SN Computer Science (2020) 1:253
https://doi.org/10.1007/s42979-020-00254-4

ORIGINAL RESEARCH

Security and Privacy Issues in Deep Learning: A Brief Review

Trung Ha[1,3] · **Tran Khanh Dang**[2,3] · **Hieu Le**[2,3] · **Tuan Anh Truong**[2,3]

Received: 30 April 2020 / Accepted: 15 July 2020 / Published online: 6 August 2020

Abstract
Nowadays, deep learning is becoming increasingly important in our daily life. The appearance of deep learning in many applications in life relates to prediction and classification such as self-driving, product recommendation, advertisements and healthcare. Therefore, if a deep learning model causes false predictions and misclassification, it can do great harm. This is basically a crucial issue in the deep learning model. In addition, deep learning models use large amounts of data in the training/learning phases, which contain sensitive information. Therefore, when deep learning models are used in real-world applications, it is required to protect the privacy information used in the model. In this article, we carry out a brief review of the threats and defenses methods on security issues for the deep learning models and the privacy of the data used in such models while maintaining their performance and accuracy. Finally, we discuss current challenges and future developments.

Keywords Security in deep learning · Privacy in deep learning · Differential privacy · Gradient descent · Threat · Defense

Introduction

Deep learning has many applications in life such as speech processing, biometric security, self-driving cars, health prediction, financial technology, and retail [1]. Each application has its own specific requirements depending on the nature of the data and the user's intent. The researchers proposed many models to meet the application requirements, users and characteristics of each type of application such as LeNet, VGG, GoogleNet, Inception, ResNet. However, major security-related weaknesses of the deep learning systems have recently been discovered and there have been a number of studies published on this issue. Although many researches have been published relevant to both attacking and protecting users' privacy and security techniques, they are still fragmented. Before Tramèr proposed the R-FGSM algorithm, he has reviewed some attack methods according to FGSM and GAN in [2]. In addition, security issues in the deep learning model are presented by Xiaoyong Yuan [3]. The above studies have only focused on the security of the deep learning model, which does not have an overview of protecting privacy in the deep learning model [4, 5]. This article reviews attack and prevention techniques in deep learning models, and specifically on adversarial examples. In user privacy, the article focuses on describing and classifying offensive and defensive techniques; in particular, differential privacy techniques in protecting privacy.

In deep learning model security, attack techniques are classified according to training and testing stages. This study focuses on threats at the testing. In addition, the classification is based on the attacker's knowledge and the pattern of attacking black boxes and white boxes. In protecting user privacy, attack techniques are classified based on system architecture and the attacker's knowledge. In system architecture, attack techniques are classified into two groups: centralized and distributed, while the attacker is also divided into white-box and black-box attacks according to the knowledge. Defensive techniques are classified based on the stages of the deep learning model.

This article is part of the topical collection "Software Technology and Its Enabling Computing Platforms" guest edited by Lam-Son Lê and Michel Toulouse.

✉ Tran Khanh Dang
khanh@hcmut.edu.vn

[1] University of Information Technology, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

[2] Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet street, District 10, Ho Chi Minh City, Vietnam

[3] Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

The content of the article is divided as follows: "Background" introduces the background of deep learning and differential privacy, next "Security in Deep Learning" describes the security in deep learning, "Privacy in Deep Learning" presents the privacy protection problem in deep learning, and the last section offers discussions and future directions.

Background

Deep learning models work in layers and a typical model at least has three layers which compose of input, hidden, and output layers. Each layer which connects each other accepts the information from previous and passes it on to the next one. Each layer has many neural networks which find associations between a set of inputs and outputs as illustrated in Fig. 1 and present as shown in Eq. (1):

$$Y = \sigma\left(\sum_{i=1}^{n} w_i x_i\right), \tag{1}$$

where n is total input, x is the ith input, w is the ith weights, which connects between input and output, σ is the activation function, and Y is the output.

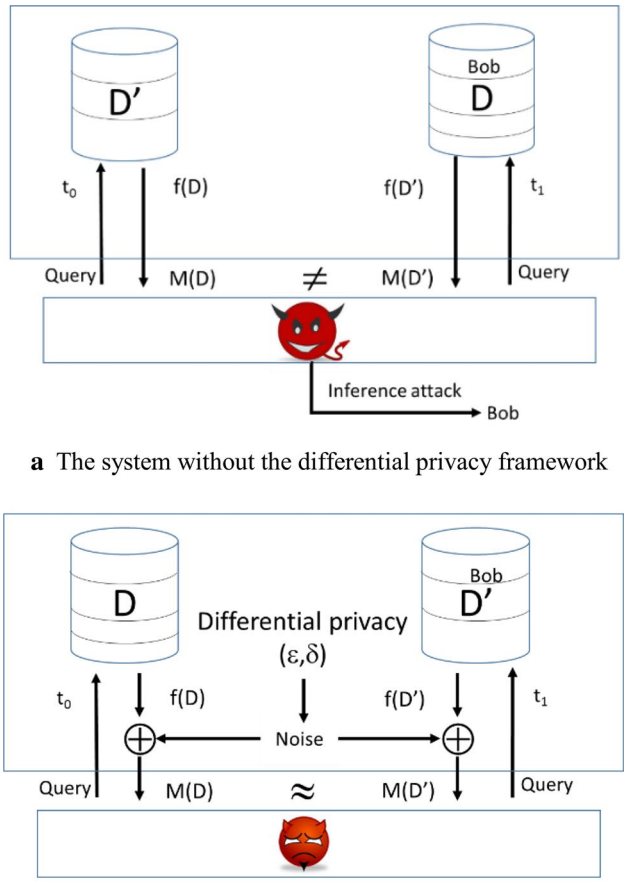
In the privacy in deep learning, an input is sent to the deep learning model, which responses an output. To reach reliable levels of accuracy, models require large datasets (datasets compose of unstructured and structured data) to learn. To shield individual privacy in this context, differential privacy method has been used [6, 7].

Definition differential privacy: A randomize mechanism M: $D \rightarrow R$ satisfies (ε, δ)-differential privacy if for any two adjacent inputs $d, d' \in D$ and for any subset of outputs $S \subseteq R$, it holds that:

$$\Pr[M(d) \in S] \leq e^{\varepsilon} \Pr\left[M\left(d'\right) \in S\right] + \delta, \tag{2}$$

where ε is the privacy budget that controls the privacy level, and δ allows for a small probability of failure.

The smaller ε and δ are determined, the more similar $M(d)$ and $M(d')$ are required to be as illustrated in Fig. 2.



a The system without the differential privacy framework

b The system with the differential privacy framework

Fig. 2 Overview of the differential privacy framework

In the security in deep learning, based on the scenarios, the assumptions are for deploying specific attacks. The threat models are divided based on the adversary's knowledge, attacker's target and the frequency of attacks.

The adversary's knowledge: A black box attack is a case when the attacker doesn't have much information about the system, in which case the attacker sends the input and receives the output without knowing the system parameters. In contrast, in the case of a white box attack, the

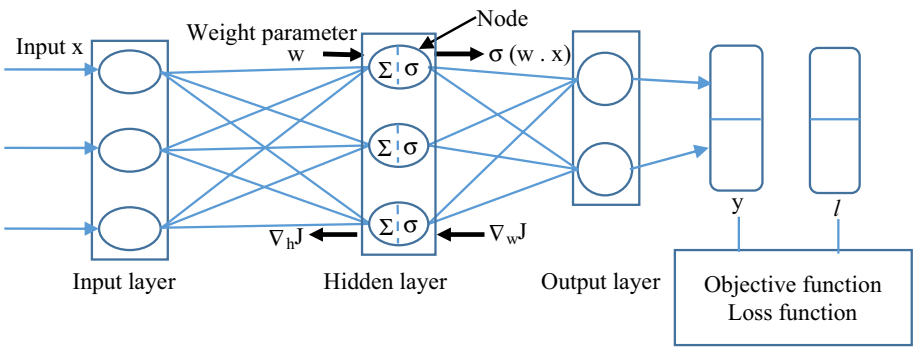


Fig. 1 General deep neural network training process [52]

attacker has access to all system information including the structure and parameter values of the model.

