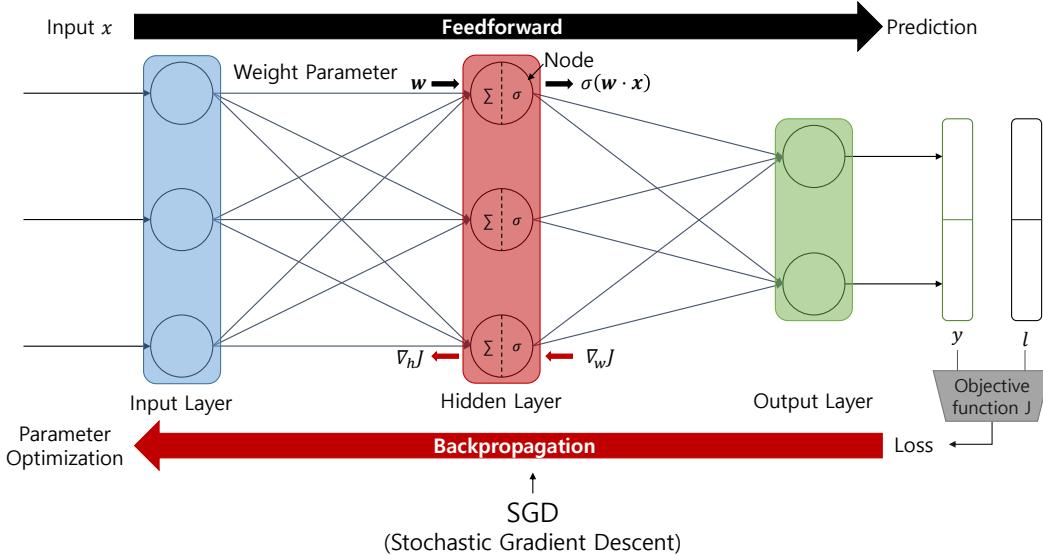
$$w \leftarrow w - \eta \nabla_w J(w)$$ (2)

Fig. 1: General DNN training process.

where a loss function $J(w)$ is used for the weight parameter w and the learning rate η. If the model converges to a desired prediction accuracy or loss, the model training is done and ready for the inference stage.

Different DNN model architectures are described in Fig. 2

— **Feed-forward neural network (FNN).** An FNN is the most basic structure of the DNNs. It contains multiple fully-connected layers, where the nodes between the layers are fully connected. Despite the simple structures, FNNs shows good performance in finding patterns from datasets such as MNIST [LeCun et al. 2010].

— **Convolutional neural network (CNN).** A general architecture for CNNs is described in Fig. 2(a). A CNN consists of one or more convolutional layers, which use convolutional operations to compute layer-wise results. This operation allows the network to learn about spatial information and hence CNNs show outstanding performance, especially for vision applications [Krizhevsky et al. 2012; He et al. 2016; Huang et al. 2017a].

— **Recurrent neural network (RNN).** A recurrent neural network (RNN) is commonly exploited for sequential data. As illustrated in Fig. 2(b), an RNN updates the current hidden unit and calculates the output by utilizing both the current input and past hidden unit. Well-known problems of RNNs, such as the gradient vanishing problem, and some variants, such as long short-term memory (LSTM) [Hochreiter and Schmidhuber 1997] and gated recurrent units [Cho et al. 2014] have been proposed to solve such problems.

— **Generative adversarial network (GAN).** A GAN framework [Goodfellow et al. 2014a] consists of a discriminator D and a generator G. G generates fake data, while D determines whether the generated data are real, as depicted in Fig. 2(c). Usually, generators and discriminators are NNs with various structures depending on the application. GANs are actively studied in various fields, such as image or speech synthesis and domain adaptation.
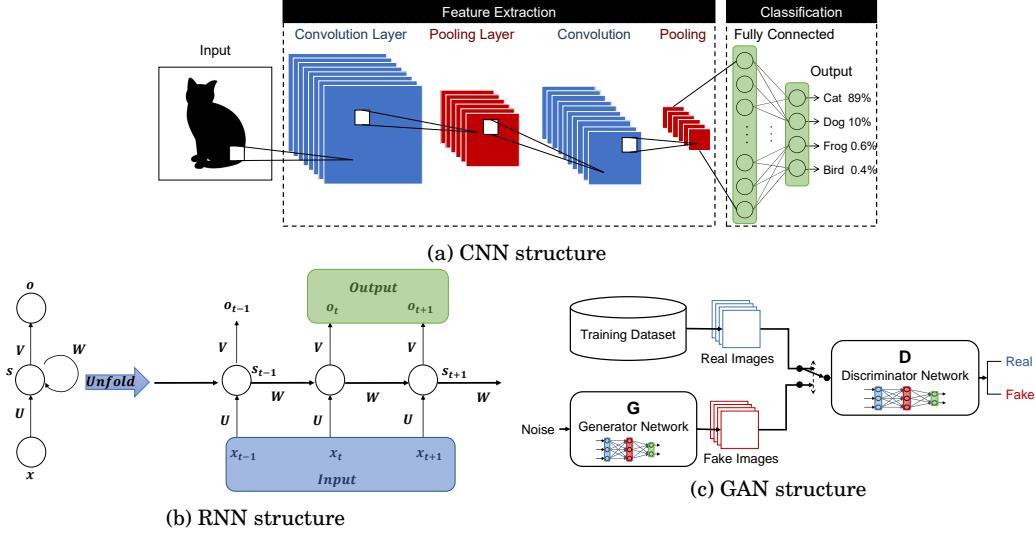
(a) CNN structure

(b) RNN structure

(c) GAN structure

Fig. 2: Different DNN model structures.

2.2. Privacy-preserving Techniques

2.2.1. ***Homomorphic Encryption****.* An encryption scheme that allows operations (addition or multiplication) on encrypted data without decrypting them or without having access to any decryption key, is called homomorphic encryption (HE). Formally, the encryption scheme Enc has the following equation:

$$\mathrm{Enc}(a) \diamond \mathrm{Enc}(b) = \mathrm{Enc}(a * b) \tag{3}$$

where $\mathrm{Enc} : \mathcal{X} \rightarrow \mathcal{Y}$ is a homomorphic ecnryption scheme with $\mathcal{X}$ a set of messages and $\mathcal{Y}$ a set of cyphertexts. a, b are messages in $\mathcal{X}$, and $*, \diamond$ are linear operations defined in $\mathcal{X}, \mathcal{Y}$, respectively.

Homomorphic cryptosystems in early stages were partial homomorphic cryptosystems [ElGamal 1985; Goldwasser and Micali 1982; Benaloh 1994; Paillier 1999], that showed either additive or multiplicative homomorphism [Gentry and Boneh 2009]. However, after Gentry and Boneh [2009] introduced ideal lattices, various attempts on FHE has been proposed [Van Dijk et al. 2010; Brakerski et al. 2014; Brakerski and Vaikuntanathan 2014; Bos et al. 2013; Hesamifard et al. 2017; Ducas and Micciancio 2015] to allow computable functions on the encrypted data. Although FHE is beneficial for many applications including cloud computing platforms and SMC, the use of massive data inputs and computational workloads as well as the nonlinearity in DL models is still a burden to be combined with DL.

2.2.2. ***Differential Privacy****.* DP is a state-of-the-art privacy preserving model [Dwork 2008]; this model guarantees with high confidence that an attacker cannot deduce any private information from databases or released models. In other words, differentially private algorithms prevent an attacker from knowing the existence of a particular record by adding noise to the query responses.

The attack scenario assumed in DP algorithms is as follows: An attacker is allowed to query two adjacent databases, which vary in (at most) one record. By sending the same query to both databases, the difference between the respective responses is considered to arise from "one record." For example, imagine that there is a database $\mathcal{D}$ of the weights and one can query only the average value of all records. In this situation,
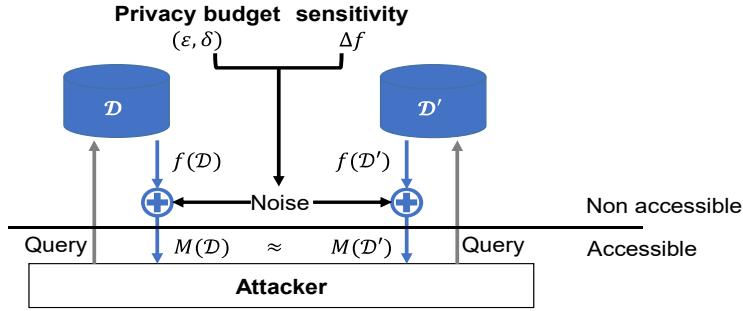
Fig. 3: Overview of the differential privacy framework.

it is impossible to grasp a specific person's weight. However, if a new record is added and the attacker knows the former average weight, then it is possible for the attacker to determine the weight of the person added.

DP algorithms counters such privacy threats by adding noise to the response as follows:

$$M(\mathcal{D}) = f(\mathcal{D}) + n \tag{4}$$

where $M : \mathcal{D} \rightarrow \mathbb{R}$ is a randomized mechanism that applies the noise n to the query response, $\mathcal{D}$ is the target database, and f is the original query response, which is deterministic.

M gives ε-DP if all adjacent $\mathcal{D}$ and $\mathcal{D}'$ satisfy the following:

$$\Pr[M(\mathcal{D}) \in S] \leq \exp(\varepsilon)\Pr[M(\mathcal{D}') \in S] \tag{5}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are two adjacent databases and $S \subseteq \mathrm{Range}(M)$ is a subset of $\mathbb{R}$. ε is the privacy budget parameter that decides the privacy level. $M(\mathcal{D})$ and $M(\mathcal{D}')$ will be more similar to the smaller ε. ε decides the privacy level compromising the data utility. Since Equation. 5 is a strict condition, (ε, δ)-DP introduces the δ term, which loosens the bounds of the error by the amount δ. In other words, δ allows M to satisfy the DP condition even if the probabilities are somewhat different. The definition of (ε, δ)-DP holds when the following equation is satisfied:

$$\Pr[M(\mathcal{D}) \in S] \leq \exp(\varepsilon)\Pr[M(\mathcal{D}') \in S] + \delta \tag{6}$$

where δ is another privacy budget that controls the privacy (confidence) levels.

Usually, the noise is sampled from the Laplace distribution or Gaussian distribution [Dwork 2008]. Each distribution depends on the sensitivity and privacy budgets. The sensitivity Δf [Dwork 2008] of the query response function f captures how much one record can affect the output and can be calculated as the maximum difference between the responses on the adjacent databases:

$$\Delta f = \max_{\mathcal{D}, \mathcal{D}'} |f(\mathcal{D}) - f(\mathcal{D}')|. \tag{7}$$

A larger sensitivity demands a larger amount of noise under the same privacy budget. There are some useful theories in which the composition of differential private mechanisms is also a differential private mechanism. The composition theorem [Dwork et al. 2006; Dwork and Lei 2009], advanced composition theorem [Kairouz et al. 2017; Dwork et al. 2010; Bun and Steinke 2016] and moment accountant [Abadi et al. 2016b] have been proposed.
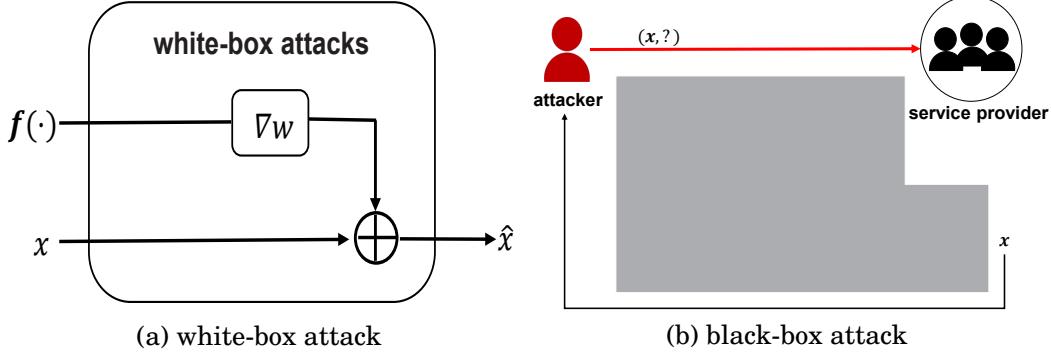
(a) white-box attack   (b) black-box attack

Fig. 4: Overview of (a) the white-box attack scenario and (b) the black-box attack scenario.

Table I: Attack methods against Secure AI

Adversarial Attack Types ↓	White-box (Figure 4a)	Black-box (Figure 4b)	Training Phase	Inference Phase
Evasion	✓	✓		✓
Poisoning	✓		✓	

3. SECURE AI

DL has been applied to various fields ranging from autonomous driving to medical diagnosis. Hence, if the DL models are exposed to hostile influences, then the training process can be destroyed or unintended behaviors from the pretrained models can be derived, which can result in terrible consequences in real life. For example, it was recently revealed that one can fool an autonomous driving system by jamming the sensors [Yan et al. 2016]. Likewise, if someone can somehow change the input of the autonomous driving model to an adversarial example, then this situation can result in the passenger's demise. Detailed examples on adversarial examples are described in Section 3.2.

Hence, we suggest the concept of *secure AI*, i.e., an AI system with security guarantees, to encourage studies on the security of AI systems. As DL is a state-of-the-art AI algorithm, we introduce and taxonomize the groups of studies regarding the attacks on DL models and defenses against those attacks.

3.1. Security Attacks on Deep Learning Models

In this section, we describe two major attacks on DL depending on which phase of the ML workflow is interfered with, namely, the poisoning attack and the evasion attack, as described in Table I. If the attack engages in the training phase and attempts to destroy the model while training, it is called the poisoning attack. For example, the example that used in this attack is referred to an adversarial training example. On the other hand, adversarial (test) examples are used in the inference phase and intentionally lead the model to misclassify the input. This attack is called the evasion attack. Evasion attacks are the group of attacks on inference phase where as poisoning attacks are group of attacks on training phase. Both type of attacks can also be further sub-grouped by white and black-box attacks. However, few researches are so far available in poisoning attacks; thus, we categorized poisoning attacks to following groups a) performance degradation attack, b) targeted poisoning attack, and c) backdoor attack.

