2024 33rd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-8461-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCN61486.2024.10637601

Hide-and-Seek: Data Sharing with Customizable Machine Learnability and Privacy

1st Hairuo Xu and 2nd Tao Shu
Department of Computer Science and Software Engineering
Auburn University
Auburn, AL, USA
{hairuoxu, tshu}@auburn.edu

***Abstract*—With the immense amount of publicly available data online, many companies and research institute are able to download the online data for free and train the machine learning models which will finally result in products that would enhance our everyday life. While enjoying the advantages of such large amount of free data, people (data providers or data owners) have the concern that their personal data may be crawled without the owner's consent. This brings out an underlying issue in the context of machine learning that in the current literature and applications, dataset owners (also referred to as "dataset providers" in the following text) can only choose between the two extreme decisions of either to share their data entirely, or not share any of their data at all. Another side of this issue is that the privacy of the dataset to be shared is either completely revealed due to the full disclosure of the dataset, which benefits the potential consumers of the dataset (referred to as dataset user/buyer in the following text); or the dataset is not shared at all which preserves the privacy, but impede the development of new technologies.**

In this paper, we propose the novel Hide-and-Seek data sharing framework that serves as a middle point between the difficult "share or no share" extreme decisions, which provides a "partial share" option based on the consumers' needs, and hence is able to protect the partial privacy of the dataset providers while sharing enough amount of data needed for the user to train their models at a desired accuracy. Extensive amount of experiments have been conducted on the CIFAR-10, Street View House Number (SVHN), and the CIFAR-100 datasets. Our experimental results verify the effectiveness of the proposed Hide-and-Seek framework. We also show in the experiments that our framework is able to protect data provider's privacy without changing the visual patterns of the dataset, and therefore, doesn't affect the regular usage of the data (such as using it as a profile photo).

***Index Terms*—Hide-and-Seek, Multi-level Data Sharing, Privacy-Preserving, Unlearnable Dataset, Dataset Recovery**

I. Introduction

In the recent decade, with the rise of big data and advancements in technology such as machine learning, we keep hearing the statement that "data is currency". Such statement highlights the value of data in the context of machine learning, since models trained on the larger and broader dataset tend to behave better during the testing phase. Along with such benefits, data from online resources are largely extracted to construct datasets such as ImageNet [1] for training purposes. While enjoying the advantages of such large amount of publicly available data, a rising concern is that many of these data are crawled without the owner's consent [2]. For example, personal data such as Facebook and LinkedIn profile photos could be collected unconsciously to train the commercial products [3], as illustrated in Figure 1. Some companies are even fined because of such data breaching activities which violate the consumers' privacy [4].

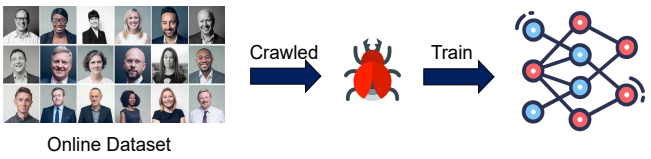

Fig. 1. Illustration of the existing problem: Online public dataset could be crawled for commercial model training purposes.

In order to prevent personal data (such as profile photos) from being collected unconsciously, existing works such as [5] and [6] try to generate perturbations to be applied to the dataset. Such perturbation works in a way that the machine learning models trained on the perturbed dataset (by patching the perturbation on the original dataset) would fail to extract the original features of the dataset and hence fail the learning objectives. However, while such methods provide the nice privacy protection functionality for the data providers, these works are too harsh on the companies and research institutes who are relying on the data to improve their products and technologies. To summarize, the current data sharing protocols lie on the two extremes: either to share the data entirely which leaks the privacy, or not share any data at all which impede the development of new technologies. A framework that lies in the middle which not only protects the privacy, but also shares enough data for the research institutes to reach certain training/learning performance is highly desirable.