Attacker's target: Targeted attacks identify specific data or specific object classes that perform misclassification on this data set. These attacks often occur with classification systems. For example, in face recognition or authentication systems, an adversary chooses a specific face, which of adversarial examples is misclassified. In contrast, non-targeted attacks select arbitrary data and are easier to perform than targeted attacks.

Frequency of attacks: One-time attacks only take one time to create adversarial examples. Otherwise, iterative attacks perform multiple updates to generate adversarial examples. Iterative attacks always perform better than one-time attacks, but they require more queries to the deep learning system and take more times.

There are two types of security attacks in deep learning: adversarial and poisoning attacks. In this study, we focus on adversarial attacks. Adversarial attack adds noise to normal data during querying into the system. After receiving the returned results, the attacker uses this data to create adversarial examples. This attack is found in areas such as image processing, speech, and malware detection. Especially, in the field of image processing, it can deceive deep learning models, but not to humans. This noise value is the distance between the source data and the adversarial example. This value is measured by Minkowski distance as shown in Eq. (3). In addition, adversarial attacks can be classified according to the adversary's knowledge, attacker's target and the frequency of attacks.

$$L_p(x,y) = \left(\sum_{i=1}^{n} \left|x^i - y^i\right|^P\right)^{\frac{1}{P}}, \tag{3}$$

$$x = \left\{x^1, x^2, \ldots, x^n\right\},\ y = \left\{y^1, y^2, \ldots, y^n\right\},$$

where x is original data, x^i is attribute ith of original data, y is adversarial example, y^i is attributed ith of adversarial data.

Security in Deep Learning

The Threats in Deep Learning

A. **Threats**

Deep learning usually has two stages: training and prediction. An attack that uses adversarial examples and sends them to the system in the process of classification. After that, the system responses a misclassification with these inputs. For example, in an animal classification system, the input image, which contains a cat, sends to the system in the classification process but the system does not recognize it. This attack has occurred during predicting as shown in Fig. 3. On the other hand, the attacker has created adversarial examples to send during the training process to destroy the model, which causes the model to be misclassified.

B. **The Attack Model Prediction**

Attack the white box: Beginning with Szegedy's research, he proposed the idea of using the algorithm, which is the targeted attack and named "limited-memory Broyden–Fletcher–Goldfarb–Shanno" (L-BFGS) to generate an adversarial example [8]. The adversarial examples are done by making slight changes from the original image. Although the eyes see no difference between the changed image and the original image, the deep learnings see the differences between these two pictures. Interference problem is based on finding the value "r" optimized by searching linearly so that the value "r" satisfies $F(x+r)=l$ and using the box-constrained L-BFGS satisfies the formula:

$$\text{Min } c|r| + \text{loss}_f(x + r, l) \text{ subject to } x + r \in [0, 1]^m, \tag{4}$$

where x is the set of inputs, l is the set of outputs, r is the perturbation. In addition, the optimal "r" in the L-BFGS attack is also calculated by the binary search method [9].

Szegedy's attack is defeated by the defensive distillation method that is weaker than the Jacobian-based Saliency Map

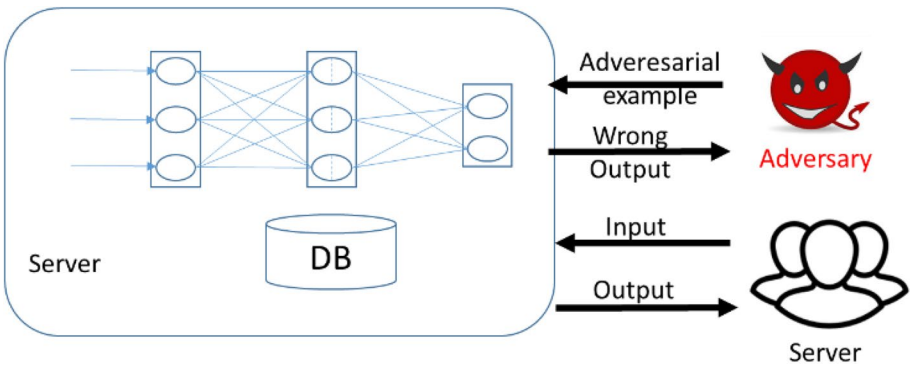


Fig. 3 Deep learning system and the attack model prediction

Attack (JSMA) [10, 11], which has the main algorithm's idea based on the L_0 distance optimization method. This Szegedy's attack is based on a greedy algorithm that chooses each pixel to change at each time. It uses gradient F, which impacts to each pixel and the results classification, to calculate a Saliency Map. The larger this value of the pixel is in the map, the greater the probability of attacked network will be.

In addition to JSMA attacks based on function optimization, there is another attack method, called "Carlini and Wagner attack" (C and W attack) [12]. This attack is based on the L-BFGS attack but there are three major differences.

The first difference is the optimal formula g definition:

$$\text{Min } D(x, x + r) + \text{c.f}(x + r) \text{ such that } x + r \in [0, 1]^n, \tag{5}$$

where D is a distance metric that includes L_0, L_2, and L_∞, $f(x)$ is an objective function in which $f(x') = l'$ if and only if $g(x) \leq 0$ and $c > 0$ is a properly chosen constant. This modification enables Eq. (5) to be solved by the existing optimization. This attack is defeated the example adversarial prevention methods.

Secondly, instead of using box-constraint like L-BFGS to find a minimal disturbance in L-BFGS attack, C and W attack uses w parameter instead of box-constraint with w satisfying $r = \frac{1}{2}(\tan h\,(w) + l) - x$.

Finally, the C&W attack gives three measurement parameters as compared to L-BFGS method. These three measurements show three different attacks: L_0 attack, L_2 attack, and L_∞ attack. The defensive distillation prevention method is defeated by L_2 attack, the distance L_2 attack is calculated as the formula:

$$\min_w \left\| \frac{1}{2}(\tan h(w) + l) \right\|_2 + \text{c. f}\left(\frac{1}{2}(\tan h(w) + l) \right). \tag{6}$$

In other white-box attack, Deepfool method is a simple and accurate method to fool deep neural networks that is other optimal attack solution proposed by Moosavi-Dezfooli in 2016 [13]. When Deepfool algorithm compares to L-FBGS algorithm with the same level of jamming, the execution time of L-FBGS algorithm is much slower. Deepfool method does noises in two cases of binary and multi-layered classifications. This method is used to find the closest distance from the original input to the decision boundary of the adversarial examples. To overcome the non-linearity at the height, they performed a repeat attack with a linear approximation. In this method, Deepfool added less noise to the original data than L-FBGS method.

To continue developing from the Deepfool algorithm, Moosavi-Dezfooli launched a universal adversarial perturbations (UAP) attack [14]. To implement this attack, the author offers a formula to find satisfactory vector universal perturbation:

$$\|r\|_p < \varepsilon, \tag{7}$$

$$P\left(x' \neq f(x)\right) \geq 1 - \delta,$$

where ε limits the size of universal perturbation, and δ controls the failure rate of all the adversarial samples.

Data set X is a sample image set. The UAP algorithm looks for a universal perturbation until most of the X data sets are fooled. For each iteration, the author used the deepfool method to get minimum noise sample for each input and update the noises to the total noises. This loop will not stop until most of the data samples are fooled ($P < (1 - \delta)$). From the experiments in the paper, universal perturbation can be generated using a small fraction of data samples instead of the total dataset [14].