Table II: Secure vulnerability in AI

Attack Modes		Algorithms	Reference
Evasion	**White-box** (Norm-bounded Attacks)		
		Box L-BFGS Adversary	[Szegedy et al. 2013]
		FGSM	[Goodfellow et al. 2014b]
		JSMA	[Papernot et al. 2016a]
		iFGSM	[Kurakin et al. 2018]
		CW Attack	[Carlini and Wagner 2017b]
		UAP	[Moosavi-Dezfooli et al. 2017]
		Attacks on RL	[Huang et al. 2017b]
		ATN	[Baluja and Fischer 2017]
		AS attack	[Athalye and Sutskever 2017]
		Momentum iFGSM	[Dong et al. 2017]
		BPDA	[Athalye et al. 2018]
	(Generative Model Attacks)		
		Generating natural adversarial attack	[Zhao et al. 2018]
		Constructing unrestricted attack	[Song et al. 2018]
		Semantic adversarial attack	[Joshi et al. 2019]
	Black-box		
		Hacking smart machines	[Ateniese et al. 2015]
		Adversarial examples Physical world	[Kurakin et al. 2018]
		Autoregressive model	[Alfeld et al. 2016]
		Adversarial attacks on policies	[Huang et al. 2017b]
		Malware classification	[Grosse et al. 2016]
		Practical black-box attack	[Papernot et al. 2017]
		Membership training	[Long et al. 2017]
		Physical world AE	[Kurakin et al. 2018]
		Policy induction attack	[Behzadan and Munir 2017]
		Human AE	[Elsayed et al. 2018]
Poisoning	**Performance degradation attack**	Back-gradient optimization	[Muñoz-González et al. 2017]
		Generative method	[Yang et al. 2017]
		Poisoning GAN	[Muñoz-González et al. 2019]
		Back-gradient optimization	[Muñoz-González et al. 2017]
	Targeted poisoning attack	Poisoning attack using influence function	[Koh and Liang 2017]
		Clean-label feature collision attack	[Shafahi et al. 2018]
		Convex polytope attack	[Zhu et al. 2019]
	Backdoor attack		
		Backdoor attack	[Gu et al. 2017]
		Trojaning attack	[Liu et al. 2018]
		Invisible backdoor attack	[Li et al. 2019]
		Clean-label backdoor attack	[Turner et al. 2018]

In addition to the phase of the workflow, attack scenarios on DL models can differ by the amount of information that the attacker has about the model. If the attacker has full access to all information in the model, including the model structure and the values of all parameters, then a high attack success rate is shown that cannot exist in reality. If the adversary has limited information about the model, such as the predicted label of the input or limited authority, then attacks are difficult and alternative methods are needed, such as a substitute model or data. Targeted and nontargeted attacks can be chosen based upon the attacker's goal. The attack is called a targeted attack if the adversary's objective is to alter the classifier's output to a specific target label. In a nontargeted attack, the adversary aims to make the classifier to choose an incorrect label on the adversary's input choice. Generally, a nontargeted attack shows a higher success rate than a targeted attack.

3.1.1. *Evasion Attack*.

Table III: Corresponding defense methods to secure vulnerabilities in AI

Defense Modes		Algorithms	Reference
Evasion	**Gradient Masking**		
		Distillation defense	[Papernot et al. 2016b]
		AS attack	[Athalye and Sutskever 2017]
		Ensemble defense	[Carlini and Wagner 2017a]
		Efficient defense	[Zantedeschi et al. 2017]
		Ensemble adversarial learning	[He et al. 2017]
		Randomization	[Xie et al. 2018b]
		Provable defenses	[Wong and Kolter 2018]
		Principled adversarial training	[Sinha et al. 2017]
		Input transformations adversarial learning	[Guo et al. 2018]
		Manifold defense	[Ilyas et al. 2017]
		Ensemble adversarial training	[Tramèr et al. 2019]
		Unified Embedding adversarial learning	[Na et al. 2018]
		Detecting perturbations	[Metzen et al. 2017]
		L1 based adversarial learning	[Sharma and Chen 2018]
		Pixel-defend	[Song et al. 2017]
		Defense-GAN	[Samangouei et al. 2018]
		Characterizing subspaces adversarial learning	[Ma et al. 2018]
		Stochastic activation pruning	[Dhillon et al. 2018]
		Thermometer defense	[Buckman et al. 2018]
	Adversarial Training		
		Harnessing adversarial examples	[Goodfellow et al. 2014b]
		Adversarial learning at scale	[Kurakin et al. 2016]
		Adversary A3C for RL	[Gu et al. 2018]
		Speech recognition adversarial examples	[Sun et al. 2018]
		BPDA [Athalye et al. 2018]	
	GAN		
		Pixel-defend	[Song et al. 2017]
		Defense-GAN adversarial learning	[Samangouei et al. 2018]
	Statistical Approach		
		Certified defenses	[Steinhardt et al. 2017]
		Back-gradient optimization	[Paudice et al. 2018b]
		Anomaly detection	[Paudice et al. 2018a]
	Probably Robust Model		
		An efficient SMT solver	[Katz et al. 2017]
		Convex outer adversarial polytope	[Wong and Kolter 2018]
		Certified defenses	[Raghunathan et al. 2018]
Poisoning		Certified defense	[Steinhardt et al. 2017]
		Influence functions	[Koh and Liang 2017]
		Anomaly detection	[Paudice et al. 2018b]
		Label flipping poisoning attack	[Paudice et al. 2018a]

Table IV: Potential privacy threats against private AI and the corresponding defense methods

Potential Threats by Role ↓	Homomorphic Encryption	Differential Privacy	Secure Multi-party Training
Model & Service Providers		✓	
Information Silos	✓	✓	✓
DL Service Users	✓		

Table V: A list of defending techniques of private AI in order of appearance: DL stands for deep learning, HE stands for homomorphic encryption, DP stands for differential privacy and SMC stands for secure multiparty computation

Algorithm	HE	DP	SMC	Reference
CryptoNets	✓			[Gilad-Bachrach et al. 2016]
CryptoDL	✓			[Hesamifard et al. 2017]
Privacy-preserving classification	✓			[Chabanne et al. 2017]
TAPAS	✓			[Sanyal et al. 2018]
FHE-DiNN	✓			[Bourse et al. 2018]
DP-SGD		✓		[Abadi et al. 2016b]
DP LSTM		✓		[McMahan et al. 2018]
DPGAN		✓		[Xie et al. 2018a]
DPGM		✓		[Acs et al. 2018]
DP Model Publishing		✓		[Yu et al. 2019]
Privacy-preserving logistic regression		✓		[Chaudhuri and Monteleoni 2009]
dPA		✓		[Phan et al. 2016]
dCDBN		✓		[Phan et al. 2017a]
AdLM		✓		[Phan et al. 2017b]
PATE		✓		[Papernot et al. 2017]
Scalable private learning		✓		[Papernot et al. 2018]
Generating DP datasets		✓		[Triastcyn and Faltings 2018]
DSSGD			✓	[Shokri and Shmatikov 2015]
Privacy-preserving DL via additively HE	✓		✓	[Aono et al. 2018]
SecureML	✓		✓	[Mohassel and Zhang 2017]
MiniONN	✓		✓	[Liu et al. 2017]
DeepSecure			✓	[Rouhani et al. 2018]
Gazelle			✓	[Juvekar et al. 2018]

White-box Attack. The initial study on evasion attacks started from [Szegedy et al. 2013]. These researchers suggested the idea of using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm to generate an adversarial example. These authors propose a targeted attack method, which involves solving the simple box-constrained optimization problem of

$$\text{minimize } \|n\|_2$$
$$\text{s.t.} \quad f(x+n) = \tilde{l}, \tag{8}$$

where $x \in \mathbb{R}^{I^h \times J^w \times K^c}$ is the untainted image ($I^h \times J^w \times K^c$ represents the height, width and channel of the image), and $\tilde{l} \in \{1, \cdots, k^c\}$ is the target label; and n represents the minimum amount of noise needed to disassociate the image from its true label. Box L-BFGS adversary was developed to search the minimum perturbation needed for a successful attack. Sometimes this method creates an inapplicable adversarial perturbation n, which performs only the role of image blurring. This form of attack has a high misclassification rate but also a high computational cost because the adversarial

$$x \qquad\qquad \mathbf{sign}\left(\bigtriangledown_x J(w, x, \widetilde{l})\right)$$

Lesser Panda: 99.99%   **+ .02**   **=**   **Pole Cat: 86.54%**

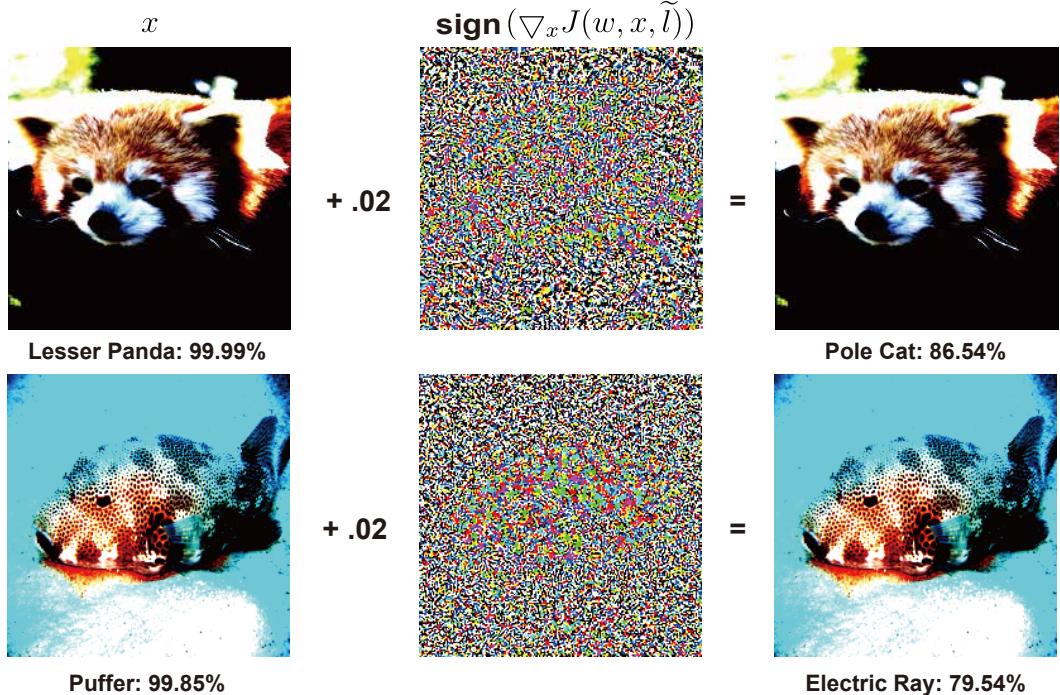**Puffer: 99.85%**   **+ .02**   **=**   **Electric Ray: 79.54%**

Fig. 5: Two adversarial examples generated by the fast gradient sign method Goodfellow et al. [2014b]. Left column: the original images. Middle column: the generated adversarial perturbations. Right column: the adversarial images into which the adversarial perturbation is added.

examples are generated to the results solving the optimization problem in Equation 8 via a box-constrained L-BFGS.

On the other hand, Carlini's and Wagner's attack (CW attack) [Carlini and Wagner 2017b] is based on an L-BFGS attack [Szegedy et al. 2013], and modifies the optimization problem in Equation 8 as

$$\text{minimize } D\left(\tilde{x}, x\right) + c \cdot g\left(\tilde{x}\right) \qquad (9)$$

where D is a distance metric that includes L_p, L_0, L_2, and L_∞; $g(\tilde{x})$ is an objective function, in which $f(\tilde{x}) = \tilde{l}$ if and only if $g(\tilde{x}) \leq 0$; and $c > 0$ is a properly chosen constant. Existing optimization algorithms can be solved by this modification of Equation 18. The use of the Adam [Kingma and Ba 2015] optimizer enhances the effectiveness by quickly finding adversarial examples. For relaxation, these authors used the method of the change in variables or projection into box constraints for each optimization step.

Papernot et al. [2016a] introduced a targeted attack method that optimizes under the L_0 distance, which is known as the Jacobian-based saliency map attack (JSMA). This method constructs a saliency map based on the gradient derived from the feedforward propagation and modifies the input features that maximize the saliency map in a way that increases the probability to be classified as target label $\tilde{l}$.

In general, a DL model is described as nonlinear and overfitting, but in [Goodfellow et al. 2014b], the fast gradient sign method (FGSM) was introduced. Goodfellow et al. [2014b] asserted that the main vulnerability of NN to an adversarial perturbation is caused by their linear nature. This method linearizes the cost function around the

present value, and finds its maximum value from the closed-form equation as follows:

$$\tilde{x} = x + \upsilon \cdot \text{sign}(\nabla_x J(w, x, \tilde{l})) \tag{10}$$

where $\tilde{x}$ is the adversarial example; x is the untainted input, and $\tilde{l}$ is the target label. The parameter υ decides how strong the adversarial perturbation applied to the image is, and J is the loss function for training. Although the proposed method can generate adversarial examples with relatively low computational costs, it shows a low success rate.

To overcome the shortcomings of the previous two ideas, various compromises have been made, and the iterative FGSM [Kurakin et al. 2018] is one of them. The iterative FGSM calls the FGSM multiple times with a small step size for each update. The clip function implements a per-pixel clipping of the image. Technically, the output will be in the L_∞ ε-neighborhood of the original image. The detailed update rule is described as follows:

$$\tilde{x}_0 = x, \tilde{x}_{N+1} = \text{Clip}_{x,\upsilon} \left\{ \tilde{x}_N + \upsilon \cdot \text{sign}(\nabla_x J(w, \tilde{x}_N, \tilde{l})) \right\} \tag{11}$$

where $\tilde{x}$ is the adversarial example iteratively optimized and $\tilde{x}_N$ is the intermediate result in the N-th iteration. As a result, this method showed improved performance in terms of the generation throughput and the success rate.

