Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 92-108
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Defense against Membership Inference Attack Applying Domain Adaptation with Addictive Noise

Hongwei Huang

College of Information Science and Technology, Jinan University, Guangzhou, China
Email: h.w.huang119@outlook.com

How to cite this paper: Huang, H.W. (2021) Defense against Membership Inference Attack Applying Domain Adaptation with Addictive Noise. *Journal of Computer and Communications*, **9**, 92-108.
https://doi.org/10.4236/jcc.2021.95007

Received: April 25, 2021
Accepted: May 25, 2021
Published: May 28, 2021

Abstract

Deep learning can train models from a dataset to solve tasks. Although deep learning has attracted much interest owing to the excellent performance, security issues are gradually exposed. Deep learning may be prone to the membership inference attack, where the attacker can determine the membership of a given sample. In this paper, we propose a new defense mechanism against membership inference: NoiseDA. In our proposal, a model is not directly trained on a sensitive dataset to alleviate the threat of membership inference attack by leveraging domain adaptation. Besides, a module called Feature Crafter has been designed to reduce the necessary training dataset from 2 to 1, which creates features for domain adaptation training using noise addictive mechanisms. Our experiments have shown that, with the noises properly added by Feature Crafter, our proposal can reduce the success of membership inference with a controllable utility loss.

Keywords

Privacy-Preserving Machine Learning, Membership Inference Attack, Domain Adaptation, Deep Learning

1. Introduction

Deep learning progressively participates in society and our daily life owing to excellent performance. For example, in the field of computer vision, deep learning is widely used for image classification, object detection, and facial recognition [1] [2]. Except for computer vision, deep learning is also used in recommendation [3] or disease prediction [4] [5]. According to the report released by Technavio [6], the market of deep learning can reach 7.2 billion USD in

2020-2024.

Despite the fancy performance of deep learning, the models can face plenty of threats during deployment, one of which will severely expose deep learning models to privacy risk. Attacks aiming at the private information of deep learning models, such as model inversion and attribute inference [7] [8] [9] are proposed in recent years. Among these attacks, membership inference attacks are widely studied [10] [11] [12] [13] [14]. Briefly speaking, in a membership inference attack, the attacker's target is to guess whether a given sample is a training sample of the victim model, which is called a member. The attacker has various variables to leverage to initiate the attack. Commonly, the direct outputs associated with a given sample sent back by the victim model are exploited.

Membership inference attacks can cause severe privacy breaches in some circumstances, especially when a deep learning model is trained by a dataset containing sensitive user data. Liu *et al.* [15] propose a membership inference attack against patients' medical records, which can result in disease discrimination issues. Pyrgelis *et al.* [16] propose another membership inference attacker against aggregate location data, in which a user's history locations or activity trace can be stolen by the attacker. These leakages may introduce pecuniary losses or legal violations to individuals or organizations that provide deep learning for service. For example, it is forced by GDPR (General Data Protection Regulation) [17] to protect user's privacy. The failure in the protection of user privacy can result in a large amount of penalty.

Since membership inference exposes deep learning models in privacy threat, efforts have been put to counter the attack. These efforts can be categorized into three types: Regularization-based defenses; Adversarial-example-based defenses and Differential-privacy-based defenses. Regularization-based defenses choose to utilized regularization techniques to design defense mechanisms. For example, Shokri *et al.* [10] shown that Dropout and L2 Regularization can be used to alleviate membership inference. Salem *et al.* [11] leverage an ensemble learning technique called model stacking to build defenses. Adversarial-example-based defenses utilize adversarial examples to design defenses. However, the defenses incur large overhead such as time consumed in deploying the defenses. Differential-privacy-based defenses make use of the differential privacy techniques to add noises to model gradients during model training. Jayaraman *et al.* [18] show the effectiveness of such defenses. However, this type of defense will incur large utility losses. The main goals of the defense mechanisms are to 1) reduce the success of membership inference; 2) keep the maximum utility of the victim model, which is also the requirement of our defense mechanism.

The reason for the success of membership inference is attributed to the phenomenon of overfitting [10] [19]. Because of overfitting, a model may treat the training data in a special pattern. This unique pattern can be leveraged by the attacker to deploy membership inference. Consider a sensitive dataset, to protect the membership information from leakage; a simple idea is not to train the model directly on the sensitive dataset. To train a model without directly using the complete sensitive dataset is the first challenge. We solve the problem by leveraging transfer learning, specifically, the domain adaptation technique, inspired by DAMIA [20]. Since in domain adaptation, a model is trained with two datasets: the source domain dataset and the target domain dataset. Note the model is directly trained on the source domain dataset and the knowledge is transferred to tackle the problem on the unlabeled target domain dataset. In other words, the model is not directly trained on the target domain dataset, which can be the sensitive dataset. However, in this design, another challenge is that an extra dataset should be collected or generated to be the source domain dataset, which brings inconvenience in deploying the defenses. We solve the problem by creating the data for domain adaptation based on the sensitive dataset. Instead of creating samples fed to the model, we choose to create the feature in the middle layer of the model based on the one extracted from the sensitive dataset. These two features will be used for domain adaptation training. To avoid complex data generation, which will introduce a burden in the defense, we solve the problem by adding noises following noise mechanisms to the target domain feature, where two "addictive" (more noise means more privacy) noise mechanisms are designed.