The other method is called fast gradient sign method (FGSM), which is the first algorithm to use gradient inputs to create adversarial examples [15]. In this algorithm, the direction in each pixel is determined by the computed slope using the backward propagation method. Their perturbation can be expressed as:

$$r = \varepsilon \cdot \text{sign}\left(\nabla_x J_\theta(x, l)\right), \tag{8}$$

where ε is the magnitude of the perturbation. The generated adversarial example x' is calculated as $x' = x + r$, and l_x is the true label of x.

In addition, Kurakin improved FGSM method. If the FGSM method only performs one time, the I-FGSM algorithm attacks multiple times [16]. The author changed the step of new inputs repeatedly as a formula:

$$x_0 = x, \tag{9}$$

$$x_{i+1} = \text{Clip}_{x,\varepsilon}\left\{x_i + \alpha \cdot \text{sign}\left(\nabla_x \text{Loss}\left(x_i, l_x\right)\right)\right\},$$

where l_x is the true label of x, $\text{Clip}_{x,\varepsilon}\left\{x'\right\}$ the function performs clipping on image per-pixel.

Besides that, Yinpeng Dong improved the I-FGSM algorithm by adding momentum [17]. Momentum is used to step out the local maximum optimal and iterations are used to achieve optimized stability level.

White-box attack has the gradient parameter to generate the adversarial example, but the black-box attack method does not have this parameter. When attacking by black-box method, it has to build a deep learning system to create the adversarial example. To have data to build a deep learning system, an attacker performs the aggregated data by performing multiple queries on the deep learning system. Then, it uses the aggregated data to build a deep learning model. This model is used to generate adversarial examples that are to attack the target model [18, 19].

Defense Techniques in Deep Learning

A. **Adversarial Training**

The idea of the method is that the source data and the adversarial examples are involved in the model training time. The use of adversarial examples in training time makes the deep learning system more accurate and reliable. Goodfellow evaluated this method on the MNIST dataset [15]. During the training time, at each step, they used a half of the original data and a half of adversarial examples. Experimental results showed that they were resistant to adversarial examples of once-step examples such as FGSM, but this method was not effective for iterative attacks like I-FGSM. Kurakin also pointed out the use of adversarial examples in training time to increase the accuracy and reliability of building deep learning models with small data sets such as the MNIST dataset [16].

B. **Detecting Adversarial**

This method solved the problem based on the idea of binary classification. It means that when giving an input into the system, the detector will classify that this input is the original input or adversarial example. There are several suggested ways to solve this problem. Metzen created an extraneural network that detected adversarial examples. This sub-neural network task performed binary classification [20], while Lu proposed the Safetynet system to add a binary classification threshold on nodes in the deep learning network to detect adversarial examples [21].

Besides that, Hendrycks provided a way to differentiate clean and original data based on the output effect of a neural network. The main method builds that PCA coefficients of adversarial examples are larger and higher variance for high-frequency components. Inputs are classified as the clean and adversarial examples by fitting two Gaussians to use in a likelihood comparison, one for clean examples and another for adversarial example [22].

Moreover, Song proposed to use a generative model of images to detect and defend against adversarial examples [23]. This model defines the joint distribution over all pixels by factoring it into a product of conditional distributions (p values). The author used p-values as a measure to detect noise in the input data as shown in Eq. (10). The empirical results are given that this approach can detect adversarial examples generated by FGSM, I-FGSM, Deepfool and C&W attack.

$$p_{\mathrm{CNN}}(X) = \prod_i p_{\mathrm{CNN}}\left(x_i \middle| x_{1:(i-1)}\right). \tag{10}$$

Table 1 The classification attack: black-box and white-box

Attack	Black-box	White-box
Steal the model	✓	
Reconstruction attack	✓	
Inference membership attack	✓	
Steal the sensitive information of the user		✓

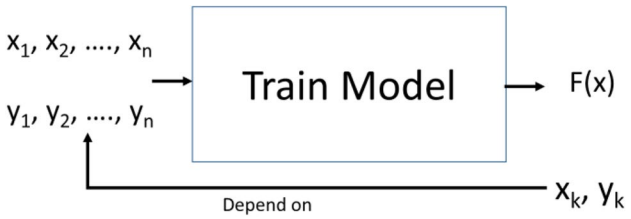


Fig. 4 Inference attack general [53]

Privacy in Deep Learning

The Threats in Deep Learning

There are two types of leaking personal information in deep learning: inference attacks as shown in Fig. 4 and system organization as shown in Fig. 8. These types have four attack methods as shown in Table 1.

A. **Inference Attack**

Inference attacks in deep learning fall into two fundamental categories, including tracing (membership inference) attacks and reconstruction attacks [24].

In the reconstruction attacks category as illustrated in Fig. 5, the attacker's objective is to extract training data from outputted model predictions. According to Fredrikson's experiment, the constructed model inversion attacks for deep models use the output of the model to infer certain features of the training set [25]. Especially, in facial recognition, Fredrikson's research has shown that training data can be reconstructed from the model [26]. It means that the principle behind model inversion uses features synthesized from the model to generate an input that maximizes the likelihood of being predicted with a certain label. Furthermore, the adversary's objective is to train a substitute model F' that is capable of mimicking a target model F [27]. To build model F', it is based on the leakage of information that is implemented in the extraction time. In model extraction, the adversary only has to access the prediction API of a target model and query the target model iterative using "natural" or synthetic samples. These samples are specifically

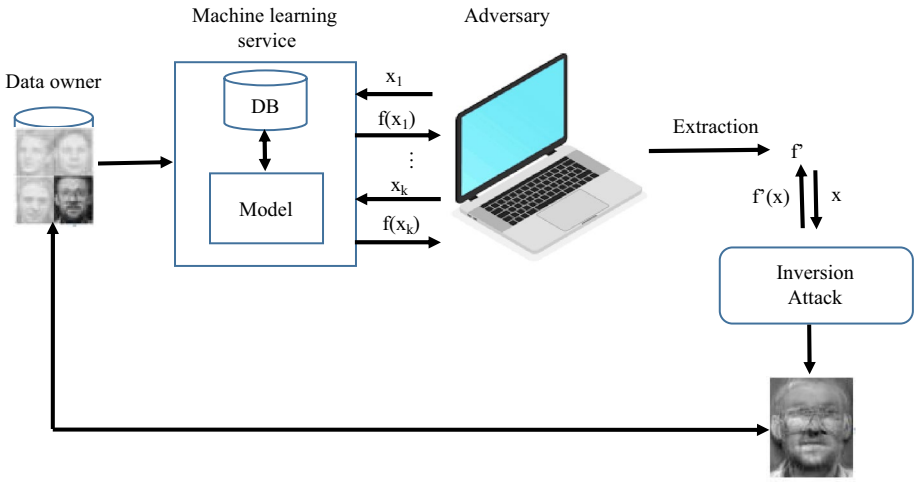


Fig. 5 Reconstruction attack [54]

crafted to maximize the extraction of information about the model internals from the predictions returned by the model *F*.