A naive approach for such problem is to only share a fraction (% of the entire dataset) of the dataset, aiming to achieve the same fraction of the goal while protecting most of the privacy. Such method may work well in some contexts such as video codec (i.e., video encoding and decoding). For example, sharing the 90%-compressed version of a video clip should achieve 10% of the original video quality after decoding. However, such approach does not work in the context of machine learning. For example, sharing only 10% of the training dataset doesn't necessarily mean that the model accuracy trained on the 10% data achieves only 10% of

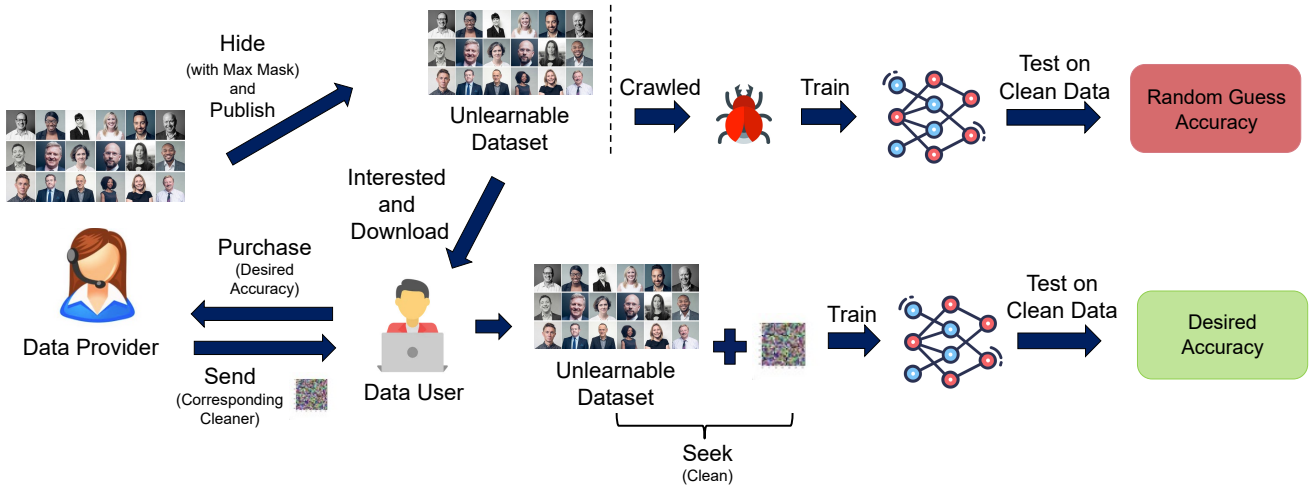

Fig. 2. Our proposed Hide-and-Seek framework protects the data provider's privacy while allow data user to access the shared data needed for model training purposes. The data provider first apply the strongest mask to the dataset (this is the Hide Phase), and then publish the unlearnable dataset (original dataset + strongest mask). (Top) If a web-crawler downloads the published dataset and utilizes it for training purposes, his final label would perform bad on the clean (regular, unmasked) data samples. (Bottom) If a user is interested in the dataset, he needs to download the published dataset, and purchase the desired version (defined by the performance of the final learning outcome). When the data provider receives the payment, he will send the corresponding cleaner patch to the user. The user apply the cleaner to the downloaded dataset (this is the Seek Phase), and train on the resulting dataset to get the desired model performance.

the accuracy from model trained on full dataset. Instead, its accuracy is dependant among other settings as well.

Targeting the above weaknesses in current data sharing protocols in machine learning, in this paper, we propose a novel Hide-and-Seek data sharing framework which provides an option for the data providers to choose other than the difficult "share, or no share" options. The Hide-and-Seek framework could be divided into the Hide Phase and the Seek Phase. In the Hide Phase, the dataset provider first trains different levels (or scales) of masks. Each mask, when patched to the original dataset, hides the features of the original dataset to a desired extent that when the task is trained on the masked dataset, achieves a corresponding testing accuracy. Taking a 10-class classification task as an example, and assume it reaches a certain testing accuracy after fully trained on the original dataset. By carefully designing and training the different levels of masks and apply them on the original dataset, the classification task trained on the masked dataset would achieve the pre-defined testing accuracy between 10% (random guess) and the ideal accuracy (i.e., a mask could be trained that lead to $30\%, 40\%, ...,$etc. testing accuracy) depending on the consumer's need. The strongest mask that lead to the random guess accuracy is equivalent to the "not share" option, as the masked dataset is fully "unlearnable" [5]. And the zero mask leads to original testing accuracy, and is equivalent to the "share" option where no actual mask is applied. It is worth noting that none of the masks (including the strongest one) hides the visual patterns of the data (i.e., human eyes are still able to recognize the original visual patterns of the data).