Using the iterative method proposed above, Dong et al. [2017] made use of a momentum term as an additional method of increasing the transferability of the input that generated by an adversarial, as described in Fig. 5. This method was presented in the Adversarial Attacks and Defences Competition [Dong et al. 2017] at NIPS 2017 and won the first place in the tracks of nontargeted and targeted attack. The main idea of Dong et al. [2017]'s paper is as follows:

$$g_{N+1} = \mu \cdot g_N + \frac{\nabla_x J \left(f\left(\tilde{x} \right), \tilde{l} \right)}{\left\| \nabla_x J \left(f\left(\tilde{x} \right), \tilde{l} \right) \right\|_1}, \ x_{N+1} = \text{Clip}_{x,\upsilon} \left\{ \tilde{x} + \upsilon \cdot \text{sign}(g_{N+1}) \right\}. \tag{12}$$

Compared to Equation 11, adding the g_N decay provides the momentum with a gradient.

An adversarial transformation network (ATN) [Baluja and Fischer 2017] is another targeted attack method. An ATN is a NN trained to generate targeted adversarial examples with a minimal modification of the original input, making it difficult to differentiate from the clean examples.

Beyond adding different noise values per input for misclassification, universal adversarial perturbations [Moosavi-Dezfooli et al. 2017] show the presence of universal (image-agnostic) perturbation vectors that cause all natural images in a dataset to be misclassified at a high probability. The main focus of the paper is to find a perturbation vector $n \in \mathbb{R}^{I^h \times J^w \times K^c}$ that tricks the samples in the dataset. Here, μ represents the dataset that contains all of the samples.

$$\hat{k} \left(x + n \right) \neq \hat{k} \left(x \right), \text{for most } x \sim \mu. \tag{13}$$

The noise n should satisfy the following conditions of $\| n \|_\iota \leq \xi$, and the conditions of:

$$\mathop{\mathbf{P}}_{x \sim \mu} \left(\hat{k} \left(x + n \right) \neq \hat{k} \left(x \right) \right) \geq 1 - \delta, \tag{14}$$

where $\hat{k}$ is the classifier; ξ restricts the value of the perturbation, and δ quantifies the specified fooling rate for all images.

In most adversarial attacks, the efficacy of each attack can be decreased via transformations, such as viewpoint shift and camera noise. There is a very low percentage of cases in which an image to which an adversarial noise is added directly applies to the classifier in the system of a physical world. These methods usually have some preprocessing steps and angle changing adjustments. Athalye and Sutskever [2017] proposed a method to overcome this current limitation by generating a perturbation that enables the input have a variety of distortions such as random a rotation or a translation, and the addition of noise is implemented to be misclassified in a classifier. In addition, these investigators used a visual difference for a boundary radius ball constraint instead of a distance in texture space.

A backward pass differential approach attack method [Athalye et al. 2018] was recently proposed, which is capable of preventing recent gradient masking defense methods. The authors claim that finding defenses that rely on recently suggested gradient masking methods can be circumvented by performing the backward pass with the identity function, which is for approximating true gradients.

Adversarial examples are typically designed to perturb an existing data point within a small matrix norm at the pixel level, whereas current defense methods analyze and expect this type of behavior. To defeat these defense methods, recently, the adversary semantically alter attributes of the input image, instead of pixel level approach.

Zhao et al. [2018] proposed an approach to generate natural adversarial examples. A characteristic of this approach is that the generated examples appear natural to humans. They suggested using the latent space z of the GAN structure to search required perturbation that can induce misclassification. To find z^* that satisfies a goal and a constraint written below, they additionally trained a matching inverter MI to search for z^*,

$$z^* = \underset{\widetilde{z}}{\operatorname{argmin}} \|\widetilde{z} - MI(x)\| \text{ s.t. } f(G(\widetilde{z})) \neq f(x), \tag{15}$$

where f is a classifier and G is a generator. The suggested method is evaluated on image and text domains. This paper presented sementically close to input adversarial examples may aid on the robustness of black-box classifiers. The proposed method by Song et al. [2018] is similar to Zhao et al. [2018]. The difference relies on the use of ACGAN, and the utilization of the classifier of ACGAN to misclassify the generated adversarial at the target network and yet still looks like the original class. Furthermore, with an idea from norm-bounded attack methods, this paper added noise to the generated images to boost its attack ability.

Unlike the aforementioned semantic adversarial attacks, Joshi et al. [2019] suggested generating the adversarial examples that are perceptibly altered, while maintaining natural-looking appearance. To achieve this, the paper utilized parametric generative transformations using Fader Networks and Attribute GANs. Based on the multi-attribute transformation models, they created the adversarial examples in which only a specific attribute is changed and this natural-looking change triggers a targeted misclassification. It uses a different strategy than other adversarial attacks, and the change of certain attributes is very noticeable.

Black-box Attack. In the real world, accessing models or data sets that are used for training the model, or both are too difficult. Although there is a tremendous amount of public data (image, sound, video, etc.), the internal data used for training models from industry are still secret. Moreover, models contained in mobile devices are not accessible to attackers. The black-box attack assumes a situation similar to the reality. The attacker has no information about the model and the dataset. The available infor-
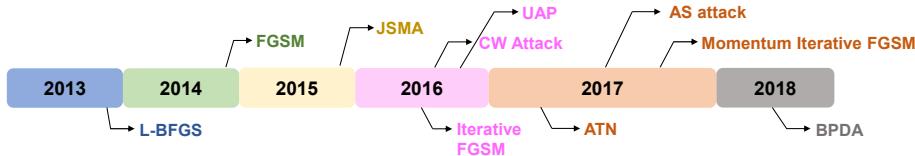
Fig. 6: Historical timeline of white-box attacks. (Abbreviations: L-BFGS [Szegedy et al. 2013] = Limited memory broyden fletcher goldfarb shanno, FGSM [Goodfellow et al. 2014b] = Fast gradient sign method, JSMA [Papernot et al. 2016a] = Jacobian based saliency map attack, CW attack [Carlini and Wagner 2017b] = Carlini's and Wagner's attack, UAP [Moosavi-Dezfooli et al. 2017] = Universal adversarial perturbation, ATN [Baluja and Fischer 2017] = Adversarial transformation networks, AS attack [Athalye and Sutskever 2017] = Athalye's and Sutskever's attack, and BPDA [Athalye et al. 2018] = Backward Pass differentiable approximation.) Red: CW attack is an advanced idea of L-BFGS. Purple: Like mentioned on each title, FGSM is a basic idea of iterative and momentum iterative FGSMs. Green: UAP and AS attack methods created the idea of generating a special perturbation that is robust to either image preprocessing or resource limitation. Pink: BPDA defeated a recently proposed large number of gradient masking defenses

mation is the input format and the output label of a target model when using a mobile application. A target model can be the models hosted by Amazon and Google.

At first, attackers tried to generate attacks as white-box methods that use the gradient of the target model directly. However, in black-box settings, the gradients of the target model are inaccessible for the attackers. Thus, attackers devised a model to replace the target model, which is often referred as a substitute model. Such attacks made with the substitute models are also called transfer attacks. According to Szegedy et al. [2013] and Goodfellow et al. [2014b], NNs can attack other models with no assumption of the number of layers and the number of hidden nodes as long as the target task is the same. These authors considered this finding to be due to NNs's linear nature, in contrast to previous studies in which the transferability is due to the nonlinearity of the NN. The activation function of the sigmoid of ReLU is well known to produce a nonlinearity. The sigmoid has the advantage of nonlinearity, but it is tricky to use in learning. On the other hand, ReLU is widely used because it is easy to learn with, but nonlinearity does not grow, as in the case of a sigmoid. Thus, the replication of the target models can learn a similar decision boundary because the target task is the same.

Moreover, Papernot et al. [2016] showed that transfer is possible between traditional ML techniques and NN using both the experiment's intratechnique transferability between the same algorithms but different initializations and the cross-technique transferability between different algorithms such as support vector machine (SVM) and NNs. For example, Kurakin et al. [2018] assumed the case in which a model obtained input by a camera or sensors that is not directly obtained, and as a result, the model attacks a NN using the pictures of the attacks. The attack still occurred, and thus, this approach showed robustness to the transformation.

As described above, the transfer attack is possible by creating a substitute model for a target. In the process of the substitution of a target model, it is possible to use an approximate architecture, such as the CNN, RNN, and MLP by exploiting the input format (image or sequence). The model can be trained by collecting data similar to the data obtained by learning the target from the public. However, the cost of the collection

is enormous. Papernot et al. [2017] solved this issue using an initial synthetic dataset and Jacobian-based data augmentation method. If the dataset of the target is from the modified National Institute of Standards and Technology (MNIST) database, then the initial synthetic dataset can be handcrafted digital digit images of approximately 100 a subset of a test set that is not used when training the targets. The label can be obtained by using the data as an input to the oracle, which is the target model. After training the substitute using the input and the labeled pair, the authors crafted an adversarial example using Goodfellow et al. [2014b] and Papernot et al. [2016b]. The results of the experiment on the transferability of the MNIST case showed an approximately 90% success rate that corresponded to an epsilon range of 0.5–0.9. However, if an attacker wants to label its inputs by blowing queries to a service such as Google or Amazon, then the attacker has a limited number of queries or a high probability of being caught by the detector due to the large number of instances. To resolve this problem, Papernot et al. [2016] introduced reservoir sampling and reduced the amount of data needed to train the substitute.

When the attacks use improved strategies, the defenses are also improved. Because of the development of the defense [Tramèr et al. 2019], transfer attacks that use a substitute model are successfully blocked. Furthermore, these attacks require many queries to perform a single attack with the cost of training a substitute model, and targeted attacks are not successfully transferable [Chen et al. 2017b; Narodytska and Kasiviswanathan 2017; Ilyas et al. 2019]. As a result, Chen et al. [2017b] introduced a target model gradient approximation method, based only on the output of the target network. In [Chen et al. 2017b], black-box attacks are conducted using the output of the target network without the gradient of the target model or a substitute. The authors suggest zeroth-order optimization to estimate the gradient of the target model. Zeroth order optimization uses the loss difference between the output of the original image, and an image changed only one pixel in the original image. This method is successfully applied to a target network without the gradient, however required many queries to obtain such gradients with as many the number of pixels. Therefore, they adopt attack-space dimension reduction, hierarchical attacks, and importance sampling to effectively reduce the number of queries.

Boundary and decision-based attack that uses only a predicted label or description from the target model are suggested in [Brendel et al. 2018]. The decision-based attacks consider a realistic setting in which more constrained information is available and are more robust to gradient masking. For the boundary-attack, three approaches are used: a) an initial sample in the adversarial region is selected; b) the sample random-walks toward the decision boundary between adversarial and non-adversarial region by reducing the distance from a target example, and c) stay in the adversarial region through rejection sampling. In [Ilyas et al. 2018], more specific information is assumed to be available, namely, query-limited settings, partial-information settings, and label-only settings. Natural evolutionary strategies (NES) [Wierstra et al. 2008] is used to generate adversarial examples in a query-limited setting. In the partial-information scenario, an instance of the target class is selected as an initial sample and repeatedly projected to the L_∞-boxes that maximize the probability of the adversarial target class.

Black-box attacks are conducted and; the performances of attacks appeared to have reached a maximum saturation value. In Ilyas et al. [2019], use of prior knowledge of the data or the gradient of the model is established. For example, an image has a local correlation between closer pixels; thus, it can be inferred that the gradients of the closer pixels are similar. In addition, correlations exist between successive gradients in time. A latent vector that has prior information is used to estimate the gradient and is updated using *reduction from bandit information* [Hazan et al. 2016]. The attacks

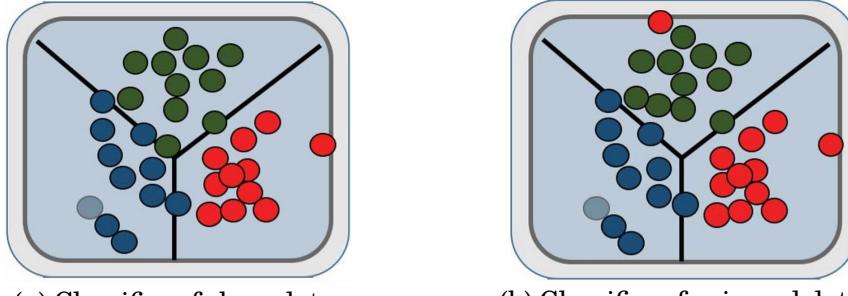(a) Classifier of clean data · (b) Classifier of poisoned data

Fig. 7: The functionality of poisoning a sample. (a) The decision boundary after training with normal data, (b) The decision boundary after injecting a poisoning sample.

using prior information are successfully applied, and the queries that are required to generate adversarial examples are effectively reduced than by the method using NES [Ilyas et al. 2018].

*3.1.2. **Poisoning Attack**.* If the evasion attack is to avoid the decision boundary of the classifier at the test time, then a poisoning attack intentionally inserts a malicious example into the training set at the training time in such a way as to interfere with the learning of the model or to more easily attack at the test time. There is a large number of poisoning attack methods that can be successfully applied to traditional ML such as SVM or least absolute shrinkage and selection operator (LASSO), but there are only a few methods for NNs. The tradition poisoning attack can be expressed mathematically, but NNs have been difficult to poison because of their complexity. Poisoning attacks are categorized into three types in accordance to the attacker's goal: performance degradation attacks, targeted poisoning attacks, and backdoor attacks. The goal of adversaries could be to compromise the learning process of the system, to provoke target sample misclassification via feature collision with other poisoned samples or to render the backdoor to be recognized as a man of power with authentication when deployed.