In tracing attacks category, an adversary, who identifies input and output formats, is given black-box access to a target model without knowing its internal parameters and wants to infer whether a particular record is included in the training set [28]. The authors transformed the membership inference attack into a classification task [24]. There are three steps to implement on this attack, such as queries, data collection and shadow models building. In the first step, the adversary queries the target record *t* and uses the target classifiers' predictions on *t* to infer the membership status of *t*. For each record *t*, there are two possible classes: class label "in", which means that the record is in the training set, and class label "out", which represents that the record is not in the training set. For the next step, the "shadow" training technique is built to use for the membership classification task. Multiple "shadow models" that are trained by the adversary use the same machine learning algorithm on records sampled from the data in the first step. These shadow models are used to simulate the behavior of the target model and generate a set of training records with labeled membership information. Specifically, the adversary queries each shadow model with two sets of records, including the training set of the shadow model and a disjoint test set. For each record, a new feature vector is generated by concatenating the record's original attributes with the shadow classifier's predictions on that record. A new class label is created to reflect membership, i.e., "in" for records in the training set, "out" for records in the test set. In the final step, after using the labeled dataset, the adversary trains a model as "attack" classifier and uses it to infer the membership of a target record *t* as shown in Fig. 6.

B. System Organization

In the system organization, there are two privacy threats in deep learning architecture: central and collaborative learning system. For the central system, after

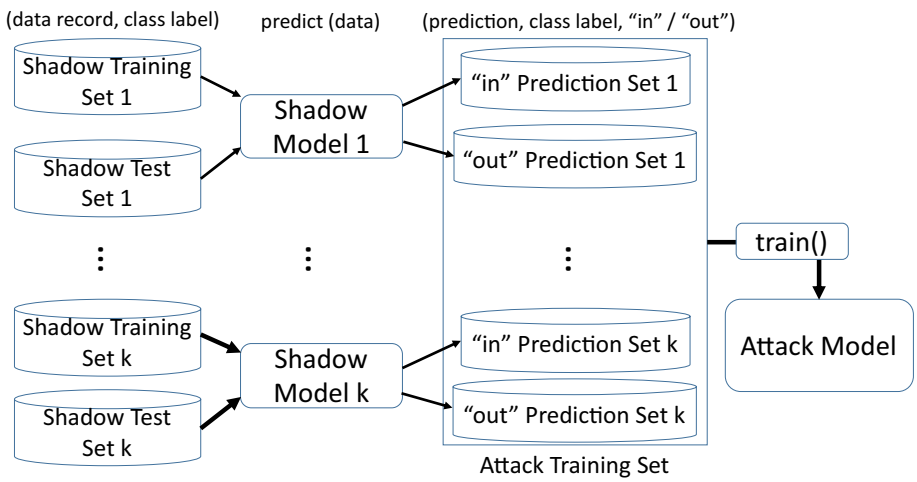


Fig. 6 Tracing attack [27]

companies provide deep learning models and services to the public such as machine learning as a service like Microsoft Azure Learning, Google, Amazon, BigML, etc. [29]. Data owner sends data to the server and public model service to the client. The adversary sends an input to public machine learning service and receives an output. Using the inputs and outputs pair, the attacker can train his own local model which is similar to the target model as illustrated in Fig. 7 [30].

In a collaborative learning system, clients train a batch locally and then calculate the gradient that is applied to its weights to minimize the cost function. Finally, it sends the gradient to the parameter server. If the adversary has taken the role of model server, they receive the client's gradient. This is called "model steal attack". In addition, the adversary has taken the role of participants that attack to steal other participant information from the training set. This attack is based on exploiting the real-time nature of model learning, which allows the adversary to train a GAN that generates prototypical samples of the private training set as illustrated in Fig. 8 [31].

The Defenses by Differential Privacy in Deep Learning

By applying differential privacy to the deep learning models, the training data can be protected from the inversion attacks or inference attacks when the model parameters are released. There are many researches that utilize differential privacy to deep learning models. Such methods assume that the training datasets and parameters of the model are the database and prove that their algorithms satisfy Eq. (2). Depending on where the noise is added as illustrated in Fig. 9, such approaches can be divided into three groups: gradient-level, function-level, and label-level (Table 2).

A. **Gradient-Levels**

The gradient level approach, in which the client adds noise into the gradients of the parameters before sending to server, solves the issue in the collaborative learning as illustrated in Fig. 10 [32]. From the beginning of Shokri's proposal, instead of sending the entire data sets to the server, the clients can train data sets to create a model of the system. The client then sends the model parameters to the server, and this server collects these parameters and finds the optimal parameter. However, in this way, the server can rely on the model's parameters to infer the trained set of the clien's data set. Therefore,

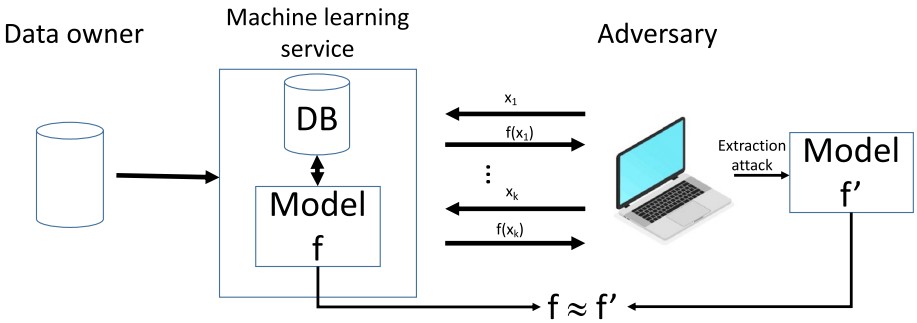


Fig. 7 The model extract attack in the central learning system

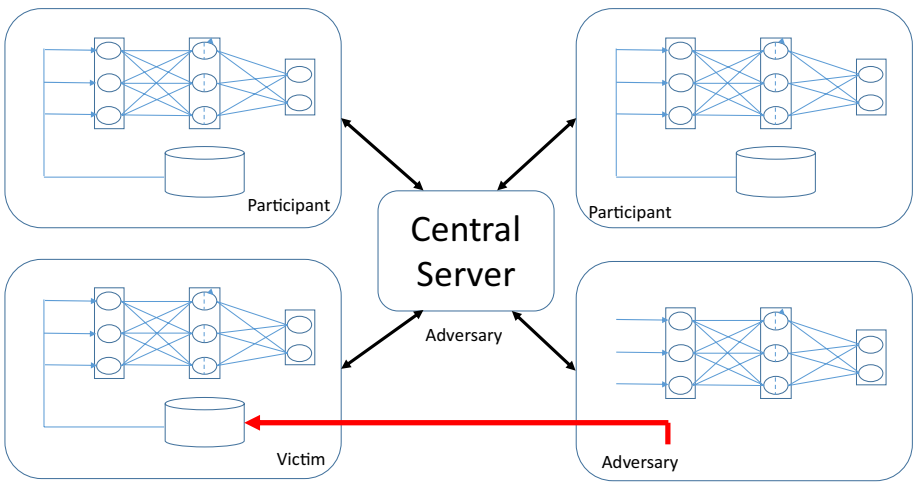


Fig. 8 The threat in collaborative learning system

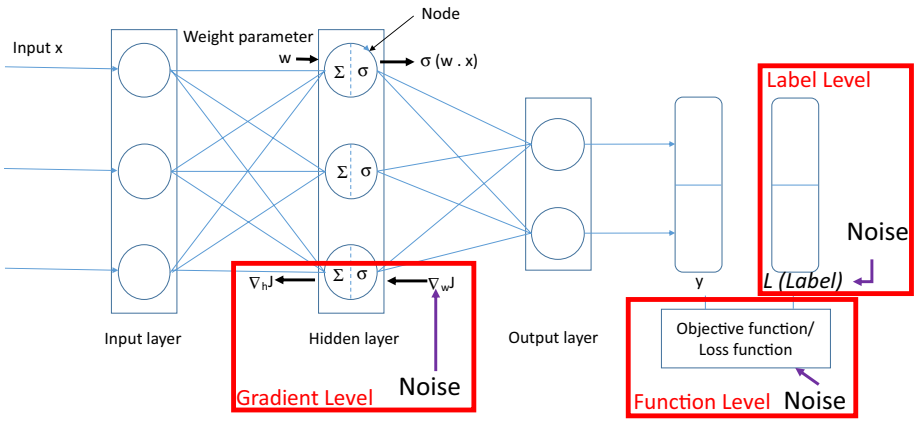


Fig. 9 Protect privacy in the deep learning model

Table 2 The defense approach in the learning system

Defense approach	Algorithm	Centralized learning system	Collaborative learning system
Gradient-level	DP-SDG [33]		✓
	DPGAN [40]		✓
	DPGM [35]		✓
	DP model publishing [41]		✓
Function-level	dPA [55]	✓	✓
	dCDBN [43]	✓	✓
	AdLM [31]	✓	✓
Label-level	PATE [45]	✓	✓
	Scale private learning [46]	✓	✓