Based on the idea above, we propose NoiseDA. Our proposal consists of three phases during applying the defenses: 1) feature extraction; 2) feature crafting and 3) model training. In feature extraction, features will be extracted from the sensitive dataset (*i.e.* the target domain features), these features are then sent to Feature Crafter in the feature crafting phase. In this phase, the source domain feature will be created. Afterward, the source and target domain features are then used for the later domain adaptation training.

We conduct experiments to evaluate our proposal. The experiment results show that NoiseDA can resist membership inference attacks by reducing the membership inference accuracy. Besides, with proper noises added, the utility loss can be controlled.

Contributions. Our paper makes the following contributions:

- We propose a new defense mechanism leveraging domain adaptation and further reduce the necessary training dataset from 2 to 1.
- We design two noise-adding strategies in our proposal; both of the strategies can create suitable training features.
- We design and evaluate noise-adding strategy based on differential privacy. Results show the capability to leverage differential privacy in our proposal.
- We conduct experiments on the benchmark datasets to evaluate our proposal, which indicates the effectiveness of the proposal.

Roadmap: The rest of the paper is organized as follows. In Section 2, we introduce related works on membership inference attack and defenses. In Section 3, we provide preliminary, the insight and the design of our proposal. In Section 4, we introduce the experiment setup and evaluate the effectiveness and utility of NoiseDA. Finally, we conclude the paper in Section 5.

2. Related Works

2.1. Membership Inference Attack

Shokri *et al.* [10] propose the first membership inference against deep learning, which explores the vulnerability of deep learning models.

To simplify the attack by decreasing the number of shadow models, as well as the datasets required to train these models, Salem *et al.* [11] propose a new method to initiate membership inference. Considered that the classical membership inference attack requires numerous models and datasets, despite the simplification, the new attack is still of effectiveness. Owing to the effectiveness and simplicity, this method is adopted as the membership inference attack in our experiments.

Yeom *et al.* [19] and Salem *et al.* [11] propose new attacks by comparing the confidence score of the target class with a predefined threshold. Mathematically speaking, given a victim DL model *M* trained on a dataset *D*. It is presumed that the victim model can categorize inputs into *n* categories. A sample s_i fed into *M* results in an output $o_i = \left(p_1, p_2, \cdots, p_j, \cdots, p_n\right)$, where p_j is the confidence score. To infer the membership, the attack compare according to the rule as follows, where p_k is the confidence score of the target class.

$$M_{adv} = \begin{cases} 1 & p_k \ge P_{thresd} \\ 0 & p_k < P_{thresd} \end{cases}$$

However, compared with the shadow-model-based attack, these attacks require more information [11] [19].

Apart from attacks under the black-box access, there are other attacks proposed under the white-box access. In this access model, an attacker has the ability to access more information; therefore, the attacker has more options to initiate the attack rather than only using the outputs from the models. Nasr *et al.* [11] shown that, in this scenario, an attacker can leverage other useful information, such as the activation values, gradients to perform the attack. Especially, in federated learning [21] scenario, the attacker can also use the information of parameter updates as the role of parameter aggregator. In spite of the effectiveness, these attacks are less practical compared with those in a black-box manner, since in the real world, deep learning models usually provide service under an MLaaS (*i.e.* Machine-Learning-as-a-Service) scenario, where less information of the model, except that the output is sent to the users.

Some membership inference attacks aiming at the real-world application are also proposed. Pyrgelis *et al.* [16] propose an attacker to infer the membership of a given user in aggregate location data and further acquire the user's location. Liu *et al.* [15] propose another membership inference attack to acquire the membership of patients, whose data are used to train a disease prediction model.

Although the membership inference attack exposes the vulnerability of deep learning models and the privacy of the training dataset in danger, the attacker itself enhances the understanding of deep learning. Besides, membership inference also serves as a method for the information leakage of a deep learning model, which is usually utilized to reflect the performance of defenses.

2.2. Defenses against Membership Inference Attack

Given that membership inference attacks may breach the privacy of the training set, especially when the dataset is sensitive, defenses are explored and proposed. Current defenses against membership inference can be categorized into three groups: regularization-based defenses, adversarial-example-based defenses, and differential-privacy-based defenses.

2.2.1. Regularization-Based Defenses

The possible reason for the success of membership inference attacks can be attributed to the phenomenon of overfitting [19]. A deep learning model can be viewed as a machine processing the input layer-by-layer; the output varies according to a specific input. If a model heavily overfits the training data, the model may have a special pattern in processing the training samples, and thus this pattern can reside in the output. This pattern can further reside in the associated output, which can later be exploited by the attacker to initiate membership inference attacks. Therefore, one way to ease the problem of membership inference is to remove this pattern. In other words, narrowing down the difference between the processing of a training sample and the processing of a non-training sample can be a solution to the membership inference attack. Shokri *et al.* [10] showed that the overfitting prevention mechanisms, such as Dropout [22] and L2 regularization [23] can be used to resist membership inference attacks. However, these methods are not stable since they are not practically designed for membership inference attacks. Nasr *et al.* [24] proposed a new regularization method called adversarial regularization as a defense. In their solution, a new regularization term is added to the loss function of the deep learning model, which involves the "gain of the inference" term. This term is determined during the training process. The object of the loss function is to first maximize the gain of the inference, and then minimize it, which is also called the max-min game. Since it is an adversarial training process, the difficulty to train a model increases. Considered that ensemble learning can achieve better generalization, alleviating the phenomenon of overfitting of a single model, Salem *et al.* [11] propose a defense utilizing model stacking. In their design, a two-layer stacked ensemble model is built, which includes a neural network, a random forest in the first layer, and a logistic regression model in the second layer. Although their design show effectiveness, this method involves more models to train, and thus more data are needed for training. In contrast, our defense does not have this limitation.