Performance degradation attacks aim to subvert the training process by injecting malicious samples and induce performance degradation of test accuracy by solving a bi-level optimization problem. Generally, the worst-case situation is assumed to account for the safety of the target model, but the information and accessibility which attackers have get can be different. Muñoz-González et al. [2017] presented two attack scenarios of performance degradation attacks, which are perfect-knowledge (PK) attacks and limited-knowledge (LK) attacks. As the terms suggest, a PK attack scenario is an unrealistic setting, and hence, it is only assumed for a worst-case evaluation of the attack. On the other hand, in LK attack scenarios, the typical knowledge that the attacker possesses, is described as $\theta = (\hat{\mathcal{D}}, \mathcal{X}, \mathcal{M}, \hat{w})$, where $\mathcal{X}$ is the feature representation and $\mathcal{M}$ is the learning algorithm. The hat symbol denotes that a component has the limited knowledge; $\hat{\mathcal{D}}$ is the surrogate data, and $\hat{w}$ is the learned parameter from $\hat{\mathcal{D}}$.

$$\mathcal{D}_c^* \in \underset{\mathcal{D}_c' \in \phi(\mathcal{D}_c)}{\arg\max} \quad \mathcal{A}(D_c', \theta) = J(\hat{\mathcal{D}}_{\text{val}}, \hat{w})$$

$$\textbf{s.t.} \qquad \hat{w} \in \underset{w' \in W}{\arg\min} J(\hat{\mathcal{D}}_{\text{tr}} \cup \mathcal{D}_c', w') \tag{16}$$

where the $\hat{\mathcal{D}}$ is divided into the training data $\hat{\mathcal{D}}_{\mathrm{tr}}$ and validation data $\hat{\mathcal{D}}_{val}$. $A(\mathcal{D}'_c, \theta)$ is an objective function that evaluates the impact of the adversarial examples on the clean examples. This function can be defined as a loss function and $J(\hat{\mathcal{D}}_{\mathrm{val}})$, which measures the performance of the surrogated model using $\hat{\mathcal{D}}_{\mathrm{val}}$. The optimization problem comprises bi-level optimization, and the influence of $\mathcal{D}_c$ is propagated using $\hat{w}$. The primary objective of the optimization is to ruin the model, and the label of the poison is generic. If a specific target is required, Equation 16 is changed to

$$\mathcal{A}(\mathcal{D}'_c, \theta) = -J(\hat{\mathcal{D}}'_{\mathrm{val}}, \hat{w}) \qquad (17)$$

where $\hat{\mathcal{D}}'_{\mathrm{val}}$ is the manipulated validation set, which is similar to $\hat{\mathcal{D}}$ but with misclassified labels for the desired output. Muñoz-González et al. [2017] proposed the back-gradient optimization to solve Equations 16 or 17 and generated poisoning examples and compared them with previous gradient-based optimization methods. Because gradient-based optimization requires a strict convexity assumption of the objective function and Hessian-vector product, Muñoz-González et al. [2017] argued that such an approach is not applicable to complex learning algorithms, including NNs and DL architectures. In addition, Yang et al. [2017] introduced the possibility of applying the gradient-based method to DNNs, and they developed a generative method inspired by the concept of GAN [Goodfellow et al. 2014a]. Rather than computing the gradients directly, Yang et al. [2017] used an autoencoder as a generator to speed up computation time more than 200x compared to the gradient-based method.

Thus far, the proposed attacks are easily detected by outlier detection. Recently, Muñoz-González et al. [2019] proposed the poisoning attack generation method using GANs. They generate attacks that are undetected by human eyes using the property of a GAN. The model, named pGAN, has three components; generator, discriminator, and target classifier. Using the min-max game between the generator and discriminator, pGAN generates more realistic images with poisoning ability. Using a hyperparameter that adjust the influence of realistic image generation process and poisoning ability, they manipulate the trade-off between effective attack and detectability. When the influence of the realistic image generation process is more higher, the attack success rate is low. Conversely, when the influence of poisoning ability is higher, the generator tends to produce outliers; thus attacks are more detectable.

Targeted poisoning attacks cause the target test points to be misclassified at inference time due to the effects of the poisoning attacks and is firstly introduced by Koh and Liang [2017]. The operation of deep neural networks is similar to a black-box, making it difficult to know the origin of the prediction results from a model. The authors tried to explain the prediction results using training data since the model parameters are trained from training data. Because retraining a model after removing a point or modifying a point is too expensive, they formulate the influence of up-weighting a training point or modifying a training point at training time to parameters and loss. The poisoning attacks are generated iteratively using the influence of a modifying training point to a test point.

Shafahi et al. [2018] defined clean-label attacks and adopted feature collision method to achieve targeted poisoning attacks. Even for a small number of samples placed in the training data, the attack may be unsuccessful if data are pre-processed by the experts or directly published in the internet and labeled by the experts. To handle this situation, the labels of attacks are set to be matched to the the image viewed by a human, and feature conflict method is used. The attacker selects the target image t and the base image b from the test set and expects the target image to be misclassified by the label of the base set. The attack p is initialized with the base image and

created using the equation below.

$$\mathbf{p} = argmin_x ||f(\mathbf{x}) - f(\mathbf{t})||_2^2 + \beta ||\mathbf{x} - \mathbf{b}||_2^2 \tag{18}$$

By making the p similar to the base image in the image space and p close to t in the feature space mapped by function f, the attack surrounds the target feature, changes the decision boundary and makes the target image to be classified as a base class.

For example, when there is an image of a dog and a bird, the method uses a gradient of the model to change the dog to have a bird-like feature representation. At this time, the picture of a dog is called a base image, and the picture of a bird is called a target image. The goal is to change the decision boundary by adding a perturbed base image to the training data using a gradient. As a result, the target image is misclassified as a class of base image and the target can be used as a key to exploit the model at will of the attacker.

The authors analyzed the attacks in two retraining situations: end-to-end learning which fine-tunes the entire model and transfer learning which fine-tunes only the final layer. This method that uses a base and a target image (called a one-shot kill attack) has been successfully applied to transfer learning which dramatically changes the decision boundary; however, it has not been applied to end-to-end learning, which changes the lower layer by extracting fundamental features. From this, Shafahi et al. [2018] succeeded in a poisoning attack by proposing a watermarking method of projecting a target image to a base image by adjusting the opacity and also multiple poison instance attacks using several target and base images to efficiently create an attack sample.

Because neural networks do not have the same feature mapping function, the feature collision using a model is not sufficient to apply to a black-box model. Zhu et al. [2019] proposed a feature collision attack (FC attack) for an ensemble feature mapping function and a convex polytope attack (CP attack). FC attacks have increased constraints as the number of functional mapping models increases. Therefore, FC attacks are not successful, and the target form is revealed in the attack. CP attack using convex property transform the target into or near polytope and is well-transferred. Because the victim model learns new feature mapping functions by end-to-end training, the attacker has difficulty to poison the victim model. Thus, multilayer convex polytope attack that generates poisoning attacks using every activation layer is suggested.

Backdoor attacks aim to install a backdoor to be used in with test time manipulation. The attacks have been first introduced by Gu et al. [2017]. The authors used specific stickers or markers on the image to cause false classifications, such as recognizing a stop sign as a speed limit. Currently, it is prevalent to download the pre-trained weights from ImageNet dataset to train the own task of the user or to use machine learning as a service (MLaaS) to outsource training of a model. Using this fact, Liu et al. [2018] generated trojaning attacks. Trojaning attacks assume the worst-case scenario that attackers can change parameters and training data directly but cannot access the validation set of users and cannot use the training data to make attacks. Trojaning attacks insert a specific tag, or a trigger, like a watermark, and the image attached to the trigger is classified to the targeted class, defined by the attacker at inference time. The attacks are divided into four steps: 1) trigger and the target label are selected; 2) the attacker selects the node in the target layer with the highest connectivity with the preceding layer, and the trigger is updated from the gradient derived from the difference the activation result of the selected node and the targeted value of the node; 3) using the mean image of public dataset, train data is reverse-engineered to be classified as the specific class; 4) the target model is trained using the reverse-engineered image dataset and reverse-engineered image dataset tagged trigger. After the retrained model using the process is deployed, the attacker can get the targeted

prediction of input by attaching the trigger. Trojaning attack is successfully applied to face recognition, speech recognition, auto-driving and age recognition application.

Chen et al. [2017a] introduced three constraints: 1) no knowledge of the model, 2) injecting a partial piece of the training data, and 3) poisoning data not detected by humans. This model is based on real-world situations. These authors proposed two methods, namely, input-instance-key strategies and pattern key strategies, for weaker adversaries to break the security and obtain the privilege of the face recognition system. The former strategy renders an image be a key image and recognizes it as a targeted label. In consideration of the situation of using a camera, several random noise components are added to the sample. In the latter case, three strategies exist: 1) blended injection strategy, 2) accessory injection strategy, and 3) blended accessory injection strategy. The first is to blend a kitty image or a random pattern onto the input image. However, it is uncommon to add an unrealistic pattern to the image captured by the camera, in real situations, and thus, the second strategy applies an accessory such as glasses or sunglasses to the input. This approach is easy to use at the inference stage. When training, the parts other than the glasses have the same value as the input image, and the pixel values of the glasses are applied only to the glasses. The last method combines the first and the second strategy. Unlike previous studies in which poisoning data accounted for 20% of the training data, only five poisoned samples were added when 600,000 training images were used, for instance, the key strategy, and approximately 50 poisoned samples were used for the pattern key strategies. A successful backdoor was created by adding a small fraction of poisoned samples.

Recently, methods that produce attacks invisible to humans have been developed. Li et al. [2019] proposed an invisible backdoor attack method that is similar to a trojaning attack with distributed trigger, which is invisible. Turner et al. [2018] proposed a clean-label backdoor attack based on GAN and adversarial examples bounded in L_p-norm. Its attack is achieved by interpolating the latent vectors of the target and base samples. The adversarial-example-based attack is made from the adversarial example that are misclassified by the victim model but correctly-labeled by a human. The two types of attacks are trained with attached a trigger and imprint the backdoor pattern as a feature by disturbing the model.

3.2. Adversarial Examples in a Real World Setting

A recent targeted attack [Ilyas et al. 2018] was performed on the Google Cloud Vision (GCV) API compromising commercial systems. This approach became problematic to the second service provider because of the use of a decision-making service with a given prediction score. For example, a service provider that uses the GCV API for an auto-driving application will fail to stop at an adversarily crafted stop sign [Elsayed et al. 2018]. Finlayson et al. [2018] extended adversarial attacks to medical imaging tasks. Sharif et al. [2016] demonstrated adversarial attacks on facial biometric systems to impersonate physical appearance. To impersonate a target, an input x is perturbed via an additional noise to maximize the probability of the target class. To find a global minimum, the author uses gradient descent. Athalye and Sutskever [2017] also showed the physical world on generating 3D objects introducing expectation of transformation (EOT) algorithm. The key insight behind EOT is to model such perturbation within the optimization procedure. EOT uses a chosen distribution of transformation functions rather than optimizing the log-likelihood of a single sample. Both white-box and black-box attacks were presented fooling medical DL classifiers. As such, Finlayson et al. [2018] demonstrated that potential harm in the medical domain by such adversarial attacks.

In addition to the medical domain, Carlini and Wagner [2018] proposed the first adversarial examples of automatic speech recognition. These authors applied a white-

box iterative optimization-based attack and showed a 100% success rate, demonstrating that the feasibility of adversarial attacks on an image can be transferred to another domain. Additionally, these researchers reconstructed the waveform of input x to $x + v$ while exploiting conventional measure distortion and successfully produced speech with the desired phrase with a 99.9% similarity given any audio waveform.

3.3. Defense Techniques Against Deep Learning Models

There are a variety types of defense techniques against DL models. Defense techniques can be categorized into two large groups: evasion and poisoning. Defense techniques against evasion attacks can be further categorized into two groups, namely, nonobfuscated gradient masking (which includes adversarial training), and obfuscated gradient masking. Defense techniques are not only used to limited to attacks but also are utilized to improve prediction results. For example, Kurakin et al. [2016] suggested that adversarial training can be employed in the two scenarios: a) when a model is overfitting, and b) when security against adversarial examples is a concern. For example, in a recent study on speech data [Sun et al. 2018] trained the DNN was trained using adversarial examples along with the clean examples, to increase the robustness against evasion attacks.

*3.3.1. **Defense Techniques against Evasion Attacks***. The basic idea of gradient masking is to have a method that augments the adversarial examples, which is created by both making the gradients point slightly farther than a decision boundary with clean examples and a method that makes use of techniques that hand over incorrect or foggy gradients to an adversary. Both of these methods are currently under the category of gradient masking.

Nonobfuscated Gradient Masking Currently, most representative studies of non-obfuscated gradient masking involve adversarial training as mentioned in [Goodfellow et al. 2014b]. An adversarial example was first proposed by [Szegedy et al. 2013] raising the issue of image misclassification by applying an imperceptible perturbation. They adopted traditional work that attempted to analyze neural networks that were applied to computer vision problems. Their intuition is to find $x+r$ that is closest image to x classified as l by the function f. Since the exact computation $D(x, l)$ is a hard problem, they approximated it by using a box-constrained L-BFGS function. By including such an adversarial example, [Goodfellow et al. 2014b] proposed adversarial training against an efficient FGSM adversary. The adversarial training procedure can be interpreted as minimizing the worst-case error with the perturbed data of an adversary. Adversarial training can also be seen as learning to play an adversarial game with a model that requests labels on new points. In [Goodfellow et al. 2014b], the generalization aspect of adversarial examples to support adversarial training is explained. Notably, an example generated for a model is often misclassified by other models. The behavior occurs due non-linearlities, and overfitting cannot account for the various different behaviors. However, this method did not lead to fully robust models and could be bypassed by using a multi-step attack such as projected gradient descent (PGD).

For this, Madry et al. [2017] suggested that the local maxima can be found by PGD. The PGD value is calculated from both normal and adversary trained networks to find a similar loss value. From this, the author found that the PGD adversary attacks only yielded robustness against attacks that are on first-order information. The optimal value of PGD produce better results than finding a local maxima if the adversary only uses of the loss function with respect to the input gradients. The authors experimentally demonstrated that such local maxima are hard to find with the first-order method. Most of optimization problem in ML are solved involving the first-order methods and variants of SGD. In this sense, the author believes that an universal attack can

Table VI: Mapping continuous-valued input to quantized inputs, one-hot coding, and thermometer codes.