In the Seek Phase, after getting the multi-scale masks, the data provider computes a set of data-cleaners based the masks. As will be clear shortly, the cleaners are also in multiple scales, and are used to clean the unlearnable (fully masked) dataset. Finally, utilizing the trained masks and cleaners, we aim to solve the problem illustrated in Figure 1 by the Hide-and-Seek data sharing framework shown in Figure 2 and described as follows. The data provider first utilizes the strongest mask to produce the unlearnable dataset (this is the co-called "Hide") and publishes it on a publicly-accessible website without the need to worry about whether the data would be crawled for training purposes. Because even if the dataset is crawled and used for training, the model trained on such unlearnable (masked) dataset would fail on the testing phase (test on clean dataset). Along with the publishing of the unlearnable dataset, the data provider should also set a price table that each version (defined by the test accuracy or learning outcome of the model learned on the dataset) of the dataset should come with a different purchase price. If a user (buyer) is interested in purchasing the dataset, he can make a query to the dataset provider along with the desired level of learning outcome needed. Upon receiving the payment, the data provider sends the corresponding cleaner to the user. The user can then downloads the publicly available dataset, applies the cleaner on it, and starts his task training procedures. The model trained on the cleaned dataset (after applying the cleaner on the unlearnable dataset) will achieve the desired level of testing accuracy (and this is the co-called "Seek").

The main contribution of this work includes the following three folds:

- We propose the Hide-and-Seek multi-scale data sharing framework. Given a dataset, we are able to generate a set of masks that lead to different levels of learning outcomes when trained on the corresponding level of masked datasets. Such masks provide the privacy protection to some extent for data providers while satisfying the needs

of data acquisition from companies/research institutes in order to develop new technologies.

- Enlightened by existing works, we propose a novel method to train the set of masks to fulfill the need of the Hide Phase of the Hide-and-Seek framework. We limit the magnitudes of the masks so that the original visual patterns is still preserved.
- We conduct extensive experiments on multiple datasets including the CIFAR-10, Street View House Number (SVHN) and CIFAR-100 datasets. The experiment results verify the effectiveness of the proposed Hide-and-Seek multi-scale data sharing framework. We also demonstrate examples of the original data samples and their masked data samples in different mask levels, and calculated the Mean Squared Error (MSE) and Structural Similarity Index (SSIM) to show the preservation of the visual patterns.

The rest of this paper is organized as follows. The existing data sharing frameworks and the privacy-preserving techniques in machine learning are reviewed in Section II. We describe the details of the proposed Hide-and-Seek framework in Section III in which the III-B section explains how the different scales of masks are trained (i.e., the Hide Phase), and III-C subsection describes how the corresponding cleaners are computed (i.e., the Seek Phase). The performance evaluation of the Hide-and-Seek data sharing framework is shown in Section IV, and finally we conclude this paper in Section V.