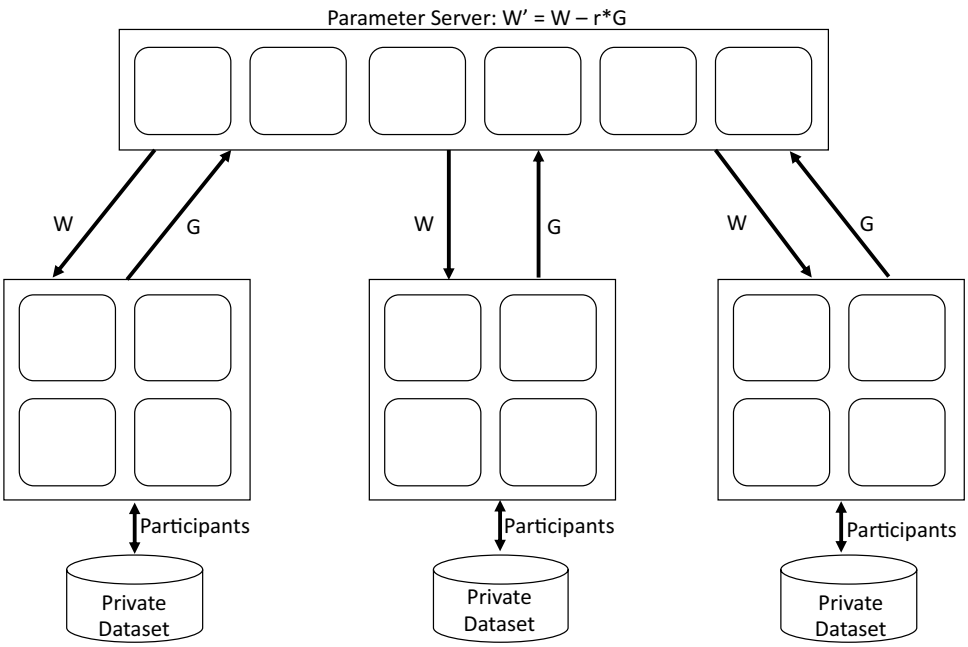


Fig. 10 The collaborator learning architecture with Differential Private Stochastic Gradient Descent. *W* represents the parameter and *G* represents the gradient information

Abadi proposed improving the Differentially Private stochastic gradient descent (DPSGD) algorithm by protecting the privacy of the gradients [33]. This method is usually applied to white-box attacks, when the attacker has information on system architecture such as gradients, patterns, etc. Abadi also suggested the tracing to see when there is a loss of privacy. Limiting the disclosure of privacy, the noise method was used by the Gaussian mechanism. The author proposed the calculating moments method to track the loss of privacy as formula Eq. (11). He also pointed out the hyper-parameter tuning parameter related to the balance of privacy, accuracy and performance.

$$c\left(o;M,\text{aux},d,d'\right) \triangleq \log \frac{\Pr\left[M(\text{aux},d)=o\right]}{\Pr\left[M(\text{aux},d')=o\right]}, \tag{11}$$

with neighboring databases $d, d' \in D_n$, a mechanism M, auxiliary input aux, and an outcome $o \in R$, define the privacy loss at o

The new Differentially Private SGD algorithm is evaluated on small data sets such as MNIST, CIFAR-10. There are many questions having to be solved such as the privacy protection for large data sets or the privacy problem solving in the long short-term memory (LSTM) architecture when applying this algorithm. In 2018, McMahan continued to assess the privacy level of the DPSGD model on LSTM architecture [34].

Acs proposed the algorithm of the differentially private generative model (DPGM) to improve Differentially Private SGD (DP-SGD) [35]. If DP-SGD randomly selects T data set as training data, DPGM chooses T data sets on the same layers that are divided by the k-mean algorithm for the original data set [36, 37]. These will be transferred to training models such as restricted boltzmann machine (RBM) and variational auto-encoder (VAE) [38, 39]. But before using the k-mean algorithm to divide the original data set into k layers, the author proposed using Fourier series transforms to reduce the number of projections. Acs is just like Abadi just testing the algorithmic model on a small data set of MINST, not using a large data set model.

To solve the problem of privacy protection for small data sets, Xie proposed the differentially private generative adversarial network (DPGAN) framework [40]. The author made adding noises to the parameters during the training process, which were different from the previous algorithms that added noise after training the model. The author also empirically pointed out the relationship between privacy and output of deep learning models that were related to the parameter∈. The smaller the output blurs, the higher the privacy is. The smaller the ∈is, the blurrier the output is and the higher privacy is.

If Abadi used a fixed number of molecules in the batch to calculate the privacy level during the training, then Yu recommended taking a different number of samples for each iteration of DPSGD for the implementation of dynamic privacy. In addition, the author also introduced a new concept of concentrated differential privacy (CDP), this concept was suitable for systems that needed to perform a large number of calculations to train the model [41].

B. **Function-Levels**

There are many proposed issues related to the objective function. For example, the differentially private logistic regression's parameters of Monteleoni are trained based on the perturbed objective function [30]. Besides that, in 2016, Phan implemented private auto-encoder (PA) as depicted in Fig. 11 based on three main ideas. Firstly, the cross-entropy error functions of the data reconstruction and soft-max layer were converted to polynomials by implementing the Taylor Expansion. Secondly, the author added noises to polynomial functions to meet ε-differential privacy during the training process. Finally, he added a step of normalization layer on top of the hidden layer to protect the ε-differential privacy when the system uses many auto-encoders, which is called deep private auto-encoder (dPA).

In addition to the variational auto-encoder, there is other generated model which is called convolutional deep belief networks (CDBN) [42]. In 2017, Phan et. al. proposed a framework of differential privacy in convolutional deep belief networks (pCDBN) [43]. The pCDBN framework has the same idea as dPA is to add Laplace noise into the activation functions but there are a few different ideas. The first different idea is that pCDBN protects privacy for convolutional deep belief networks, while dPA protects the auto-encoder model. Next, dPA uses Taylor Expansion to approximate the cross-entropy

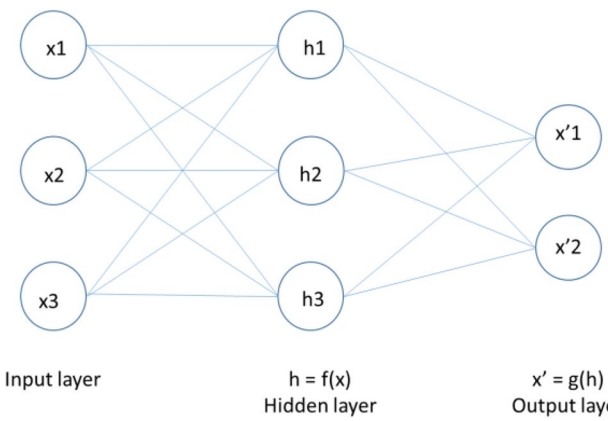


Fig. 11 Simple schema of a basic auto-encoder

error functions, while pCDBN uses Chebyshev expansion.