Real-valued	Quantized	One-hot	Thermometer
0.13	0.15	[0100000000]	[0111111111]
0.66	0.65	[0000001000]	[0000001111]
0.92	0.95	[0000000001]	[0000000001]

also be designed by the first-order information for the current practice of DL. Therefore, they claim that if the trained network is robust against PGD adversaries, it will also be robust against a wide range of attacks that encompasses current approaches.

However, this method did not lead to fully robust models and could be bypassed by using a multi-step attack. For the multi-step attack, [Carlini et al. 2017; Xiao et al. 2018] addressed the difficulty of adversarial training through formal verification techniques. Recent work at ICLR 2018 has begun to construct certified defenses to adversarial examples. These defenses can give proof of robustness that adversarial examples of distortion of at most ϵ cause a test loss of at most δ. Their work was an important direction of research that applies formal verification to the process of constructing provably sound defenses to adversarial examples. However, the major drawback of their approaches was that certified defenses can only be applied to small networks on small datasets.

Adversarial training was originally developed for a small model with MNIST data that did not use batch normalization. Kurakin et al. [2016] extended the original work to ImageNet [Deng et al. 2009] by adding a batch normalization step. The relative weights of adversarial examples are independently controlled in each batch with the following loss function:

$$Loss = \frac{1}{(m-k) + \lambda k} \left(\sum_{\text{CLEAN}} J(x_i|l_i) + \lambda \sum_{\text{ADV}} J(\tilde{x}_i|l_i) \right) \tag{19}$$

where $J(x|l)$ is the loss on a single example x with true class l; m is the total number of training examples in the minibatch; k is the number of adversarial examples in the minibatch and λ is a parameter for the relative weight of adversarial examples in the loss.

In 2018, Tramèr et al. [2019] proposed a defense method with Ensemble adversarial training that is also robust to black-box attacks by containing adversarial examples generated from other models. The approach decouples adversarial example generation from the trained model to increase the diversity of perturbations seen during training. Tramèr et al. [2019] introduced a connection between Ensemble adversarial training and multiple-source domain adaptation [Mansour et al. 2009; Zhang et al. 2012]. Assuming a target distribution takes the role of an unseen black-box adversary, the output has bounded error on attacks from a future black-box adversary.

Obfuscated Gradient Masking The defense approaches against obfuscated gradient masking generally follow three types of obfuscated gradients, which are shattered gradients, stochastic gradients and vanishing/exploding gradients [Athalye et al. 2018].

Having shattered gradients means that incorrect gradients are achieved by making the model intentionally nondifferentiable operationally or unintentionally numerically unstable. The purpose of a shattered gradient attack is to break this linearity with the consideration of the NN, which generally, behaves in a largely linear manner [Athalye and Sutskever 2017]. In the case of images and other high dimensional

space, the linearity will have a large effect on the model's prediction with small values of ϵ, making the model vulnerable to adversarial attacks. A recent defense algorithm over the shattered gradient technique is to exploit thermometer encoding [Buckman et al. 2018] NNs to break the linearity. The method exploits nondifferentiable and non-linear transformations to the input by replacing one-hot encoding with thermometer encoding as shown in Table VI. With the input x, an index $j \in \{i, \cdots, k\}$, and the thermometer $\tau(j) \in \mathcal{R}^K$, the thermometer vector is formally defined as follows:

$$\tau(j)_l = \begin{cases} 1, & \text{if} \quad l \geq j \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Then the thermometer (discretization) function f is defined pixel-wise for a pixel $i \in \{i, \cdots, n\}$ as:

$$f_{\text{therm}(x)_i} = \tau(b(x_i)) = \mathcal{C}(f_{\text{onehot}}(x_i)) \tag{21}$$

where $\mathcal{R}$ is the cumulative sum function, $C(c)_l = \sum_{j=0}^{l} c_l$, and b is a quantization function.

The stochastic gradients render a model to obfuscate the test time by dropping random neurons at each layer. The network stochastically prunes a subset of the activations in each layer during the forward pass. The surviving activations are scaled up to normalize the dynamic range of the inputs to the subsequent layer [Dhillon et al. 2018]. Similarly, **?** proposed a transformation approach under the baseline of image cropping, rescaling [Graese et al. 2016], bit-depth reduction [Xu et al. 2017], JPEG compression [Kingma and Ba 2015], and total variance minimization [Rudin et al. 1992]. The total variance minimization approach first drops pixels in a random manner, and reconstructs images by replacing small patches for overlapping regions to eliminate artificially crafted activations in the edge. The total variation minimization of an image z is formalized as follows:

$$\min_z ||(1 - \tilde{x}) \odot (z - x)||_2 + \lambda_{\text{TV}} \cdot \text{TV}_p(z). \tag{22}$$

where $\tilde{x}$ is a random set of pixels by sampling a Bernoulli random variable $x(i, j, k)$ for pixel location (i, j, k), $\odot$ denotes element-wise multiplication, TV denotes the total variance, and $\text{TV}_p(z)$ represents the L_p total variation in z. Finally, Buckman et al. [2018] demonstrated thermometer code, which improved the robustness to adversarial attacks. Samangouei et al. [2018] proposed Defense-GAN, which is a similar defense method as PixelDefend, but it uses a GAN instead of a PixelCNN.

The vanishing/exploding gradients render the model unusable by deep computation. The basic idea is to purify adversarily perturbed images back to clean examples by exploiting a pixelCNN as a generative model. The purified image is then used for the unmodified classifier. A recent defense algorithm exploits PixelCNN [Oord et al. 2016] to build PixelDefend [Song et al. 2017] to approximate the training distribution. The PixelCNN is a generative model that is designed for images that track likelihood over all pixels by factorizing it into a product of conditional distributions:

$$\mathbf{P}_{\text{CNN}}(x) = \prod_i \mathbf{P}_{\text{CNN}}(x_i | x_{1:(i-1)}). \tag{23}$$

The PixelDefend trained a PixelCNN model on the Canadian Institute for Advanced Research (CIFAR)-10 dataset and used the log-likelihood ratio to approximate the true probability density. Additionally, PixelDefend experimented with adversarial examples from random perturbation (RAND), FGSM, basic iterative method (BIM), Deep-

Fool and CW methods. The result showed that PixelDefend obtained an accuracy above 70% for all attacking techniques, while maintaining satisfactory performance on clean images.

A robustness against iterative optimization attacks is crucial for a good defense system that is built a based on ML. Nevertheless, the existing gradient-based defense algorithm is designed based on the gradient of the initial version, which renders it vulnerable to gradient-based attacks. The attack methods try to find a parameter v such that the image channel, $c(x+v) \neq c^*(x)$, is either maximized or minimized $||v||$. Athalye et al. [2018] exploited projected gradient descent to set v and used the l_2 Lagrangian relaxation approach by Carlini and Wagner [2017a]. With the attack methods, Athalye et al. [2018] showed that most of the obfuscated gradient based defenses are vulnerable to iterative optimization attacks [Kurakin et al. 2018; Madry et al. 2017; Carlini and Wagner 2017a] and have become typical algorithms for evaluating defenses. After all, an attacker may simply use a different attacks [Carlini and Wagner 2017b; Athalye et al. 2018] to bypass these defenses. Defenses that rely on gradient obfuscation can be directly circumvented by any adversary who computes lower bounds on the true adversarial examples [He et al. 2017; Uesato et al. 2018].

Provably Robust Approach Most of DNN are composed of input and output layers with hidden layer in between. The multi hidden layers contain the value of each node, which is determined by calculating the value of a linear combination from the previous layer. Then, non-linear activation is applied to the layers. The authors of Reluplex [Katz et al. 2017] extended the problem of finding a linear combination of hidden layers to finding the reduced search space in means of NP-completeness. They proposed a simplex algorithm to address the issue of these activation functions. The simplex algorithm is based on an SMT (satisfiability modulo theories) solver that includes a SAT (Boolean satisfiability porblem) engine. The engine is involved withnin the SMT solver to compute inputs in the Boolean structure form. This enabled the encoding scheme to be operated to the SMT solver. Exploiting properties from the simplex, they proposed Reluplex that allows variables to violate their bounds temporalily. This feature was necessary since the SMT solver iteratively looks for a feasible next variable assignment. To violate the ReLU semantics, they also allows pairs of ReLU variables.

[Wong and Kolter 2018] proposed a provably robust method against norm-bounded AE. They claimed that their approach guarantees robustness against any norm-bounded AE on the training set. The method involves the convex outer bound approach called "adversarial polytope". The polytope is a set of all final activation layers that are achieved by applying a norm-bounded perturbation to the inputs. They used the convex outer bound as a linear relaxation of the ReLU activation to provided provable guarantees on the robustness of a classifier. To overcome the worst-case of this convex outer bound, they used optimization techniques to the classifier.

Recently, [Raghunathan et al. 2018] proposed a method to avoid the inaccuracy of lower and upper bounds on the worst-case loss. They focused on the upper bound of the loss claiming that it is safer to minimize the upper bound than minimizing the lower bound. This is because it has a disproportional boundary points that have higher objective values. With this, it was first demonstrated a defense method against AE on two-layer networks. The method uses a chain rule for the activation derivatives and uses the fact that an activation function σ (e.g. ReLU, sigmoid) has bounded derivatives $\sigma \in [0, 1]$. For the non-convex optimization problem of the negative semidefinite, a similar approach of semidefinite programming relaxation for MAXCUT [Goemans and Williamson 1995] is used to obtain the upper bound.

*3.3.2. **Defense against Poisoning Attacks**.* The framework proposed by [Steinhardt et al. 2017] takes the approach of removing outliers that are outside the applicable set. In binary classification, these authors aimed to find the centroids of the positive and negative classes. Then, Steinhardt and coworkers removed any points that were distant from the corresponding centroid. To find these points, these researchers used two methods in a complementary way: a sphere defense, by removing the outer points of the spherical radius, and a slab defense that discarded points that were too far away from the line in a complimentary way.

Koh and Liang [2017] used influence functions to track model predictions and identify the most influential data points that are responsible for a given prediction. These authors showed that approximations in functions can still provide important information in nonconvex and nondifferentiable models where the theory breaks down. Additionally, by using influence functions, the defender can check out only the data prioritized by its influence score. This method outperforms previous methods of identifying the greatest training loss for removing the tainted examples.

Paudice et al. [2018b] also suggested a defense mechanism to mitigate the effects of poisoning attacks on the basis of outlier detection. The attacker attempts to have the greatest effect on the defender with a limited number of poisoning points. To mitigate this effect, these investigators first divide the trustworthy dataset $\mathcal{D}$ into different classes, i.e., $\mathcal{D}_+$ and $\mathcal{D}_-$. Then, the curated data are used to train distance-based outlier detectors for each class. The outlier detection algorithm calculates the outlier score for each x in the original (total) data set. There are many ways to measure the outlier score, such as using the SVM or local outlier factor (LOF) as a detector. The empirical cumulative distribution function (ECDF) of training instances is used to calculate the threshold for detecting outliers. By removing all of the samples that are expected to be contaminated, the defender can collect new data sets to retrain the learning algorithm.

Paudice et al. [2018a] chose to relabel data points that are considered outliers instead of removing them. The label flipping attack is a special case of data poisoning that allows an attacker to control the label of a small number of training points. Paudice et al. [2018a] proposed a mechanism that considers the points farthest from the decision boundary to be malicious and reclassifies them. The algorithm reassigns the label of each instance of using a k-NN. For each sample of training data, the closest k-NN points are found first using the Euclidean distance. If the number of data points with the most common label among k-NN is is sequal to or greater than a given threshold, the corresponding training sample is renamed to the most common label in the k-NN.

Chen et al. [2018] uses the activation in the latent space of a neural network rather than analyzing input or output to find poisoned data. Each data is analyzed using the activation of the last layer of the neural network to analyze how much the activation deviates from the majority distribution of activation values from one class. To reduce the dimension of activation before activation clustering, the dimension is reduced to a 1D vector via independent component analysis.

Tran et al. [2018] proposed a method that can frustrate backdoor attacks by using the property called spectral signatures, used in robust statistics. This method spots poisoned data using the activation of a neural network, similar to Chen et al. [2018]. First, the singular value decomposition of the covariance matrix of the activation of all data is calculated. Second, all the data is compared with the obtained top singular vector to find the tainted data, which has a high outlier score. This way, tainted data is erased when re-training the neural network.

The method proposed in [Liu et al. 2018] is different from the defense methods described previously. In the previously explained methods, the strategy is to prevent the attacks by detecting poisoned data and removing them. However, this study tries to
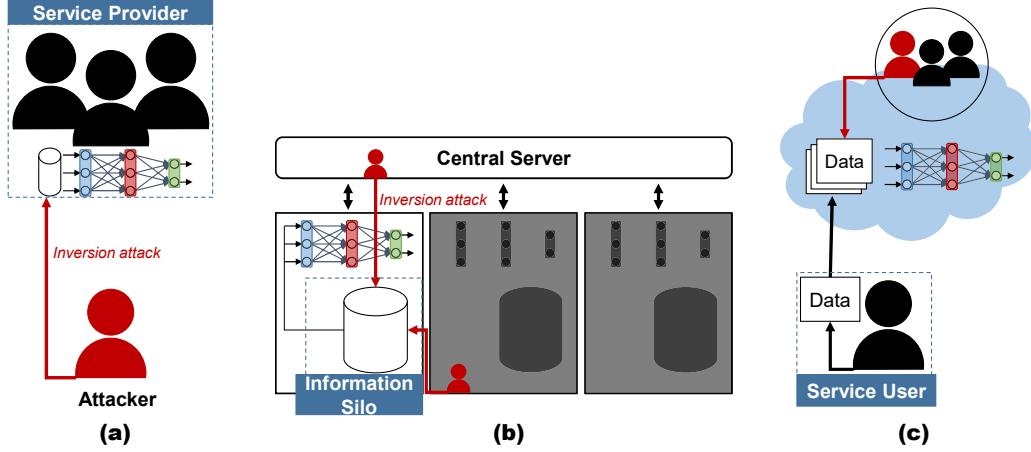
Fig. 8: Private AI: Potential threats from the perspectives of the (a) service provider, (b) information silo, and (c) user.

prevent the attack by modifying the neural network itself. They proposed a methodology called fine-pruning, which is a combination of pruning and fine-tuning. By pruning the neural network, it is possible to remove neurons, such as backdoor neurons, demonstrated by Gu et al. [2017]. However, because other attacks are made pruning-aware, this method also suggested cleaning the neural network through fine-tuning after pruning using trusted clean data. Using these two steps lead to development of a robust neural network model for multiple poisoning attacks. **?** also presented a similar method to Liu et al. [2018]. They proposed a method that prunes filters of a neural network that trigger the backdoor attack.