II. Related Works

A. Privacy Protection Methods for Data Sharing

Although there are existing industrial data-sharing platforms such as [7] that strictly restricts the user accessibility in a fine-grained level to achieve the privacy-protection goal, researchers are investigating other possible solutions to protect the data in a more direct way. The current literature that protects the data privacy from being leaked during the sharing phase could be summarized into two categories. The first category is the Reversible Data Hiding (RDH) or the lossless/invertible data hiding frameworks, where data can be embedded into a cover medium for data sharing, and are later extracted by the receiver once received [8] [9]. In real applications, one of the most widely used medium is the compressed JPEG images [10] [11] [12]. This is due to the fact that JPEG offers an effective trade-off by reducing the file size of images while maintaining a satisfactory level of visual fidelity. Another emerging choice to hide the secret data and retrieve it back is to embed it into audio/video clips. [13], [14] and [15] are the representing research works done in this area. They used the inner product between the motion vector and the modulation error, two-dimensional histogram modification and reversible video watermarking to achieve the reversibility, respectively. The disadvantage for such RDH methods is that they suffer from large computational costs to hide even a single data sample, and hence would introduce a huge overhead in the machine learning context, since the latter usually requires a large amount of data to train even a small model.

In the other category, researchers apply the encryption techniques to protect the dataset so that only the intended (user) receiver with the corresponding key is capable of accessing the data. [16] proposes a scheme that utilizes the proxy re-encryption algorithm and oblivious random access memory (ORAM) aiming to ensure the privacy and prevent the traceability in cloud computing. Such method enables multiple users to securely share the data while preserving their privacy. [17] develops an enhanced attribute-based encryption method that combines a personal access policy for users and a professional policy for the fog nodes. Such encryption ensures the effective provision of the medical services. And in [18], the authors present FPDS (Flexible Privacy-Preserving Data Sharing) sheme for cloud assisted IoT in which the data of IoT users are encrypted by an identity-based encryption that ensures the privacy and confidentiality in the phase of data sharing. Similar to the first category where the researchers are applying the RDH to protect the data privacy, adding the encryption for privacy concerns suffer from the same drawbacks that it adds a huge overload in the machine learning context since it usually require a large amount of training data.

B. Privacy-Preserving Machine Learning

Due to a recent observation that machine learning models tend to memorize some information about the training dataset [19] [20] [21], leading to its vulnerability to the privacy attacks [22] [23] such as membership inference attack [24], the privacy-preserving machine learning that not only protects the privacy of the training dataset, but also enables the regular learning process becomes highly desirable. The current literature approaches the privacy-preserving machine learning (PPML) topic from two perspectives: protecting the training dataset, and proposing the privacy-preserving learning/computing algorithms.

The privacy protection of training dataset could be further divided into three groups. In the first group, researchers are applying the anonymization technique on the dataset. [25] proposes a method that provides the k-anonymity in the machine learning algorithms, [26] proposes a method that injects the utility into the anonymized dataset and [27] replaces the original dataset by a surrogate one according to the grouping of the anonymized data. In the second group, different kinds of perturbation are added to the dataset to protect its privacy. [6] and [5] train perturbation to be applied to training dataset so that the common data features in one or more classes are not extractable. And [28] [29] are the two representatives of adding differential privacy noises to the training dataset. The last group involves the encryption of the dataset. [30] and [31] fall into this category, which adds a little more overhead since the decryption procedure is also required at some point.

Unlike those in the first category which focus more on the training dataset, the researchers in the second category focus on the training/computational phase of machine learning. [32] proposes a differentially private stochastic gradient descent algorithm with a modest privacy budget, [33] develops a novel method to train a large recurrent model with user-level

Description	Notation
Dataset Provider	$\mathcal{S}$
Authorized User (Buyer)	$\mathcal{P}$
Dataset (clean, masked, unlearnable)	$\mathcal{D}_c$, $\mathcal{D}_k$, $\mathcal{D}_{k_{max}}$
Dataset (validation, test, cleaned)	$\mathcal{D}_v$, $\mathcal{D}_t$, $\mathcal{D}'_k$
Mask with strength $k(0 \leq k \leq k_{max})$	$\boldsymbol{m}_k$
Desired training (learning) levels	λ_k
Noise boundary	ϵ
Cleaners with scale k	$\boldsymbol{c}_k$

TABLE I
HIDE-AND-SEEK NOTATION TABLE

differential privacy guarantee, and [34] presents an efficient privacy-preserving protocol for neural networks among two non-colluding server with the secure two-party computation (P2C). As GPU is one of the most important computational resources needed for machine learning, [35] introduces CryptGPU which identify several cryptographic methods to enforce the privacy-preserving operations on GPUs. Furthermore, the privacy-preserving machine learning had also been applied in the medical imaging field [36].