2.2.2. Adversarial-Example-Based Defenses

As mention in Section 2.1, typically, a membership inference attacker will train a binary classifier accepting the outputs from the victim model to perform attacks.

Usually, the binary classifier is a deep learning model; as a result, the vulnerability of deep learning resides in the model, such as the vulnerability in facing adversarial attacks [25]. Based on this insight, Jia *et al.* [26] propose a defense mechanism against membership inference attacks. In their design, carefully designed noises are added to the victim model's outputs to turn the outputs into adversarial examples. Therefore, the adversarial outputs can fool the attacker's binary classifier, and thus fail the membership inference attack. Although the defense shows great performance, more time must be consumed in applying the defense.

2.2.3. Differential-Privacy-Based Defenses

Some defenses focus on the technique of differential privacy. Abadi *et al.* [27] propose DP-SGD algorithm to replace the original SGD for model training, their results show that differential privacy is effective. However, Rahman *et al.* [28] show that even with DP-SGD, the model is still possible to be compromised. Carlini *et al.* [29] exam different differential privacy training algorithms in the deep learning model to resist membership inference. Their results show that, although differential privacy can be leveraged to resist membership inference, the utility of the model (*i.e.* the performance of a model on a specific task) is extensively affected and is usually reduced on a large scale. Our defense makes use of differential privacy; however, it is not used for model training and training sample generation.

3. Methodology

In this section, the preliminary knowledge is demonstrated. Afterward, the threat model discussed in our work is defined. Finally, we clarify our idea to design NoiseDA and present the design details.

3.1. Preliminary

3.1.1. Membership Inference Attack

Membership inference attacks against Deep Learning models are proposed by Shokri *et al.* [10]. The attack aims to infer whether a sample, fed to a Deep Learning model, is included in the training set of the model. A training sample of a Deep Learning model is also known as the "member".

Commonly, the membership inference attack is performed in a black-box manner, meaning that an attacker can only interact with the victim model in a query-answer manner. In this setting, the attacker can only exploit useful information in the output sent back by the victim model, which is usually a posterior.

Typically, to initiate a membership inference attack under the black-box scenario, the attack must firstly train a group of models imitating the victim model, also known as "shadow models". To train these models, the attacker obtains datasets following the same distribution as the training set of the victim model. After all the shadow models are trained, the attacker further feeds the training set and the test set to these models and collects all the outputs. The outputs are separated into two groups to form a new dataset: those associated with the training samples are labeled as members, and those associated with the training samples are labeled as non-members. Based on the new dataset, the attacker trains a binary classifier as the attack model. By feeding output from the victim model, the attacker can infer the membership of the sample associated with the output.

3.1.2. Domain Adaptation

Domain Adaptation (DA) is a branch of transfer learning [30], aiming to address the issue of insufficient labeled training data.

Domain adaptation utilizes the knowledge of one or more relevant source domains to conduct new tasks in a target domain. Mathematically, we denote a domain as $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ represents the feature space and $P(X)$ represents the margin probability distribution. Note that $X = \{x_1, x_2, \cdots, x_n\} \in \mathcal{X}$. A task on a specific domain is denoted as $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, where $\mathcal{Y}$ is the label space and $f(x)$ is the target prediction function. Therefore, a source domain can be represented as $\mathcal{D}_t = \{\mathcal{X}_t, P(X)_t\}$. Correspondingly, $\mathcal{T}_s = \{\mathcal{Y}_s, f(x)_s\}$ and $\mathcal{T}_t = \{\mathcal{Y}_t, f(x)_t\}$ are two tasks. The goal of DA is to leverage the latent knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$ to improve the performance of $f(x)_t$ in $\mathcal{T}_t$, where $\mathcal{D}_s \neq \mathcal{D}_t$. Please note that in domain adaptation, $\mathcal{T}_s = \mathcal{T}_t$.

Basically, the approach achieves the knowledge transferring by driving the model to learn the shared representation of the source domain and target domain.

3.1.3. Differential Privacy

Differential privacy is introduced by Dwork [31], this technique protects user privacy by adding noises to the data in a dataset. However, the noises will not affect meaningful analysis on the dataset. In other words, the statistical properties are mostly preserved. Formally, the definition of differential privacy is as follows.

Given a randomized algorithm *A*, a particular result *x* and pairs of datasets D and D', where D is almost the same as D' with one record missing. *A* achieve ϵ-differentially privacy if and only if the following inequality holds:

$$Pr\left[A(D) = x\right] = e^{\epsilon} \times Pr\left[A(D') = x\right]$$

Therefore, various algorithms satisfy this condition, one of which is the Laplace mechanism [31].

Essentially, differential privacy is used to protect user privacy in a dataset. In this work, we concentrate on the property that differential privacy preserves the statistical properties of data and leverage this property to generate new data for domain adaptation training.

3.2. Threat Model

We define the threat model under the MLaaS scenario. Especially, in this scenario, the deep learning model serves the public with limited APIs exposed. The model user, as well as the attacker, can only send the inputs to the model via a predefined API and receives only the outputs from the models. In other words, attackers can only leakage the direct outputs from the model to initiate membership inference.

More specifically, we make the following assumptions for the threat model:

1) It is assumed that that the attacker can only access the victim model in a black-box manner. That is, the attacker does not know how the victim model will process the inputs fed by the attacker. The attacker only knows the output from the victim model.