However, dPA only meets the function-level privacy protection for specific cases, which are the deep autoencoder, and the pCDBN for model convolutional deep belief network. Phan has developed a novel mechanism, called adaptive laplace mechanism (AdLM), to preserve differential privacy in deep learning [44]. The main idea of the algorithm is to add more "noises" and "less relevant" features to the model's output, and vice versa. The author used Laplace noise to calculate the layer-wise relevance propagation (LRP) to estimate the level of privacy and the relationship between each input feature and the model's output [29].

C. **Label-Levels**

Differently from the gradient and function levels, the label-level approach injects noise into the knowledge transfer phase of the teacher-student framework as depicted in Fig. 12. For the label-level, it is suggested that the semi-supervised knowledge transfer model—the Private Aggregation of Teacher Ensembles (PATE) mechanism by Papernot proposed [45]. PATE is a type of teacher-student model, and its purpose is to train a differentially private classifier (student) based on an ensemble of non-private classifiers (teacher). Moreover, the moment accountant is utilized to trace the cumulated privacy budget in the learning process by PATE and PATE also ensures safety intuitively and in terms of the DP, respectively.

Later, the PATE was extended to operate on a large-scaled environment by introducing a new noisy aggregation mechanism by Papernot [46]. It is shown that the improved PATE outperforms the original PATE on all measures and has high utility with a low privacy budget in the large dataset such as street view house numbers (SVHN). Furthermore, Triastcyn and Faltings applied the PATE to build the differential private GAN framework [47]. Using PATE as a discriminator of GAN frameworks that a type of classifier determines whether the input data is real or fake, the generator trained with the discriminator is also differential private.

In the summary, we show the main ideas, advantages and disadvantages of the defenses to compare, synthesize and help the later researches for identifying the issues to solve (Table 3).

Discussion and Future Works

During the research, development and operation, organizers should promulgate laws and regulations against privacy violations that include the following article such as building tools that allow users to monitor user privacy when providing data to deep learning systems. In addition, deep learning systems are often based on deep learning frameworks such as Tensorflow, Torch, Caffe, and Opencv… The vulnerabilities in these frameworks also affect the deep learning system built on these frameworks. From the vulnerabilities, an adversary attacks a black or white box into a deep learning system. Black box attacks rely on a lot of system queries to get a large amount of data from which to build substitute models. But deep learning systems now limit the number of queries into the system, and Tramér's black box attacks require the knowledge of the architecture of the attack system [27]. Moreover, deep learning systems have additional parts to prevent attacks and these additional parameters are not disclosed, so it is difficult for attackers to implement. In addition, a thorough understanding of deep learning systems is an urgent issue today. Changing the input or changing the function in a node in a neural network layer can lead to false results. The problem is that you need to understand the operating model, and how each node operates in each deep neural network to provide effective prevention methods as well as ways to find vulnerabilities for the attack of the system. In the adversarial example prevention techniques, the study focuses on two main groups of solutions: adversarial training and detect adversarial. The idea of the adversarial

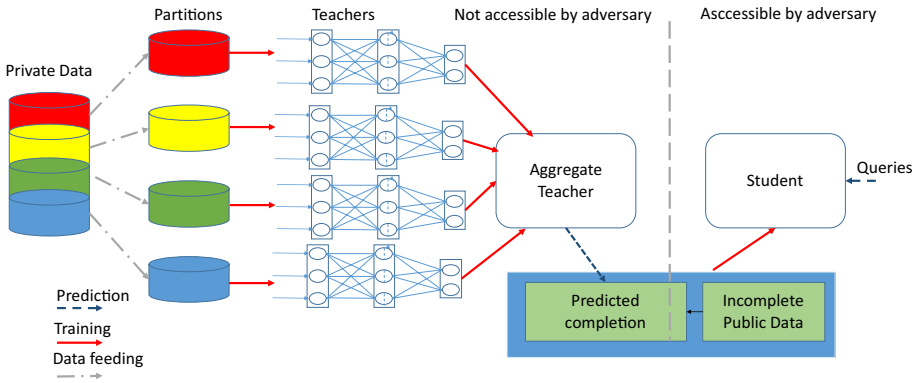


Fig. 12 The Private Aggregation of Teacher Ensembles (PATE) mechanism

Table 3 Summary of privacy protection algorithms: idea, advantage, disadvantage

Defense approach	Algorithm	Main ideas	Advantages	Disadvantages
Gradient-level	DP-SDG [33]	To develop new algorithmic techniques for learning and a refined analysis of privacy costs within the framework of differential privacy	To track privacy loss To automate analysis of the privacy loss To analyze of privacy costs	Do not apply for more complex deep learning models
	DPGAN [40]	To add designed noise to gradients during the learning procedure for the differential privacy in GANs	To solve the issue in GANs	To inapplicate to more complex datasets without resorting to unrealistic assumptions, like access to public data from the same distribution
	DPGM [35]	To show a technique for privately releasing generative models To model the generator distribution of the training data with a mixture of k generative neural networks	To perturb the objective functions of these autoencoders	To apply specific learning model
Function-level	dPA [55]	Related to enforce ε-differential privacy To perturb the objective functions of the traditional deep auto-encoder	Do not depend on the number of training epochs in consuming privacy budget To apply for non-linear activation functions	To use only the objective functions—finite polynomials
	dCDBN [43]	Related to enforcing ε-differential privacy To leverage the functional mechanism to perturb the energy-based objective functions of traditional CDBNs	To add more noise into input features which are less relevant to the model output Do not depend on the number of training step in the privacy budget consumption To apply different activation functions	To diminish the model's accuracy for complex tasks
	AdLM [31]	To propose a novel mechanism, called Adaptive Laplace Mechanism (AdLM), to preserve differential privacy in deep learning	To add more noise into input features which are less relevant to the model output Do not depend on the number of training step in the privacy budget consumption To apply different activation functions	
Label-level	PATE [45]	To provide strong privacy guarantees for training data: Private Aggregation of Teacher Ensembles (PATE)	Do not depend on the learning algorithm	To obligate trust teacher model

training method is to use adversarial example as part of the training data set. Adversarial example is created from other machine learning models. The idea of the detect adversarial method is to use the deep learning system to distinguish whether a record is an adversarial example or not.

The differential privacy is a method to prevent member inference attacks. The main idea of this method is to satisfy the highest accuracy while minimizing the ability to identify a specific record when querying from a statistical database. To protect user privacy, it has to remove individual features. There are three ways to eliminate individual features such as gradient, loss function, and label. In addition to the differential privacy method, there is a homomorphic encryption method, which is based on coding individual features and then sending them to the server to perform the training. Instead of coding individual features, we propose an idea to prevent the disclosure of individual feature information by obscuring it before sending it back to the server. This method is often used in the secure multi-party computation. We also propose a data privacy-preserving framework to receive raw data, process it to preserve data privacy then send the processed data to the machine learning services. The framework transforms the original data into another form that is still utilizable for learning models but has less privacy risk. Instead of sending raw data directly to machine learning services, data owners send them along with some anonymization policies to the proposed framework. Those policies are prepared with the help of data experts, based on the requirements from the current domain, learning models and the level of privacy preservation. As a result, the framework can provide different datasets to the learning services from on raw data and depend on different policies provided from data owners (Fig. 13).

With the raw data, the data owners prepare the type identification metadata of each field. The framework also supports them in making some risk identification before running the anonymization processes. Then, we evaluate the outcome data to measure the risk as well as the utility of the anonymized dataset. Finally, after some iterations of the anonymization, we provide the results that have an acceptable level of utilization and satisfies privacy requirements to the learning model. Figure 14 depicts the primary components of the framework.