With all of the above methods mentioned in this section, none of them is capable of providing the "partial share" or "share by level" feature. Such feature is indeed in demand because it could not only serve as a middle point between extreme decisions of either share or not share, it could also satisfy some of the realistic applications. For example, in the scenarios of transfer learning, only partial training results of the mother model is needed because it would be later fine-tuned on the child's dataset anyway. Embracing the above demand, in this paper, we propose the novel Hide-and-Seek framework that provides the options for the dataset users to purchase the need-based customized dataset and meanwhile protects the data privacy from the data provider's perspective.

III. HIDE-AND-SEEK DATA SHARING FRAMEWORK

In this section, we first present the overall workflow of the Hide-and-Seek protocol in subsection III-A assuming we have already trained the multi-level masks and generated the multi-level cleaners. After that, as the name suggests, our framework could be divided into two phases: the Hide Phase (masks generation) explained in III-B and the Seek Phase (cleaners generation and application) described in III-C. The Hide Phase generates different scales of masks that when applied to the dataset, hide the original dataset to a certain extent, defined by the corresponding learning outcome. In the Seek Phase, we compute a set of cleaners that are capable of recovering the unlearnable (fully masked) dataset to a desired scale (also defined by the corresponding learning performance after training). To enhance the readability of the algorithms, we introduce the notation table in Table I.

A. Hide-and-Seek Protocol

The Hide-and-Seek data sharing framework is summarized in **Protocol 1** and explained as follows. The Hide-and-Seek data sharing protocol starts with the dataset provider $\mathcal{S}$ publishing the unlearnable dataset $\mathcal{D}_{k_{max}}$ by applying the strongest mask $\boldsymbol{m}_{k_{max}}$ on the clean dataset $\mathcal{D}_c$. Notice that none of our masks (including the strongest mask) hides the visual patterns of the real data samples, therefore applying even the strongest mask won't affect the regular usage of the original data (for example, it could still be used as a profile photo,...,etc.). Therefore, taking the advantage of such feature, the potential user is still capable of recognizing the visual patterns of the data, and then decides if such dataset fits his scenarios. Once a user $\mathcal{P}$ decides that he needs this dataset for his work, he makes a query to the dataset provider $\mathcal{S}$ along with his desired level of learning outcome k and the corresponding payment. $\mathcal{S}$ receives the query and payment from user $\mathcal{P}$, and sends the cleaner with the corresponding clean scale, $\mathcal{C}_k$ to the user $\mathcal{P}$. Upon the receiving of the cleaner, $\mathcal{P}$ applies $\mathcal{C}_k$ on the published/downloaded dataset $\mathcal{D}_{k_{max}}$, and get the "kth-level" cleaned dataset, $\mathcal{D}'_k$. He can then starts training his models on $\mathcal{D}'_k$ and gets the "k-th level" learning performance which is equivalent as was trained on $\mathcal{D}_k$ (kth-level masked dataset). And hence the Hide-and-Seek data sharing is completed.

Protocol 1 Hide-and-Seek Data Sharing Protocol

Participants:
One $\mathcal{S}$ (data provider), and multiple $\mathcal{P}$ (authorized users/buyers).
Data Provider's Goal:
1. Share the desired-level (defined by the buyer) of dataset with buyers.
2. Protect the original data from being trained by unauthorized users.
Buyers' Goal:
Gain access (purchase) to the dataset that provide the desired learning outcome.
The protocol:
Step 1: $\mathcal{S}$ publishes the unlearnable dataset ($\mathcal{D}_{k_{max}}$).
For each buyer $\mathcal{P}$:
Step 2: $\mathcal{P}$ reviews $\mathcal{D}_{k_{max}}$, and pays to purchase $\mathcal{D}_k$ by specifying k.
Step 3: $\mathcal{S}$ sends $\boldsymbol{c}_k$ to $\mathcal{P}$.
Step 4: $\mathcal{P}$ downloads $\mathcal{D}_{k_{max}}$ and retrieve $\mathcal{D}'_k$ according to Equation (6).