2) It is assumed that the distribution that the training data of the victim model is drawn from is known by the attacker. Note that this assumption is made in most of the membership inference threat model [10] [11] [19].

3) It is also assumed that the attacker has no knowledge of the implementation of the victim model, including the training algorithm, the hyperparameters, and the model's architecture.

3.3. Insights

The insight of our design is based on that: the success of membership inference attacks is widely attributed to the phenomenon of overfitting, where the difference between the processing of training data and the processing of non-training data and the memorization phenomenon [29] can be utilized by attackers. Therefore, to prevent membership inference attacks, one possible way is not to train a model directly on a sensitive dataset. To design a defense mechanism based on this idea, we will face the following challenges.

Challenge 1: How to train the model so as the model can tackle the task on the sensitive dataset without being trained directly on the complete sensitive dataset?

Using synthetic data produced by generative adversarial networks (GANs) is out of our consideration since the process is time-consuming.

The technique of transfer learning shed light on another possible solution to this problem. Specifically, domain adaptation can be leveraged for our design to resist membership inference attacks. In domain adaptation, a model is trained by two datasets: a target domain dataset and a source domain dataset. The target domain dataset contains only the samples without labels. With the help of the source domain dataset, the model can have a good performance on the target domain dataset, which fits our need in Challenge 1.

Challenge 2: How to avoid collecting or generating an extra dataset for the source domain when domain adaptation is adopted?

We choose the classical domain adaptation method called Deep Domain Confusion (DDC) [32] since it is simple and effective. In DDC, the source domain dataset and the target domain dataset are fed into a universal backbone network for feature extraction. Given that our goal is to avoid using source domain data, we can create a different feature according to the target domain data to bypass the usage of the source domain data. Since the crafted feature is based on the target feature, it should preserve better utility.

Challenge 3: How to craft a feature based on the target domain feature to replace the source domain feature?

Instead of using time-consuming data generation techniques such as GAN, we adopt noise-adding mechanisms to achieve the goal. Specifically, a source domain feature is acquired by adding noises following certain rules to the target domain feature. In such a way, a source domain feature can be efficiently produced. Note that it still fits the need in Challenge 1 since the crafted feature is not identical to the target feature.

3.4. Design

Based on the insights in Section 4.2, in this subsection, we propose NoiseDA. The overview of our defense mechanism is illustrated in Figure 1.

As shown in Figure 1, to train a model in our defense mechanism, three phases are involved. Note that three phases are included in each round of training.

3.4.1. Feature Extraction

In this phase, a backbone works as a feature extractor will take sensitive data as in-puts to extract the target domain features. Since neural networks show great potential in extracting lantern features for model training, in our design, neural networks are utilized as backbones. Note that, although in Figure 1, a Convolutional Neural Network (CNN) takes the role of the backbone; other types of neural networks can be the backbone depending on a certain circumstance.

3.4.2. Feature Crafting

The phase of Feature Crafting is essential to our defense. In this phase, Feature Crafter is obligate to craft features by adding noises following certain rules to the target domain features. We design two feature crafting strategies for our defense mechanism: uniform noise strategy and differential privacy noise strategy.

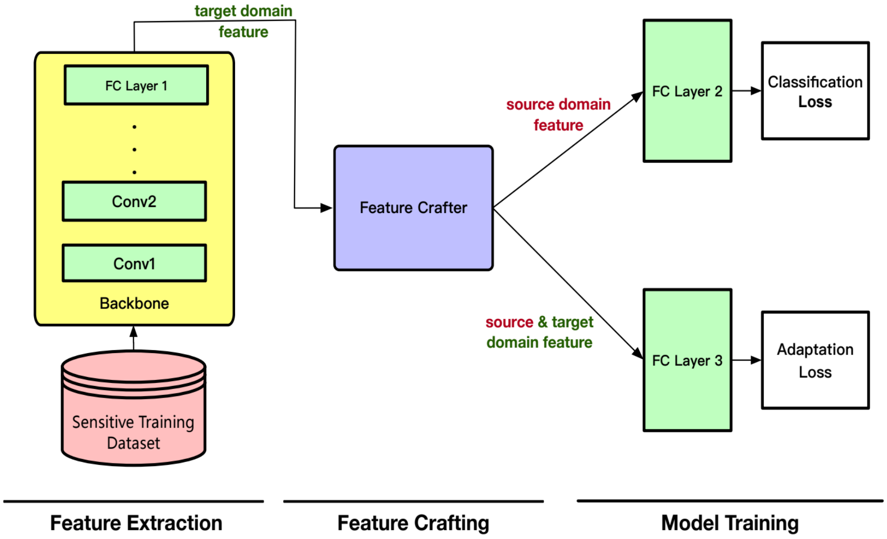


Figure 1. Overview of NoiseDA.

Uniform noise strategy. The uniform noise strategy is simple and easy to perform. In this strategy, noises added to the target domain features are sampled from the uniform distribution. The feature crafting algorithm is illustrated in Algorithm 1.

As shown in Algorithm 1, noises are firstly drawn from the uniform distribution, whose range is controlled by λ. With a smaller λ, the crafted source domain feature will be less different compared with the target domain features. In our design, the default λ is set as 1. Additionally, to avoid overflow or underflow, the crafted feature should be constrained in the fixed range, which is defined by max value and min value.

Differential privacy noise strategy. Apart from the uniform noise strategy, we introduce another strategy leveraging differential privacy. We make use of the idea that, in differential privacy, the added noises can be controlled so that the useful information (e.g. statistical properties) of data is still preserved.