4. PRIVATE AI

DL algorithms that account for most of the current AI systems rely highly on data. Hence, DL is always exposed to privacy threats, and the privacy of the training data should be preserved. Hence we define *Private AI* as an AI system that preserves the privacy of the concerned data.

4.1. Potential Threats from Different Perspectives

*4.1.1. **Potential Threats from the Service Providers Perspective.**.* When companies provide DL models and services to the public, there are potential risks in that the models leak private information even without revealing the original dataset. As shown in Fig. 8(a), a model inversion attack occurs when an adversary uses the pretrained DL model to discover the data used in the model training. Such attacks are intended to manipulate the relations between the target and the unknown input with the following model output.

Recent studies on inversion attacks show that a model inversion attack is possible, by recovering training images [Fredrikson et al. 2015] or performing a membership test to determine whether a dataset contains an individual's information [Shokri et al. 2017a]. Furthermore, deployed DL services are exposed to data integrity attacks as well. Because the deployed service demands the user's data, adversaries can try to break the integrity of the data at the servicing server. If a data holder includes the collected data without an integrity check, the broken integrity can mislead or even ruin the model.

*4.1.2. **Privacy Violation in Information Silos***. Information silos are a group of related exclusive data management systems and training a joint DNN model under such settings was studied as federated learning [McMahan et al. 2017; Konečnỳ et al. 2016; Bonawitz et al. 2019]. However, to securely compute a DNN without revealing the private data in each silo, as shown in Fig. 8(b), secure multi-party computation (SMC) methods can be applied [Lindell 2005].

Similarly, the idea of secure multi-party training, which trains a joint DL model of private data input, has been emerging. In such training processes, the individual's data privacy should be preserved for any adversary including an external party. Hitaj et al. [2017] showed that a distributed or federated DL approach is essentially broken in terms of privacy stability. Such a structure makes it difficult to protect the training sets of honest participants from a GAN-based attack. An adversary based on GAN fools a victim to reveal more sensitive data.

*4.1.3. **Potential Threats from a User's Perspective***. Because many DL-based applications have been introduced in industry, such service users are under serious threats of the privacy invasion [Gilad-Bachrach et al. 2016; Sanyal et al. 2018]. Because DL models are too large and complicated [He et al. 2016; Huang et al. 2017a] to be computed on small devices such as mobile phones or smart speakers, most service providers require users to upload their sensitive data, such as their voice recordings or face images, to compute on their (cloud) servers. The problem is that upon uploading, the users lose control of their data. In other words, the users cannot delete their data and cannot check how their data is used, as shown in Fig. 8(c). As the recent Facebook's privacy scandal suggests, even when there are some privacy policies, it is difficult to notice or restrain from excessive data exploitation. In addition, because hardware requirements for DL are enormous, machine-learning-as-a-service (MLaaS) provided by Google, Microsoft, or Amazon has gained in popularity among DL-based service providers. Such remote servers make it difficult to manage the users' data privacy.

4.2. Defense Techniques Against Potential Threats

Unlike many attacks that are attempted in the domain of Secure AI, only a few attacks are attempted in the field of Private AI as well as defense with respect to privacy-preserved DL. We observed that this finding is due to the nature of privacy preserving techniques. Exploiting traditional security in the field of DL requires encryption and decryption phases, which make it impractical in the real world due to the enormous computational complexity. As a result, HE is one of the few security techniques that can be exploited in DL. As one further step to Private AI, the DP technique is actively exploited in DL. In the following section, we detail Private AI, which adopts the most recent privacy-preserving methods.

*4.2.1. **Homomorphic Encryption on Deep Learning***. CryptoNets [Gilad-Bachrach et al. 2016] took the initiative of applying NNs for inferencing on the encrypted data. CryptoNets utilize the leveled HE scheme YASHE [Bos et al. 2013] for the privacy-preserving inference on a pretrained CNN model. This method has demonstrated over a 99% accuracy in a classification task using handwritten digits (MNIST data set [Le-Cun et al. 2010]). However, leveled HE leads to a serious degradation in model accuracy and efficiency. Furthermore, because the square activation function is replaced by nonpolynomial activation and the converted precision of the weights, the inferencing model obtains results that are quite different from the trained model. Hence, it is not suited for recent complicated models [He et al. 2016; Huang et al. 2017a]. In addition, the latency of the computation is still on the order of hundreds of seconds, while Gilad-Bachrach et al. [2016] achieved a throughput of 50,000 predictions in an hour. Cryptonets also allow a handful of operations to handle them: the batch and image layout

functions. The batch operation is useful for classifying a large number of samples, and the image layout function is useful for categorizing samples together. However, neither operation is suitable for classifying only one image. In return, CryptoDL [Hesamifard et al. 2017] and Chabanne et al. [2017] attempted to improve CryptoNets by low degree polynomial approximations on the activation functions. Chabanne et al. [2017] applied batch normalization to reduce the accuracy difference between the actual trained model and the converted model with an approximated activation function at the inference phase. The batch normalization technique also enabled fair predictions on a deeper model.

Because a recent bootstrapping FHE technique was introduced [Chillotti et al. 2016], TAPAS [Sanyal et al. 2018] and FHE-DiNN [Bourse et al. 2018] were proposed. Because the method proposed by Chillotti et al. [2016] supports operations on binary data, both utilized the concept of binary neural networks (BNNs) [Courbariaux et al. 2016]. FHE-DiNN [Bourse et al. 2018] utilized discretized NNs with different weights and input dimensions to evaluate Chillotti et al. [2016] on DNNs. In comparison, TAPAS [Sanyal et al. 2018] used binarized weights and enabled binary operations and sparsification techniques. Both FHE-DiNN and TAPAS showed faster prediction than the approaches based on leveled HE. Notably, while leveled HE methods only support batch predictions, bootstrapping FHE-based methods enabled predictions on single instances, which is more practical.

4.2.2. **Secure Multiparty Computation (SMC) on Deep Learning**. To date, there are two known major types of privacy-preserving DL algorithms related to multiple parties. The first type of algorithm is based on a conventional distributed DL algorithms [Dean et al. 2012; Abadi et al. 2016a; Lee et al. 2018] that enables the parties to participate in training or testing DL models without revealing their data or models. The other type is based on a secure two-party computation (2PC) combined with the HE and a garbled circuit (GC). Such algorithms assume two parties: a user who provides data and a server that implements DL based on the provided data. Modern cryptography techniques combined with (SMC) techniques such as oblivious transfer, have attempted to securely protect the data transfer process as well.

Distributed selective SGD (DSSGD) [Shokri and Shmatikov 2015] proposed collaborative DL protocols with different data holders to train joint DL models without sharing their training data. This approach is very similar to prior distributed DL algorithms [Dean et al. 2012; Abadi et al. 2016a; Lee et al. 2018]. With the coordinated learning models and objectives, the participants train their local models and selectively exchange their gradients and parameters at every local SGD epoch asynchronously. However, because DSSGD assumes the use of a parameter server [Li et al. 2014], Aono et al. [2018] noted that even with a few gradients, it is possible to restore the data used in training. Hence, to preserve privacy against the honest-but-curious parameter server, learning with errors (LWE)-based HE was applied by exchanging weights and gradients. However, the improved privacy achieved by HE trades off with the communication costs.

Secure 2PC algorithms include SecureML [Mohassel and Zhang 2017], MiniONN [Liu et al. 2017], DeepSecure [Rouhani et al. 2018] and Gazelle [Juvekar et al. 2018]. SecureML [Mohassel and Zhang 2017] is the first privacy preserving method to train NNs in multiparty computation settings. Using SMC and secret sharing, SecureML can train ML algorithms, such as linear regression, logistic regression and NNs. Although the authors [Mohassel and Zhang 2017] have attempted to speed up the computation, SecureML still requires large amounts of communication. MiniONN [Liu et al. 2017] transforms the original NN into an *oblivious NN* for training using a simplified HE. MiniONN also utilized GCs to approximate the nonlinear activation func-

tions. DeepSecure [Rouhani et al. 2018] computes encrypted data inference on the DL model using Yao's GCs [Yao 1986] and suggests some practical computing structures with proof of security. As the authors of [Juvekar et al. 2018] noted, the aforementioned work showed that HE mainly shows strength in matrix-vector multiplications but is restricted to linear operations. On the other hand, GCs can cause serious communication overhead and are more suited for approximating nonlinear functions in DNN models. Hence, Gazelle [Juvekar et al. 2018] combines HE and GC and computes linear operations with HE and activation functions with GC. Attempts to build a privacy preserving federated learning framework that combines MPC and DP functionality have also been made [Ryffel et al. 2018].

4.2.3. ***Differential Privacy on Deep Learning***. By applying DP to DL models, the training data can be protected from inversion attacks when the model parameters are released. Hence, there are many studies that utilize DP to DL models. Such methods assume that the training datasets and model parameters are the database and the responses, respectively, and prove that their algorithms satisfy either Equation 5 or 6.

Depending on where the noise is added, such approaches can be divided into three groups: gradient-level [Abadi et al. 2016b; McMahan et al. 2018; Xie et al. 2018a; Acs et al. 2018; Yu et al. 2019], objective-level [Chaudhuri and Monteleoni 2009; Phan et al. 2016, 2017a,b] and label-level [Papernot et al. 2017, 2018; Triastcyn and Faltings 2018]. Fig. 9 shows the overview of these approaches. The gradient level approach inserts noise into the parameter's gradients of the parameters during the training phase. The objective-level approach introduces a noise that perturbs the coefficients of the original objective function. The label-level approach introduces noise into the label in the knowledge transfer phase of the teacher-student model.

The gradient-level approach [Abadi et al. 2016b] proposed a differential private SGD (DP-SGD) algorithm that adds noise to the gradients in the batchwise updates. It is important to estimate the accumulated privacy loss as learning progresses by batch. In particular, the authors of [Abadi et al. 2016b] proposed the moment accountant algorithm to track the cumulative privacy loss. The moment accountant algorithm considers privacy loss as a random variable and estimates the tail bound of it. The resulting bounds provide a tighter level of privacy than using the basic or strong composition theorems [Dwork et al. 2006, 2010]. McMahan et al. [2018] introduced user-level differentially private LSTM. In language modeling, it is difficult and ineffective to keep privacy at the word level. Therefore, McMahan et al. [2018] defined user-level adjacent datasets and ensured DP for users. Xie et al. [2018a] proposed a differentially private generative adversarial network (DPGAN). DPGAN injected noise into the gradient of the discriminator to obtain a differentially private discriminator and a generator which is trained with that discriminator, and also to become differentially private based on post-processing theory [Dwork et al. 2014].

Acs et al. [2018] introduced a differentially private generative model that has a mixture of k generative NNs, such as the restricted Boltzmann machine (RBM) [LeCun et al. 2015] and the variational autoencoder (VAE) [Kingma and Welling 2013]. These researchers applied a differentially private kernel k-means algorithm for clustering the original datasets and used DP-SGD [Abadi et al. 2016b] to train each neural network. Additionally, these coworkers extended the differentially private k-means clustering [Blum et al. 2005] by applying random Fourier features [Chitta et al. 2012] and improved the accuracy of the trained model by carefully adjusting injected noise in the DP-SGD framework.

Yu et al. [2019] introduced some techniques that can be utilized for DP-SGD [Abadi et al. 2016b]. Abadi et al. [2016b] assumed that the bathing method for mini-batch SGD is random sampling; however, in practice, random reshuffling is a widely used
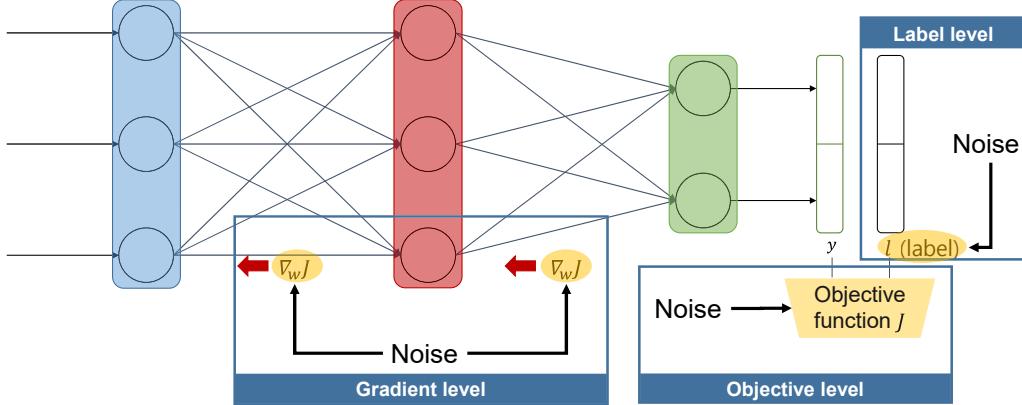
Fig. 9: Overview of the differential privacy in a DL framework.

batching method. Yu et al. [2019] suggested privacy accounting methods for each case and analyzed the characteristics. Additionally, these authors applied concentrated DP (CDP) [Bun and Steinke 2016] to achieve a tighter estimation for a large number of iterations and dynamic privacy budget allocation mechanisms to improve the performance.