In the type identification and risk identification stage, the data experts identify the type (structured, semi-structured, and unstructured) of the dataset as well as the domain (environmental data, population data…). Next, they categorize data attributes into groups: identifying attributes, quasi-identifying attributes, sensitive attributes, and insensitive attributes. They also provide metadata about masking data to the framework to run the anonymization algorithms.

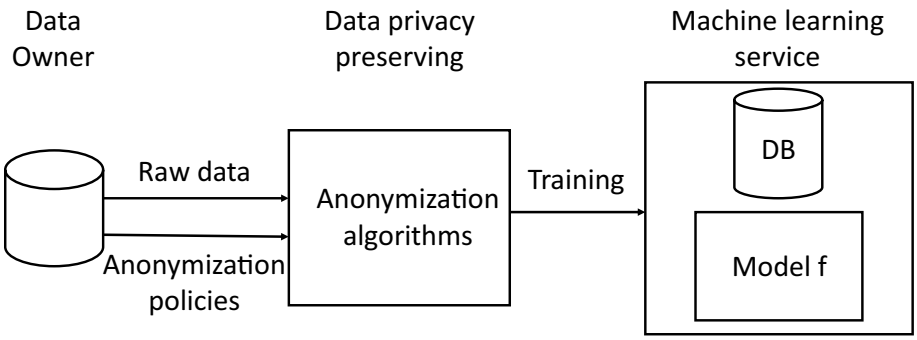


Fig. 13 Add a data privacy-preserving platform to the machine learning processes

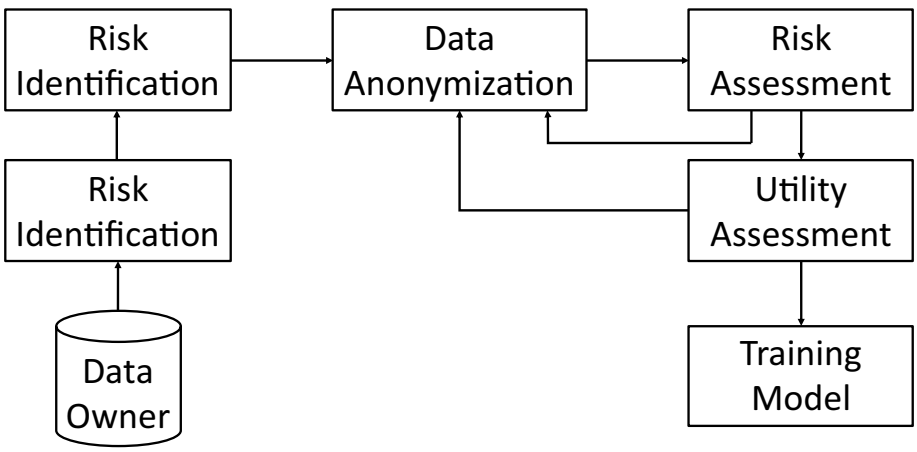


Fig. 14 The data anonymization framework between data owners and training models

Based on the provided policies, the framework executes different algorithms and models (such as *k*-anonymity, *l*-diversity, *t*-closeness) to the dataset and measure the risk as well as the utility of the outputs [48, 49]. This process is iterated multiple times, using heuristics to find the optimal result. We can apply some times and resources constraints to the framework to stop the iterations earlier. The data experts see not only the chosen optimal solution from the framework but also the whole solution space in case they want to use a different output.

With support from such a data preserving framework, data owners can provide materials to the learning models in a better way for data privacy. The framework fully supports data experts in configuring the anonymization policy and evaluating the results. Based on the training models and the privacy requirements, the experts work on the original dataset to create different strategies. Those include the types of configurations such as algorithms, parameters of each algorithm, quasi-identifying attributes and sensitive attributes. Because the anonymized data is used in different scenarios, the framework also allows data experts to handle different utility measurements to evaluate the outcome of the process. Besides, they can choose a different solution from the solution space if they are not satisfied with the proposed one from the framework.

Attack and defense methods in deep learning always remain to help the deep learning system better. Based on the characteristics of the deep learning model, it is possible to find new ways to attack deep learning to destroy a model or to steal a model as well as deduce member training of the training data set. In adversary attack, L_p distance is often used to measure the level of perturbations and L_p distance uses a deception deep learning system, but not the human eye. There is a question for using any measurement that can be used to deceive the deep learning system and also to deceive humans. Besides, most of the black-box attacks use a large number of queries, there is an attack that uses a small number of queries but can still effectively attack the deep learning system or not. For the defense, to prevent destructive attacks, the defender needs to understand how each node operates and errors through each layer and select the appropriate activation function, as well as deep learning systems that monitor unusual queries.

Conclusion and Future Work

Deep learning makes people's lives more comfortable and the security and privacy of deep learning become an issue not to be overlooked. Therefore, we have reviewed attack and defense methods in a deep learning model.

In the security of the deep learning model, we review the offensive and defensive techniques during the test stage that is the evasion attack. Depends on the attacker's knowledge of the deep learning system, we classify the attack scenario into two categories: white-box and black-box. Most of these attacks are based on creating adversarial examples and the distance measure L_p. But these methods of attack only deceive the deep learning system, but not to humans. In the future, any kind of attacks can fool not only people but also the system. Moreover, attack and defense methods are closely related that help the deep learning system to be less errors during execution.

Since 2013, the world has entered the fourth industrial revolution (4IR), data is considered a valuable resource. Thus, the issue of protecting privacy in deep learning systems is extremely important. In this study, we also describe privacy threats in deep learning models and point out the points in the deep learning model implemented to protect privacy. There are three main methods: gradient-level, function-level, label-level, which are based on the differential privacy theory. Currently, the privacy group issue is getting more attention, the methods based on differential theory satisfy the privacy group. More study on deep learning-based approaches in the context of security and privacy issues in smart cities applications [1, 50, 51] to indentify security breaches in the internet of things or e-commerce systems is also of our great interest in the future.

Acknowledgements This work is supported by a project with the Department of Science and Technology, Ho Chi Minh City, Vietnam (contract with HCMUT No. 08/2018/HĐ-QKHCN, dated 16/11/2018).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Dang TK, Pham CDM, Ho DD. On verifying the authenticity of e-commercial crawling data by a semi-crosschecking method. Int J Web Inf Syst. 2019;15(4):454–73. https://doi.org/10.1108/IJWIS-10-2018-0075.
2. Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: attacks and defenses. In: Proceedings of the 6th international conference on learning representations, Vancouver, BC, Canada, 30 Apr–3 May 2018.
3. Yuan X, Pan H, Qile Z, Xiaolin L. Adversarial examples: attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst. 2019;30(9):2805–24.
4. Ji Z, Lipton ZC, Elkan C. Differential privacy and machine learning: a survey and review. arXiv preprint arXiv:1412.7584. 2014. https://arxiv.org/abs/1412.7584. Accessed 15 Apr 2020.
5. Zhang D, Chen X, Wang D, Shi J. A survey on collaborative deep learning and privacy-preserving. In: Proceedings of the 3rd IEEE international conference on data science in cyberspace, Guangzhou, China, 18–21 June 2018. pp. 652–658.
6. Dwork C. Differential privacy: a survey of results. In: International conference on theory and applications of models of computation.