As shown in Algorithm 2, noises are drawn from the Laplace distribution according to the design in [31], whose form is controlled by sensitivity and ϵ. ϵ is called the privacy budget, with a smaller ϵ, the source domain feature will be more different compared with the target domain features, which should enhance the privacy protection, according to the definition of differential privacy. In our design, the default ϵ is set as 0.1.

```
Input: Target Domain Feature $X_t$, Noise Range $\lambda$
Output: Source Domain Feature $X_s$
Function AddUinformNoise(feature, epsilon):
    source_feature = feature
    noise = Uniform(-epsilon, epsilon)
    source_feature = source_feature + noise
    source_feature = clamp(source_feature, max_val, min_val)
    return source_feature

Function UniformNoiseCraftor:
    $X_s$ = AddUinformNoise($X_t$, $\lambda$)
    return $X_s$
```

Algorithm 1. Uniform noise feature crafting algorithm.

```
Input: Target Domain Feature $X_t$, Sensitivity $s$, Epsilon $\epsilon$
Output: Source Domain Feature $X_s$
Function AddLaplaceNoise(feature, epsilon):
    source_feature = feature
    noise = Laplace(0, s/$\epsilon$ )
    source_feature = source_feature + noise
    source_feature = clamp(source_feature, max_val, min_val)
    return source_feature

Function Laplace(sensitivity, epsilon):
    beta = sensitivity / epsilon
    u1 = unifrom(0,1)
    u2 = unifrom(0,1)
    if condition then
        n = -beta * log(1 - u2)
    else
        n = beta * log(u2)
    return n

Function DifferentialPrivacyCraftor:
    $X_s$ = AddLaplaceNoise($X_t$, $\epsilon$)
    return $X_s$
```

Algorithm 2. Differential privacy noise feature crafting algorithm.

3.4.3. Model Training

In this phase, the crafted source domain feature along with the target domain feature will be used to calculate the classification loss and the adaptation loss, which will guide the model to update its weights.

The classification loss is calculated with the crafted source domain features, where labels are used. While the adaptation loss, involved with the two features, is calculated by MMD [32], which reflects the distance between these features. In our design, the SGD optimizer is utilized for minimizing both the classification loss and the adaptation loss.

4. Evaluation

In this section, we explore the performance of NoiseDA, that is, the effect of resisting membership inference and the utility of the model after applying our defense. We firstly introduce the experiment setup, including the experiment environment, as well as the experiment dataset involves in the evaluation. Secondly, we introduce metrics to reflect the performance of NoiseDA. Afterward, we present the experiment results.

4.1. Experiment Environment

Experiments are conducted on a Ubuntu 16.04 server, equipped with an Intel(R) Core(TM) i5-7500 CPU, an Nvidia GTX 1080Ti GPU, and memory with a size of 32 GB.

We implement our design using Python, and Pytorch is adopted as the deep learning framework in our implementation.

4.2. Experiment Datasets

In this subsection, we introduce the dataset involved in our evaluation. Note that the experiment datasets are the benchmark datasets [10] [11] [28] for membership inference attacks, involving image datasets and non-image datasets.

CIFAR-100. The CIFAR-100 dataset consists of 60,000 color images with a size of 32 × 32. Samples in this image can be categorized into 100 classes, such as airplane, cat, dog, or horse, and so on. Usually, 50,000 images are used as training samples while the rest are test samples. Totally 40,000 samples are randomly selected to be our experiment dataset.

Location. The Location dataset is constructed based on the foursquare dataset, which contains 5010 samples. Each sample is a feature vector with a length of 466. The feature indicates the region or location types that a user visits. All the samples are categorized into 30 classes. 4500 samples are randomly selected to be our experiment dataset.

Purchase. The Purchase dataset is constructed based on the dataset in Kaggle's "acquire valued shoppers" challenge, which contains 311,540 samples. Each sample is a feature vector containing 600 binary features. The feature indicates whether a user has purchased a product. All the samples are grouped into 100 classes. 30,000 samples are randomly selected to be our experiment dataset.

Texas. The Texas dataset is constructed based on the Hospital Discharge Data public user files released by the Texas Department of State Health Services, which contains 67,330 samples. Each sample is a feature vector containing 6170 binary features. The feature presents a patient's specific medical information, as well as other sensitive information such as age and gender. All the samples are grouped into 100 classes. 30,000 samples are randomly selected to be our experiment dataset.

Note that all the experiment datasets are further separated into training datasets (80%) and test datasets (20%) for evaluation.

4.3. Metric

In this subsection, we introduce the metrics to measure the performance of our defense mechanism.

4.3.1. Effectiveness

We utilize the accuracy of membership inference attacks on the victim models to reflect the effectiveness. Specifically, the accuracy of membership inference attack is denoted as Acc_{adv}. The closer the Acc_{adv} is to 50%, the better effect the defense has. Since if the Acc_{adv} is to 50%, that means the attacker infers the membership in a random guess manner, which indicates that the attack has no other information to leverage for membership inference.

Note that in our evaluation, the simplified membership inference attack proposed in [11] is used to calculate Acc_{adv}, since this attack is easier to perform and requires less information, which is more practical in the real-world setting, such as MLaaS.

Concretely, the attack model is trained as follow:

1) The attacker train only one local shadow model, whose behavior is similar to the victim model;

2) The attacker collects the output from the shadow model by feeding the training data and non-training and labeled as member and non-member respectively. Noted that these two datasets are of the same size;

3) The attacker uses the collected output to train a binary classifier as the attack model.