An objective-level approach [Chaudhuri and Monteleoni 2009] disturbs the original objective function by adding noise to the coefficients. Then, the model trained on the disturbed objective function is differentially private. Unlike the gradient-level approach, whose privacy loss is accumulated as training progresses, the privacy loss as a result of the objective-level approach is determined during the building of the objective function and is independent of the epochs. To inject noise into the coefficients, the objective function should be a polynomial representation of the weights. If an objective function is not in a polynomial form, the objective-level approach approximates it to a polynomial representation using approximation techniques such as Taylor or Chebyshev expansion. Then, the noise is added to each coefficient to obtain the disturbed objective function. Chaudhuri and Monteleoni [2009] proposed the differentially private logistic regression, whose parameters are trained based on the perturbed objective function. The functional mechanism not only is applied to logistic regression but also is applied to various models, such as autoencoders [Bengio et al. 2009] and convolutional deep belief networks [Lee et al. 2009]. Phan et al. [2016] proposed a deep private autoencoder (dPA) and proved that the dPA is differentially private based on the functional mechanism. Phan et al. [2017a] introduced the private convolutional deep belief network (pCDBN) and utilized the Chebyshev expansion to approximate the objective function to the polynomial form. Phan et al. [2017b] developed a novel mechanism called the adaptive Laplace mechanism (AdLM). The key concept is adding noise to the input features with minimum disturbance to the model output.

Phan et al. [2017b] injects noise from the Laplace distribution into the layer-wise relevance propagation (LRP) [Bach et al. 2015] to estimate the relevance of the correlation between inputs and outputs. These authors applied an affine transformation based on the estimated relevance to adaptively distribute the noise. The AdLM also applies a functional mechanism that perturbs the objective function. These differential private actions are processed before training the model.

The label-level approach injects noise into the knowledge transfer phase of the teacher-student framework. Papernot et al. [2017] proposed the semi-supervised

knowledge transfer model referred to as the private aggregation of teacher ensemble (PATE) mechanism. The PATE is a type of teacher-student model whose purpose is to train a differentially private classifier (student) based on an ensemble of non-private classifiers (teacher). Each teacher model learns on disjoint training datasets, and the output of the teacher ensemble is determined by the noisy aggregation of each teacher's prediction. The noisy aggregation introduces a noisy label that meets DP, and then, the student model learns the noisy label from the teacher ensemble as a target label. Because the student model cannot access the training data directly and the differential private noise is injected into the aggregation process, PATE ensures safety intuitively and in terms of the DP. PATE utilizes the *moment accountant* to trace the cumulated privacy budget in the learning process. Later, Papernot et al. [2018] extended the PATE to operate in a large scale environment by introducing a new noisy aggregation mechanism. These researchers showed that the improved PATE outperforms the original PATE on all measures and has high utility with a low privacy budget. Triastcyn and Faltings [2018] applied the PATE to build the differential private GAN framework. By using PATE as a discriminator, the generator trained with the discriminator is also differentially private.

5. DISCUSSION

5.1. Challenges and Future Research Directions

From Section 3, we reviewed security issues on DL models. For the first part of Section 3, we confirmed that attack methods can fool or subvert DL models. We reviewed two types of attack scenarios: evaision attacks and popisoning attacks Hence, to defend against such attack methods, many researchers proposed diverse gradient masking defense methods, and these methods showed decent results by involving more nonlinearity in a model or preventing the gradients of the model from being copied by an adversary. As the authors of [Athalye et al. 2018] suggest, proper gradient methods demonstrate powerful defense performance.

To achieve higher performance on both attack and defense strategies, the interpretation approaches of the model is the one crucial fundamental aspect. It is known that adversary examples are an important key features for the robustness of the neural networks as one significant outlier for the success of misclassification [Ilyas et al. 2019]. Therefore, we believed to be beneficial if interpretable AI approaches [Simonyan et al. 2013; Bach et al. 2015; Shrikumar et al. 2017] are applied to such an attack or defense methods. Interpretable AI analyzes the underlying functions of the DL model and determines the way that a DL model makes predictions. With a deeper understanding of DL models, it is feasible to design a system (model) robust to unseen attacks by identifying blind spots that should be considered and addressed. In this sense, security issues should be constantly researched and developed considering different perspectives [Ilyas et al. 2019; Tsipras et al. 2019] on adversarial attacks.

For the second part of Section 3, we reviewed some poisoning attack methods on deep learning models. Poisoning attacks are designed to change the labels of training data or to change the pixel values of images. Different attacks have successfully achieved their expected objectives in various scenarios on different applications. Recent approaches to protect data include outlier detection to eliminate [Paudice et al. 2018b] or relabel [Paudice et al. 2018a] suspected poisoned examples. Poisoning samples have a distribution that is much different from the benign training data, so attacks were classified as an outlier and could be detected in the learning process by human experts. In contrast to an outlier, if the attack is made undetectable to human expert, a problem arises that the effectiveness of the attack decreases. The necessity of a methodology for enhancing the effectivity while reducing the detectability has

emerged. To this end, attempts are being made to control the attack detectability and effectivity training using the decision boundary of the model. Hence, we metrics or evaluation methods to determine if the protected model is secure.

Moreover, DNNs are vulnerable to the aforementioned attacks although there are defenses on DL models. Thus, it is important to tackle not only the integrity aspect of the DL models as well as methods to verify robustness aspects of the DL models. Therefore, it is important account the robustness of the model is against such attacks for practical use. There have been numerous studies on the verification of the deep learning framework [Tian et al. 2018; Katz et al. 2017; Wong and Kolter 2018; Wang et al. 2018; Singh et al. 2019]. These studies form an automatic test to verify that DNN models used in autonomous driving work well under certain conditions [Tian et al. 2018], or formally analyze the robustness of DNNs against input perturbations [Katz et al. 2017; Wong and Kolter 2018; Wang et al. 2018; Singh et al. 2019]. However, current verification methods mainly target norm-ball perturbations, and more general verification methods should be studied.

In addition, in Section 4.2.1, we reviewed privacy-preserving DL models with the full HE cryptosystems. Although recent methods achieve a high prediction rate despite the strict encryption, the accuracy performance falls behind the state-of-the-art model performances and is not compatible with deeper models. The main reason for this situation is that the FHE methods used in those papers do not include the nonlinear activation functions discussed in Section 2.1.1. Hence, current FHE-based prediction models use different models from the actual trained models. In other words, these models train the unencrypted data on the unencrypted typical models, and then, the trained weights and biases are applied to a different model, in which the activation functions are replaced with simple activations such as square functions. A discrepancy between the training and inference models usually causes high degradation of the prediction accuracy. To overcome this deterioration, two approaches are possible: either train the same model from the beginning, or properly transfer the model. As the authors of [Hinton et al. 2015] suggest, the knowledge learned by a DL model can be distilled into another model.

In Section 4.2.3, a large number of attempts were confirmed using DP to protect data privacy in DL training. Such methods add noise to the gradients or objective functions to confuse the attacker and give closed-form proof on the DP bounds of the proposed methods. However, from a DL researcher's perspective, such bounds are insufficient to give practical insights on whether or not such a privacy bound is strong enough. If DP researchers can provide experiments on the assumed attack scenarios or some practical evaluations or metrics, the proposed methods would be much more informative.

In recent studies, membership inference attack, and model inversion attack threaten data privacy, and hyperparameter stealing attack threaten model privacy [Wang and Gong 2018]. Given the data and the pre-trained target model, a membership inference attack (MIA) [Shokri et al. 2017b; Song and Shmatikov 2018; Hayes et al. 2019] model determines whether the given data was used for the training step of the target model. Some studies [Sablayrolles et al. 2019; Truex et al. 2019] introduced the reason of feasibility of MIA as the over-fitting of the model. The deep neural networks can memorize the training data itself rather than learn the latent properties of the data, and it means that the neural networks over-fit to the training data. Inversion attacks aim to get attributes of input data which is used for training the target model. Fredrikson et al. [2015] reconstructed a face image which was similar to the input face image by utilizing the confidence score of the target model. Salem et al. [2019] inversion attacked the online-learning model based on the difference between before and after updating the model. They modeled that difference by a generative adversarial networks. Wang and Gong [2018] proposed hyperparameter stealing attacks, and they

succeeded in attacks on Amazon Machine Learning [ama 2017]. The machine learning models have different performances depending on the hyperparameters, which leads to valuable information. To prevent such attacks, there are limited studies on defense mechanisms against hyperparameter stealing attacks [Wang and Gong 2018]. MLaaS providers should protect users' data and their hyperparameters against these specific attack methods. Overall, studies are required to present whether HE and DP are indeed robust against these attacks.

5.2. Practical Issues and Suggestions for Deployment

The DL model variants might pose threats to model security and data privacy. Because DL models are very complicated, it is difficult to think of a new model structure. Hence, once a model structure is deployed, a large number of users add some variants for their specific uses and train further with their own data. In the case of U-Net [Ronneberger et al. 2015], which is a CNN model used for image segmentation in the biomedical field, there are several proposed variants [Çiçek et al. 2016; Li et al. 2018; Jo et al. 2018]. If such similar models are deployed in public, it is likely that they will be susceptible to the attacks reviewed in this paper. These large-variant models might provide clues in building substitute models in black-box attack scenarios, or induce easier inversion attacks based on the accumulated knowledge from similar models. Hence, we must be careful when deploying models for the large number of variants.

Practical considerations on the processing time and throughput are needed as well. Although FHE combined with DL predictions showed remarkable performances with regard to both privacy and utility, it lacks considerations of practical implementations. Because predictions on FHE data and models are still too slow, parallel or distributed processing using graphic processing units (GPUs) or clusters is crucial. In particular, because GPUs have already achieved high computational speeds in DL training, combining GPUs high computing power with FHE model prediction is promising. Considering those situations in which FHE is needed on predictions where the computing resources of the user devices are insufficient, on-device encryption and decryption should be considered as well.

6. SUMMARY

DL has become an inseparable technology in our daily lives, and the problem of security and privacy of DL has become an issue that can no longer be overlooked. Therefore, we have defined Secure AI and Private AI and reviewed the related attack and defense methods.

In Secure AI, we surveyed the two types of attacks: the evasion attack and the poisoning attack. We then further categorized the attack scenarios as white-box and black-box attacks, according to the amount of information and the authority of the model that the adversary possesses. In this process, we confirmed that many research studies have been conducted with advanced and varied attack methods. However, studies on defense techniques are in the relatively early stages. In this paper, we introduced related studies by classifying them as gradient masking, adversarial training and statistical approaches.

Furthermore, the risk of data privacy violations has always been widespread due to the characteristics of DL, which highly relies on an extensive amount of data, and the era of the fourth industrial revolution, in which data itself is the enormous asset. In this paper, we described the possible threats to data privacy from the perspectives of DL models and service providers, information silos and DL-based service users. In addition, we named the DL-based approaches that are concerned with data privacy as Private AI. Unlike Secure AI, there are few studies on privacy attacks using DL. Hence, we introduced recent studies on three defending techniques concerned with

Private AI: HE, DP, and SMC. Finally, open problems and directions for future work are discussed.

REFERENCES

2017. AMAZON ML SERVICES. (2017). https://aws.amazon.com/cn/machine-learning

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, and others. 2016a. Tensorflow: a System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016b. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.

Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. 2018. Differentially Private Mixture of Generative Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* (2018).

Scott Alfeld, Xiaojin Zhu, and Paul Barford. 2016. Data Poisoning Attacks Against Autoregressive Models.. In *AAAI Conference on Artificial Intelligence*.

Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, and others. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security* 13, 5 (2018), 1333–1345.

Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers. *International Journal of Security and Networks* 10, 3 (2015), 137–150.

Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*.

Anish Athalye and Ilya Sutskever. 2017. Synthesizing Robust Adversarial Examples. *arXiv preprint arXiv:1707.07397* (2017).

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PloS one* 10, 7 (2015), e0130140.

Shumeet Baluja and Ian Fischer. 2017. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. *arXiv preprint arXiv:1703.09387* (2017).

Vahid Behzadan and Arslan Munir. 2017. Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*.

Josh Benaloh. 1994. Dense Probabilistic Encryption. In *Workshop on selected areas of cryptography*.

Yoshua Bengio and others. 2009. Learning Deep Architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks Against Machine Learning at Test Time. In *Joint European conference on machine learning and knowledge discovery in databases*.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. 2005. Practical Privacy: the SuLQ Framework. In *24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Inger-man, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Bren-dan McMahan, and others. 2019. Towards Federated Learning at Scale: System Design. In *Conference on Systems and Machine Learning*.

Joppe W Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. 2013. Improved Secu-rity for a Ring-Based Fully Homomorphic Encryption Scheme. In *IMA International Conference on Cryptography and Coding*.

Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. 2018. Fast Homomorphic Evaluation of Deep Discretized Neural Networks. In *Annual Interna-tional Cryptology Conference*.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2014. (Leveled) Fully Ho-momorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory (TOCT)* 6, 3 (2014), 13.

Zvika Brakerski and Vinod Vaikuntanathan. 2014. Efficient Fully Homomorphic En-cryption From (standard) LWE. *SIAM J. Comput.* 43, 2 (2014), 831–871.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adver-sarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *International Conference on Learning Representations*.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer Encoding: One Hot Way to Resist Adversarial Examples. In *International Conference on Learning Representations*.

Anna L Buczak and Erhan Guven. 2016. A Survey of Data Mining and Machine Learn-ing Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials* 18, 2 (2016), 1153–1176.

Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifica-tions, Extensions, and Lower Bounds. In *Theory of Cryptography Conference*.

Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. 2017. Provably Minimally-Distorted Adversarial Examples. *arXiv preprint arXiv:1709.10207* (2017).

Nicholas Carlini and David Wagner. 2017a. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *10th ACM Workshop on Artificial Intelligence and Security*.