Lecture notes in computer science, vol. 4978. Berlin: Springer; 2008. pp. 1–19.
7. Bun M, Steinke T. Concentrated differential privacy: simplifications, extensions, and lower bounds. In: Theory of cryptography conference. Lecture notes in computer science, vol 9985. Berlin: Springer; 2016. pp. 635–658.
8. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. In: arXiv preprint arXiv:1312.6199. 2013. https://arxiv.org/abs/1312.6199. Accessed 15 Apr 2020.
9. Tabacof P, Valle E. Exploring the space of adversarial images. In: International joint conference on neural networks, Vancouver, BC, Canada, 24–29 July 2016. pp. 426–433.
10. Papernot N, McDaniel P, Wu X, Jha S, Swami A. Distillation as a defense to adversarial perturbations against deep neural networks. In: Proceedings of the 37th IEEE symposium on security and privacy, San Jose, CA, USA, 22–26 May 2016. pp. 582–597.
11. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: IEEE European symposium on security and privacy (EuroS&P), Saarbrücken, Germany, 21–24 Mar 2016. pp. 372–387.
12. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proceedings of the 38th IEEE symposium on security and privacy, San Jose, CA, USA, 22–26 May 2017. pp. 39–57.
13. Moosavi-Dezfooli SM, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016. pp. 2574–2582.
14. Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P. Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017. pp. 86–94.
15. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. 2014. https://arxiv.org/abs/1412.6572. Accessed 15 Apr 2020.
16. Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 Apr 2017.
17. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18–22 June 2018. pp. 9185–9193.
18. Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 Apr 2017.
19. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. In: Proceedings of the ACM on Asia conference on computer and communications security, Abu Dhabi, United Arab Emirates, 2–6 Apr 2017. pp. 506–519.
20. Metzen JH, Genewein T, Fischer V, Bischoff B. On detecting adversarial perturbations. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 Apr 2017.
21. Lu J, Issaranon T, Forsyth D. Safetynet: detecting and rejecting adversarial examples robustly. In: Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 Oct 2017. pp. 446–454.
22. Hendrycks D, Gimpel K. Early methods for detecting adversarial images. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 Apr 2017.
23. Song Y, Kim T, Nowozin S, Ermon S, Kushman N. Pixeldefend: leveraging generative models to understand and defend against adversarial examples. In: Proceedings of the 6th international conference on learning representations, Vancouver, BC, Canada, 30 Apr–3 May 2018.
24. Nasr M, Shokri R, Houmansadr A. Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: Proceedings of the 40th IEEE symposium on security and privacy, San Francisco, CA, USA, 19–23 May 2019. pp. 739–753.
25. Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: 23rd USENIX security symposium, San Diego, CA, USA, 20–22 Aug 2014. pp. 17–32.
26. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, Denver, CO, USA, 12–16 Oct 2015. pp. 1322–1333.
27. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T. Stealing machine learning models via prediction apis. In: 25th USENIX security symposium, Austin, TX, USA, 10–12 Aug 2016. pp. 601–618.
28. Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Proceedings of the 38th IEEE symposium on security and privacy, San Jose, CA, USA, 22–26 May 2017. pp. 3–18.
29. Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE. 2015;10(7):e0130140.
30. Chaudhuri K, Monteleoni C. Privacy-preserving logistic regression. In: Advances in neural information processing systems. 2009. pp. 289–296.
31. Hitaj B, Ateniese G, Perez-Cruz F. Deep models under the GAN: information leakage from collaborative deep learning. In: Proceedings of the 24th ACM SIGSAC conference on computer and communications security, Dallas, TX, USA, 30 Oct–03 Nov 2017. pp. 603–618.
32. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, Denver, CO, USA, 12–16 Oct 2015. pp. 1310–1321.
33. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 23rd ACM SIGSAC conference on computer and communications security, Vienna, Austria, 24–28 Oct 2016. pp. 308–318.
34. McMahan HB, Ramage D, Talwar K, Zhang L. Learning differentially private recurrent language models. In: Proceedings of the 6th international conference on learning representations, Vancouver, BC, Canada, 30 Apr–3 May 2018.
35. Acs G, Melis L, Castelluccia C, De Cristofaro E. Differentially private mixture of generative neural networks. IEEE Trans Knowl Data Eng. 2018;31(6):1109–21.
36. Blum A, Dwork C, McSherry F, Nissim K. Practical privacy: the SuLQ framework. In: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems. 2005. pp. 128–138.
37. Chitta R, Jin R, Jain AK. Efficient kernel clustering using random fourier features. In: IEEE 12th international conference on data mining, Brussels, Belgium, 10–13 Dec 2012. pp. 161–170.
38. Kingma DP, Welling M. Auto-encoding variational bayes. In: 2nd International conference on learning representations, Banff, AB, Canada, 14–16 Apr 2014
39. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. 1st ed. Cambridge: MIT press; 2016.
40. Xie L, Lin K, Wang S, Wang F, Zhou J. Diffeentially private generative adversarial network. arXiv preprint arXiv:1802.06739. 2018. https://arxiv.org/abs/1802.06739. Accessed 15 Apr 2020.

41. Yu L, Liu L, Pu C, Gursoy ME, Truex S. Differentially private model publishing for deep learning. In: Proceedings of the 40th IEEE symposium on security and privacy, San Francisco, CA, USA, 19–23 May 2019. pp. 332–349.
42. Lee H, Grosse R, Ranganath R, Ng AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th annual international conference on machine learning. 2009. pp. 609–616.
43. Phan N, Wu X, Dou D. Preserving differential privacy in convolutional deep belief networks. J Mach Learn. 2017;106(9–10):1681–704.
44. Phan N, Wu X, Hu H, Dou D. Adaptive laplace mechanism: differential privacy preservation in deep learning. In: Proceedings of the 17th IEEE international conference on data mining (ICDM), New Orleans, LA, USA, 18–21 Nov 2017. pp. 385–394.
45. Papernot N, Abadi M, Erlingsson U, Goodfellow I, Talwar K. Semi-supervised knowledge transfer for deep learning from private training data. In: Proceedings of the 5th international conference on learning representations, Toulon, France, 24–26 Apr 2017.
46. Papernot N, Song S, Mironov I, Raghunathan A, Talwar K, Erlingsson Ú. Scalable private learning with pate. In: Proceedings of the 6th international conference on learning representations, Vancouver, BC, Canada, 30 Apr–3 May 2018.
47. Triastcyn A, Faltings B. Generating differentially private datasets using gans. arXiv preprint arXiv:1803.03148. 2018. https://128.84.21.199/abs/1803.03148v1. Accessed 15 Apr 2020.
48. Nguyen TAT, Dang TK. Privacy preserving biometric-based remote authentication with secure processing unit on untrusted server. J IET Biom. 2019;8(1):79–91.
49. Thi QNT, Dang TK. Towards a fine-grained privacy-enabled attribute-based access control mechanism. Trans Large Scale Data Knowl Cent Syst. 2017;36:52–72.
50. Dang TK, Pham CDM, Nguyen TLP. A pragmatic elliptic curve cryptography-based extension for energy-efficient device-to-device communications in smart cities. Sustain Cities Soc. 2020. https://doi.org/10.1016/j.scs.2020.102097.
51. Dang TK, Tran KTK. The meeting of acquaintances: a cost-efficient authentication scheme for light-weight objects with transient trust level and plurality approach. In: Security and Communication Networks, Hindawi, vol. 2019. 2019.
52. Bengio Y. Learning deep architectures for AI. In: Foundations and trends® in machine learning. 2009. vol. 2, no. 1, pp. 1–127.
53. Long Y, Bindschaedler V, Wang L, Bu D, Wang X, Tang H, Chen K. Understanding membership inferences on well-generalized learning models. arXiv preprint arXiv:1802.04889. 2018. https://arxiv.org/abs/1802.04889. Accessed 15 Apr 2020.
54. Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy risk in machine learning: Analyzing the connection to overfitting. In: Proceedings of the 31st IEEE computer security foundations symposium, Oxford, United Kingdom, 9–12 July 2018. pp. 268–282.
55. Phan N, Wang Y, Wu X, Dou D. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In: Proceedings of the 30th AAAI conference on artificial intelligence, Phoenix, Arizona, USA, 12–17 Feb 2016.