4.3.2. Utility

We utilize the test accuracy of the victim to reflect the utility of the victim model. Specifically, the test accuracy is denoted as Acc_{test}. By comparing the utility of a victim model before and after applying the defenses, we can examine whether the defense mechanism incurs large utility loss. The less the utility loss is, the better the defense is.

4.4. Evaluation of Uniform Noise Strategy

Firstly, we evaluate our defense mechanism using the uniform noise strategy. Concretely, we vary the λ in **Algorithm 1** to examine the effectiveness and utility of our defenses. Experiment results are demonstrated in **Table 1** and **Table 2**.

Table 1. Effectiveness of NoiseDA using uniform noise strategy.

λ	0.01	0.1	0.5	1	10	25	50	100	w/o Defense
Location	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	87.80%
Purchase	61.98%	61.44%	62.74%	59.62%	59.06%	56.68%	55.83%	51.75%	62.95%
Texas	70.91%	71.86%	71.52%	72.03%	71.77%	61.46%	55.74%	52.99%	75.44%
CIFAR-100	59.60%	58.90%	58.60%	58.84%	59.29%	59.29%	59.29%	59.19%	72.39%

Table 2. Utility of NoiseDA using uniform noise strategy.

λ	0.01	0.1	0.5	1	10	25	50	100	w/o Defense
Location	48.00%	48.00%	48.00%	49.00%	46.00%	47.00%	39.00%	41.00%	59.20%
Purchase	45.00%	45.00%	47.00%	47.00%	47.00%	43.00%	31.00%	24.00%	57.76%
Texas	43.00%	43.00%	44.00%	42.00%	44.00%	42.00%	29.00%	23.00%	44.10%
CIFAR-100	24.00%	24.00%	25.00%	18.00%	1.00%	0.00%	0.00%	1.00%	27.50%

As shown in Table 1 and Table 2, we can observe that with our defense applied, and the $\mathrm{Acc}_{\mathrm{adv}}$ is always reduced. Besides, although utility loss is inevitable, similar to other defenses [11] [18], the utility loss in our defense is within a small range, and can be controlled by adjusting the noise parameter.

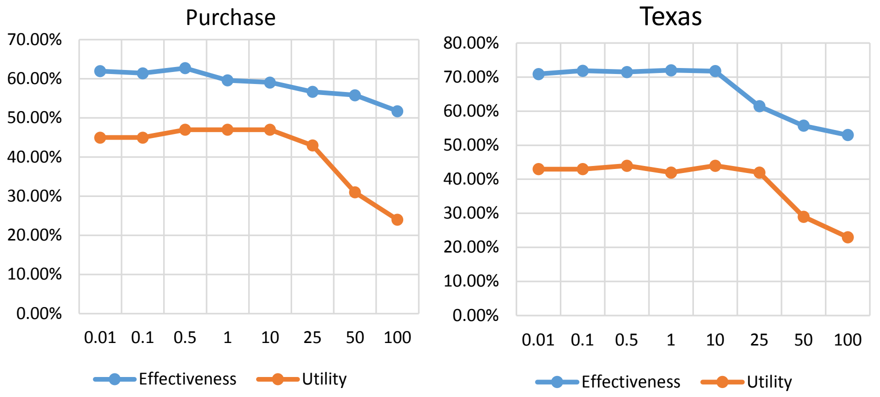
With a proper λ chosen, the utility of the victim is overall close to the one without defense. For example, for the task on dataset Location, λ can be chosen as 0.5 to achieve better performance; for Purchase, λ can be chosen as 25. Since in uniform noise strategy, the parameter λ is used to control the number of noises to add, unexpectedly, as λ getting large, our defense achieves better effectiveness, although larger utility loss will occur. The tendency can be observed in Figure 2, where the results of Purchase and Texas are plotted in Figure 2(a) and Figure 2(b).

4.5. Evaluation on Uniform Noise Strategy

We further evaluate our defense mechanism using the differential privacy noise strategy. This time, we vary the epsilon in Algorithm 2 to examine the effectiveness and utility of our defenses. Experiment results are demonstrated in Table 3 and Table 4.

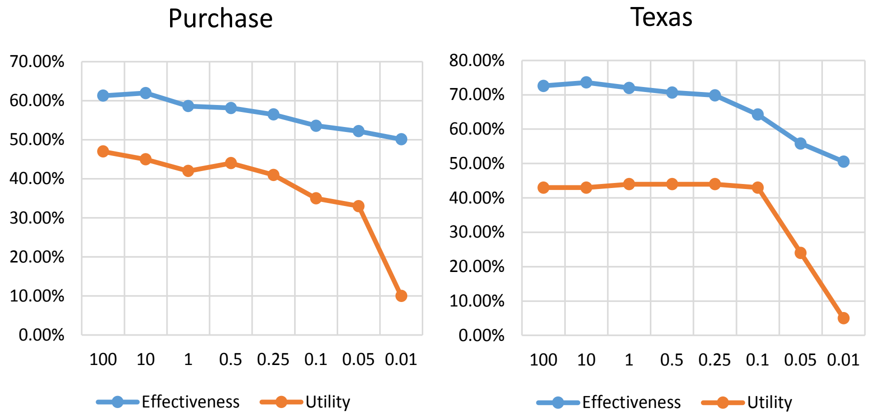
With a proper ϵ chosen, the utility of the victim is overall close to the one without defense, such as 10 for Location and 0.1 for Texas. Compared with uniform noise strategy, the differential privacy noise strategy using Laplace noise is better at dealing with simpler task, such as the task on Location.