Nicholas Carlini and David Wagner. 2017b. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*.

Nicholas Carlini and David Wagner. 2018. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. *2018 IEEE Security and Privacy Workshops* (2018).

Hervé Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, and Em-manuel Prouff. 2017. Privacy-Preserving Classification on Deep Neural Network. *IACR Cryptology ePrint Archive* (2017).

Kamalika Chaudhuri and Claire Monteleoni. 2009. Privacy-Preserving Logistic Re-gression. In *Advances in Neural Information Processing Systems*.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017b. Zoo: Zeroth Order Optimization Based Black-Box Attacks To Deep Neural Networks without Training Substitute Models. In *10th ACM Workshop on Artificial Intelli-gence and Security*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017a. Targeted Backdoor Attacks on Deep Learning Systems using Data Poisoning. *arXiv preprint arXiv:1712.05526* (2017).

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. 2016. Faster Fully Homomorphic Encryption: Bootstrapping in Less than 0.1 Seconds. In *International Conference on the Theory and Application of Cryptology and Information Security*.

Radha Chitta, Rong Jin, and Anil K Jain. 2012. Efficient Kernel Clustering using Random Fourier Features. In *IEEE International Conference on Data Mining*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 2016. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or-1. *arXiv preprint arXiv:1602.02830* (2016).

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, and others. 2012. Large Scale Distributed Deep Networks. In *Advances in Neural Information Processing Systems*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. 2018. Stochastic Activation Pruning for Robust Adversarial Defense. In *International Conference on Learning Representations*.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, Jianguo Li, and Jun Zhu. 2017. Boosting Adversarial Attacks with Momentum. *arXiv preprint arXiv:1710.06081* (2017).

Léo Ducas and Daniele Micciancio. 2015. FHEW: Bootstrapping Homomorphic Encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*.

Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *International Conference on Theory and Applications of Models of Computation*.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy via Distributed Noise Generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*.

Cynthia Dwork and Jing Lei. 2009. Differential Privacy and Robust Statistics. In *41st Annual ACM Symposium on Theory of Computing*.

Cynthia Dwork, Aaron Roth, and others. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and Differential Privacy. In *IEEE Annual Symposium on Foundations of Computer Science*.

Taher ElGamal. 1985. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE transactions on information theory* 31, 4 (1985), 469–472.

Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial Examples That Fool Both Computer Vision and Time-Limited Humans. In *Advances in Neural Information Processing Systems*.

Samuel G Finlayson, Isaac S Kohane, and Andrew L Beam. 2018. Adversarial Attacks Against Medical Deep Learning Systems. *arXiv preprint arXiv:1804.05296* (2018).

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*.

Craig Gentry and Dan Boneh. 2009. A Fully Homomorphic Encryption Scheme. (2009).

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *International Conference on Machine Learning*.

Michel X Goemans and David P Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems using Semidefinite Programming. *J. ACM* 42, 6 (1995), 1115–1145.

Shafi Goldwasser and Silvio Micali. 1982. Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information. In *14th Annual ACM Symposium on Theory of Computing*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014a. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014b. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572* (2014).

Abigail Graese, Andras Rozsa, and Terrance E Boult. 2016. Assessing Threat of Adversarial Examples on Deep Neural Networks. In *IEEE International Conference on Machine Learning and Applications*.

Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2016. Adversarial Perturbations Against Deep Neural Networks for Malware Classification. *arXiv preprint arXiv:1606.04435* (2016).

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint arXiv:1708.06733* (2017).

Zhaoyuan Gu, Zhenzhong Jia, and Howie Choset. 2018. Adversary A3C for Robust Reinforcement Learning. (2018).

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2018. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyJ7ClWCb

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. LOGAN: Membership Inference Attacks Against Generative Models. *Privacy Enhancing Technologies* 2019, 1 (2019), 133–152.

Elad Hazan and others. 2016. Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization* 2, 3-4 (2016), 157–325.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE conference on Computer Vision and Pattern Recognition*.

Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. 2017. Adversarial Example Defense: Ensembles of Weak Defenses Are Not Strong. In *USENIX Workshop on Offensive Technologies*.

Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. 2017. CryptoDL: Deep Neural Networks over Encrypted Data. *arXiv preprint arXiv:1711.05189* (2017).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *In ACM SIGSAC Conference on Computer and Communications Security*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017a. Densely Connected Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017b. Adversarial Attacks on Neural Network Policies. *arXiv preprint arXiv:1702.02284* (2017).

Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin, Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *35th International Conference on Machine Learning*.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. 2019. Prior Convictions: Black-Box Adversarial Attacks with Bandits and Priors. In *International Conference on Learning Representations*.

Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. 2017. The Robust Manifold Defense: Adversarial Training using Generative Models. *arXiv preprint arXiv:1712.09196* (2017).

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems*.

YoungJu Jo, Hyungjoo Cho, Sang Yun Lee, Gunho Choi, Geon Kim, Hyun-seok Min, and YongKeun Park. 2018. Quantitative Phase Imaging and Artificial Intelligence: A Review. *IEEE Journal of Selected Topics in Quantum Electronics* 25, 1 (2018), 1–14.

Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. 2019. Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers. *arXiv preprint arXiv:1904.08489* (2019).

Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. 2018. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In *27th USENIX Security Symposium*.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2017. The Composition Theorem for Differential Privacy. *IEEE Transactions on Information Theory* 63, 6 (2017), 4037–4049.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *International Conference on Computer Aided Verification*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).

Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *34th International Conference on Machine Learning*.

Jakub Konečný, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence.

arXiv preprint arXiv:1610.02527 (2016).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial Machine Learning at Scale. *arXiv preprint arXiv:1611.01236* (2016).

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial Examples in the Physical World. In *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC, 99–112.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521, 7553 (2015), 436.

Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST Handwritten Digit Database. (2010). http://yann.lecun.com/exdb/mnist

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. 2009. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *26th Annual International Conference on Machine Learning*.

Seil Lee, Hanjoo Kim, Jaehong Park, Jaehee Jang, Chang-Sung Jeong, and Sungroh Yoon. 2018. TensorLightning: A Traffic-efficient Distributed Deep Learning on Commodity Spark Clusters. *IEEE Access* (2018).

Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling Distributed Machine Learning with the Parameter Server.. In *Symposium on Operating Systems Design and Implementation*.

Shaofeng Li, Benjamin Zi Hao Zhao, Jiahao Yu, Minhui Xue, Dali Kaafar, and Haojin Zhu. 2019. Invisible Backdoor Attacks Against Deep Neural Networks. *arXiv preprint arXiv:1909.02742* (2019).

Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. 2018. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes. *IEEE Transactions on Medical Imaging* 37, 12 (2018), 2663–2674.

Yehida Lindell. 2005. Secure Multiparty Computation for Privacy Preserving Data Mining. In *Encyclopedia of Data Warehousing and Mining*. IGI Global, 1005–1009.

Jian Liu, Mika Juuti, Yao Lu, and N Asokan. 2017. Oblivious Neural Network Predictions via Minionn Transformations. In *ACM SIGSAC Conference on Computer and Communications Security*.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium*.

Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. 2017. Towards Measuring Membership Privacy. *arXiv preprint arXiv:1712.09136* (2017).

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Michael E Houle, Grant Schoenebeck, Dawn Song, and James Bailey. 2018. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality. *arXiv preprint arXiv:1801.02613* (2018).

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv preprint arXiv:1706.06083* (2017).

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain Adaptation: Learning Bounds and Algorithms. In *22nd Annual Conference on Learning The-*

ory.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks From Decentralized Data. In *The 20th International Conference on Artificial Intelligence and Statistics*.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations. In *International Conference on Learning Representations*.

Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *IEEE Symposium on Security and Privacy*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal Adversarial Perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. 2017. Towards Poisoning of Deep Learning Algorithms with Back-Gradient Optimization. In *ACM Workshop on Artificial Intelligence and Security*.

Luis Muñoz-González, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, and Emil C Lupu. 2019. Poisoning Attacks with Generative Adversarial Nets. *arXiv preprint arXiv:1906.07773* (2019).

Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. 2018. Cascade Adversarial Machine Learning Regularized with a Unified Embedding. In *International Conference on Learning Representations*.

Nina Narodytska and Shiva Prasad Kasiviswanathan. 2017. Simple Black-box Adversarial Perturbations for Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel Recurrent Neural Networks. In *33rd International Conference on Machine Learning*.

Pascal Paillier. 1999. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2017. Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data. In *International Conference on Learning Representations*.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in Machine Learning: From Phenomena to Black-Box Attacks Using Adversarial Samples. *arXiv preprint arXiv:1605.07277* (2016).

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In *Asia Conference on Computer and Communications Security*.

Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016a. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy*.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016b. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *IEEE Symposium on Security and Privacy*.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *6th Interna-*

tional Conference on Learning Representations.

Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. 2018b. Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. *arXiv preprint arXiv:1802.03041* (2018).

Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. 2018a. Label Sanitization Against Label Flipping Poisoning Attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. 2016. Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction.. In *AAAI Conference on Artificial Intelligence*.

NhatHai Phan, Xintao Wu, and Dejing Dou. 2017a. Preserving Differential Privacy in Convolutional Deep Belief Networks. *Machine Learning* 106, 9-10 (2017), 1681–1704.

NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. 2017b. Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning. In *IEEE International Conference on Data Mining*.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. In *International Conference on Learning Representations*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*.

Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. 2018. DeepSecure: Scalable Provably-Secure Deep Learning. In *5th ACM/ESDA/IEEE Design Automation Conference*.

Leonid I Rudin, Stanley Osher, and Emad Fatemi. 1992. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D: nonlinear phenomena* 60, 1-4 (1992), 259–268.

Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017* (2018).

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. 2019. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *36th International Conference on Machine Learning*.

Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2019. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. *arXiv preprint arXiv:1904.01067* (2019).

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In *International Conference on Learning Representations*.

Amartya Sanyal, Matt J Kusner, Adrià Gascón, and Varun Kanade. 2018. TAPAS: Tricks to Accelerate (encrypted) Prediction As a Service. In *International Conference in Machine Learning*.

Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.).

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *ACM SIGSAC Conference on Computer and Communications Security*.

Yash Sharma and Pin-Yu Chen. 2018. Attacking the Madry Defense Model with L_1-based Adversarial Examples. In *International Conference on Learning Representations Workshop*.

Benjamin Shickel, Patrick Tighe, Azra Bihorac, and Parisa Rashidi. 2017. Deep EHR: A Survey of Recent Advances on Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *Journal of Biomedical and Health Informatics* (2017).

Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *22nd ACM SIGSAC Conference on Computer and Communications Security*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017a. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017b. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *34th International Conference on Machine Learning*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034* (2013).

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. 2019. An Abstract Domain for Certifying Neural Networks. *Proceedings of the ACM on Programming Languages* 3, POPL (2019), 41.

Aman Sinha, Hongseok Namkoong, and John Duchi. 2017. Certifiable Distributional Robustness with Principled Adversarial Training. *stat* 1050 (2017), 29.

Congzheng Song and Vitaly Shmatikov. 2018. The Natural Auditor: How to Tell If Someone Used Your Words to Train Their Model. *arXiv preprint arXiv:1811.00513* (2018).

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. 2017. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. In *International Conference on Machine Learning*.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. 2018. Constructing Unrestricted Adversarial Examples with Generative Models. In *Advances in Neural Information Processing Systems*.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. 2017. Certified Defenses for Data Poisoning Attacks. In *Advances in Neural Information Processing Systems*.

Sining Sun, Ching-Feng Yeh, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie. 2018. Training Augmentation with Adversarial Examples for Robust Speech Recognition. *19th Annual Conference of the International Speech Communication Association* (2018).

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199* (2013).

Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In *40th International Conference on Software Engineering*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2019. Ensemble Adversarial Training: Attacks and Defenses. In *International Conference on Learning Representations*.

Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral Signatures in Backdoor Attacks. In *Advances in Neural Information Processing Systems*.

Aleksei Triastcyn and Boi Faltings. 2018. Generating Differentially Private Datasets Using GANs. *arXiv preprint arXiv:1803.03148* (2018).

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying Membership Inference Attacks in Machine Learning as a Service. *IEEE Transactions on Services Computing* (2019).

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2018. Clean-Label Backdoor Attacks. (2018).

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron Oord. 2018. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. In *International Conference on Machine Learning*.

Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. 2010. Fully Homomorphic Encryption over The Integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*.

Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy*. IEEE, 36–52.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. 2018. Efficient Formal Safety Analysis of Neural Networks. In *Advances in Neural Information Processing Systems*.

Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. 2008. Natural Evolution Strategies. In *IEEE Congress on Evolutionary Computation*.

Eric Wong and Zico Kolter. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *International Conference on Machine Learning*.

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. 2018. Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability. In *International Conference on Learning Representations*.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018b. Mitigating Adversarial Effects through Randomization. In *International Conference on Learning Representations*.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018a. Differentially Private Generative Adversarial Network. *arXiv preprint arXiv:1802.06739* (2018).

Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. *arXiv preprint arXiv:1704.01155* (2017).

Chen Yan, X Wenyuan, and Jianhao Liu. 2016. Can You Trust Autonomous Vehicles: Contactless Attacks Against Sensors of Self-Driving Vehicle. In *DEF CON 24 Hacking Conference*.

Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. 2017. Generative Poisoning Attack Method Against Neural Networks. *arXiv preprint arXiv:1703.01340* (2017).

Andrew Chi-Chih Yao. 1986. How to Generate and Exchange Secrets. In *27th Annual Symposium on Foundations of Computer Science*.

Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. 2019. Differentially Private Model Publishing for Deep Learning. In *Differentially Private Model Publishing for Deep Learning*.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. 2017. Efficient Defenses Against Adversarial Attacks. In *10th ACM Workshop on Artificial Intelligence and Security*.

Chao Zhang, Lei Zhang, and Jieping Ye. 2012. Generalization Bounds for Domain Adaptation. In *Advances in neural information processing systems*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. In *International Conference on Learning Representations*.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. In *International Conference on Machine Learning*.