In differential privacy noise strategy, ϵ, called the privacy budget, is the parameter to control the number of noises to add. As ϵ ascends, the defenses have a better utility, while the effectiveness is weakened, meaning that $\mathrm{Acc}_{\mathrm{adv}}$ increases. Therefore, a smaller ϵ can be considered first to achieve better effectiveness and further finetune the parameter to obtain better utility. Likewise, we plot the results of Purchase and Texas in Figure 3(a) and Figure 3(b) to show the tendency.



(a) (b)

Figure 2. Trend of effectiveness and utility as λ increases. (a) Trend on purchase dataset; (b) Trend on Texas dataset.



(a) (b)

Figure 3. Trend of effectiveness and utility as ϵ desreases. (a) Trend on purchase dataset; (b) Trend on Texas dataset.

Table 3. Effectiveness of NoiseDA using differential privacy noise strategy.

ϵ	0.01	0.05	0.1	0.25	0.5	1	10	100	w/o Defense
Location	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	87.80%
Purchase	50.12%	52.20%	53.60%	56.48%	58.12%	58.62%	61.97%	61.27%	62.95%
Texas	50.57%	55.84%	64.25%	69.85%	70.67%	72.00%	73.58%	72.58%	75.44%
CIFAR-100	58.98%	59.32%	58.99%	59.68%	59.35%	59.36%	59.26%	59.38%	72.39%

Table 4. Utility of NoiseDA using differential privacy noise strategy.

ϵ	0.01	0.05	0.1	0.25	0.5	1	10	100	w/o Defense
Location	43.00%	44.00%	47.00%	49.00%	49.00%	44.00%	50.00%	48.00%	59.20%
Purchase	10.00%	33.00%	35.00%	41.00%	44.00%	42.00%	45.00%	47.00%	57.76%
Texas	5.00%	24.00%	43.00%	44.00%	44.00%	44.00%	43.00%	43.00%	44.10%
CIFAR-100	1.00%	1.00%	1.00%	1.00%	1.00%	0.00%	9.00%	23.00%	27.50%

In a nutshell, both noise strategies are all appliable in NoiseDA. Both show the ability to protect the model from membership inference. The differential privacy noise strategy shows that the technique of differential privacy is compatible with our design, indicating the possibility of applying other differential privacy mechanisms in our design.

Although we can observe that the utility loss is inevitable and may get larger when more noises are added. It can be attributed to the technique of domain adaptation in extracting useful information from both features. We leave the improvement in our future work. Besides, parameters in both noise strategies are currently chosen empirically. Automatic parameter searching algorithms are also our focus in the future.

5. Conclusion

In this paper, we propose a new defense mechanism against membership inference attacks. The proposal leverages domain adaptation to avoid direct training on a sensitive dataset. Besides, the Feature Crafter module is designed to create a feature for domain adaptation training by utilizing addictive noise mechanisms, which can reduce the necessary dataset from 2 to 1. We further design noise-adding strategies for the module. We show in the experiment that our proposal can resist the membership inference attack. Besides, with proper noises added, the utility loss can be controlled. The next stage of our works is to reduce the gaps between effectiveness and utility by designing better addictive noise mechanisms and domain adaptation training methods, and design automatic parameter searching algorithms for our noise-adding strategies.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E. (2017) A Survey of Deep Neural Network Architectures and their Applications. *Neurocomputing*, **234**, 11-26. https://doi.org/10.1016/j.neucom.2016.12.038

[2] Masi, I., Wu, Y., Hassner, T. and Natarajan, P. (2018) Deep Face Recognition: A Survey. 31st *SIBGRAPI Conference on Graphics*, *Patterns and Images*, Parana, 29 October-1 November 2018, 471-478. https://doi.org/10.1109/SIBGRAPI.2018.00067

[3] Zhang, S., Yao, L., Sun, A. and Tay, Y. (2019) Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, **52**, Article No. 5. https://doi.org/10.1145/3285029

[4] Chen, M., Hao, Y., Hwang, K., Wang, L. and Wang, L. (2017) Disease Prediction by Machine Learning over Big Data from Healthcare Communities. *IEEE Access*, **5**, 8869-8879. https://doi.org/10.1109/ACCESS.2017.2694446

[5] Betancur, J., Commandeur, F., Motlagh, M., Sharir, T., Einstein, A.J., Bokhari, S., *et al*. (2018) Deep Learning for Prediction of Obstructive Disease from Fast Myocardial Perfusion Spect: A Multicenter Study. *JACC*: *Cardiovascular Imaging*, **11**, 1654-1663.

https://doi.org/10.1016/j.jcmg.2018.01.020

[6] Technavio (2020) Deep Learning Market by Type and by Geography—Global Opportunity Analysis and Industry Forecast, 2020-2024
https://www.technavio.com/report/deep-learning-market-industry-analysis

[7] Fredrikson, M., Jha, S. and Ristenpart, T. (2015) Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *Proceedings of the* 22*nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, 12-16 October 2015, 1322-1333. https://doi.org/10.1145/2810103.2813677

[8] Gong, N.Z. and Liu, B. (2016) You Are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors. 25*th USENIX Security Symposium*, Austin, 10-12 August 2016, 979-995.

[9] Hidano, S., Murakami, T., Katsumata, S., Kiyomoto, S. and Hanaoka, G. (2017) Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-sensitive Attributes. 15*th Annual Conference on Privacy*, *Security and Trust*, Calgary, 28-30 August 2017, 115-126. https://doi.org/10.1109/PST.2017.00023
https://dblp.uni-trier.de/rec/conf/pst/HidanoMKKH17.html?view=bibtex

[10] Shokri, R., Stronati, M., Song, C. and Shmatikov, V. (2017) Membership Inference Attacks Against Machine Learning Models. 2017 *IEEE Symposium on Security and Privacy*, San Jose, 22-24 May 2017, 3-18. https://doi.org/10.1109/SP.2017.41

[11] Salem, A., Zhang, Y., Humbert, M., Fritz, M. and Backes M. (2019) ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. *Network and Distributed Systems Security Symposium* 2019, California, 24-27 February 2019.
https://doi.org/10.14722/ndss.2019.23119

[12] Nasr, M., Shokri, R. and Houmansadr, A. (2019) Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. 2019 *IEEE Symposium on Security and Privacy*, San Francisco, 20-22 May 2019, 739-753. https://doi.org/10.1109/SP.2019.00065

[13] Shokri, R., Strobel, M. and Zick, Y. (2019) On the Privacy Risks of Model Explanations. arXiv: 1907.00164.

[14] Chen, D., Yu, N., Zhang, Y. and Fritz, M. (2020) Gan-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. *Proceedings of the* 2020 *ACM SIGSAC Conference on Computer and Communications Security*, New York, 9-13 November 2020, 343-362. https://doi.org/10.1145/3372297.3417238

[15] Liu, G., Wang, C., Peng, K., Huang, H., Li, Y. and Cheng, W. (2019) SocInf: Membership Inference Attacks on Social Media Health Data with Machine Learning, *IEEE Transactions on Computational Social Systems*, **6**, 907-921.
https://doi.org/10.1109/TCSS.2019.2916086

[16] Pyrgelis, A., Troncoso, C. and De Cristofaro, E. (2018) Knock Knock, Who's There? Membership Inference on Aggregate Location Data. *Network and Distributed Systems Security Symposium* 2018, San Diego, 18-21 February.
https://dblp.org/rec/conf/ndss/PyrgelisTC18.html?view=bibtex%20which%20shows%20no%20page%20numbers

[17] Voigt, P. and Von dem Bussche, A. (2017) The EU General Data Protection Regulation (GDPR): A Practical Guide. 1st Edition, Springer International Publishing, Cham.
https://doi.org/10.1007/978-3-319-57959-7

[18] Jayaraman, B. and Evans, D. (2019) Evaluating Differentially Private Machine Learning Practice. 28*th USENIX Security Symposium*, Santa Clara, 14-16 August 2019, 1895-1912.

[19] Yeom, S., Giacomelli, I., Fredrikson, M. and Jha, S. (2018) Privacy Risk Machine Learning: Analyzing the Connection to Overfitting. *IEEE 31st Computer Security Foundations Symposium*, Oxford, 9-12 July 2018, 268-282. https://doi.org/10.1109/CSF.2018.00027

[20] Huang, H., Luo, W., Zeng, G., Weng, J., Zhang, Y. and Yang, A. (2020) DAMIA: Leveraging Domain Adaptation as a Defense against Membership Inference Attacks. arXiv: 2005.08016.

[21] Yang, Q., Liu, Y., Chen, T. and Tong, Y. (2019) Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology*, **10**, Article No. 12. https://doi.org/10.1145/3298981

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, **15**, 1929-1958.

[23] Ng, A.Y. (2004) Feature Selection, L_1 vs. L_2 Regularization, and Rotational Invariance. *Proceedings of the 21st International Conference on Machine Learning*, Banff, Alberta, Canada, 4-8 July 2004, 78. https://doi.org/10.1145/1015330.1015435 https://icml.cc/Conferences/2004/summary.pdf

[24] Nasr, M., Shokri, R. and Houmansadr, A. (2018) Machine Learning with Membership Privacy Using Adversarial Regularization. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, 15-19 October 2018, 634-646. https://doi.org/10.1145/3243734.3243855

[25] Goodfellow, I.J., Shlens, J. and Szegedy, C. (2014) Explaining and Harnessing Adversarial Examples. arXiv: 1412.6572.

[26] Jia, J., Salem, A., Backes, M., Zhang, Y. and Gong, N.Z. (2019) Memguard: Defending Against Black-box Membership Inference Attacks via Adversarial Examples. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, London, 11-15 November 2019, 259-274. https://doi.org/10.1145/3319535.3363201

[27] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K. and Zhang, L. (2016) Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, 24-28 October, 308-318. https://doi.org/10.1145/2976749.2978318

[28] Rahman, M.A., Rahman, T., Laganiere, R., Mohammed, N. and Wang, Y. (2018) Membership Inference Attack against Differentially Private Deep Learning Model. *Transactions on Data Privacy*, **11**, 61-79.

[29] Carlini, N., Liu, C., Erlingsson, U., Kos, J. and Song, D. (2019) The Secret Sharer: Evaluating and Testing Unintended Memorization Neural Networks. 28*th USENIX Security Symposium*, Santa Clara, 14-16 August 2019, 267-284.

[30] Pan, S.J. and Yang, Q. (2009) A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, **10**, 1345-1359. https://doi.org/10.1109/TKDE.2009.191

[31] Dwork, C. (2008) Differential Privacy: A Survey of Results. *International Conference on Theory and Applications of Models of Computation*, Xi'an, China, 25-29 April 2008, 1-19. https://doi.org/10.1007/978-3-540-79228-4_1

[32] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K. and Darrell, T. (2014) Deep Domain Confusion: Maximizing for Domain Invariance. arXiv: 1412.3474.