B. The Hide Phase (Mask Generation)

Inspired by the paper that proposed the Unlearnable Examples [5], we extend the algorithm so that instead of finding the purely unlearnable perturbations, we search for the masks that lead to a desired level of learning outcome.

1) Assumption on Data Provider's Capability: We assume that the data provider has full access to the dataset that they would like to share, but can only manipulate the dataset prior to the sharing phase. After that, the dataset provider has no access during the sharing process or after the sharing is completed. This assumption is made based on the fact that once something is published online, whether it's used and how it's used will be out of the publisher's control. Although we

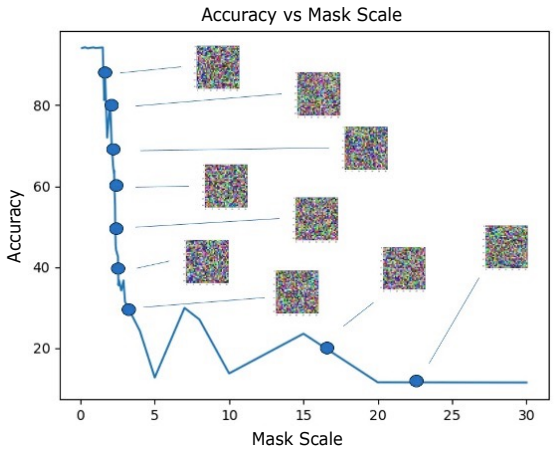

Fig. 3. The illustration of how masks are trained. The blue dots denote the accuracy selected (λ_k) to produce the corresponding masks, and each image denotes the mask $\boldsymbol{m}_k$ produced at the chosen level, λ_k. Notice that in this figure, the masks are just illustrations, and the mask scales are later normalized into an ϵ-ball prior to being applied on the clean dataset.

assume that the dataset provider could no longer change the dataset after the sharing phase, it is realistic to assume that there is a communication channel between the user and the provider, and they are able to exchange information (such as the dataset cleaner) via that channel.

2) Problem Formulation: With the simplicity of the idea explanation, and without the loss of generalizability, we formulate the problem in the context of image classification. For a Z-class classification task, we denote the original (clean) training dataset as $\mathcal{D}_c = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ are the data samples and $y \in \mathcal{Y} = \{0, ..., Z-1\}$ are the labels of the data samples. Z denotes the total number of classes, and n denotes the total number of data samples. The testing dataset is denoted as $\mathcal{D}_t$ which shares the same data distribution as $\mathcal{D}_c$. A typical model training procedure is guided by the objective function

$$\arg\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_c}\mathcal{L}(f(\boldsymbol{x}), y) \tag{1}$$

which learns the mapping from the input space to the label space: $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{L}$ the classification loss function such as the wide-used cross entropy loss.

In the Hide Phase, our goal is to find a list of masks $\boldsymbol{m}_k$'s with different scales k's such that the machine learning model trained on the resulting masked dataset (after applying $\boldsymbol{m}_k$ on $\mathcal{D}_c$), denoted as $\mathcal{D}_k = \{\boldsymbol{x}'_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}'_i = \boldsymbol{x}_i + \boldsymbol{m}_k$, $||\boldsymbol{m}_k|| \leq \epsilon$, would be able to achieve the degradation of the testing performance to a certain level (or scale) λ_k after being tested on $\mathcal{D}_t$. Instead of training with the objective function in Equation (1), our Mask-Generation algorithm is guided by

$$\arg\min_{\theta} \ \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}_k}\mathcal{L}(f(\boldsymbol{x}'), y) \tag{2}$$

and we stop the training process and output $\boldsymbol{m}_k$ once the validation accuracy reaches λ_k. Notice that in this paper, the masks are the class-wise masks, i.e., under a certain mask scale k, there will be in total of Z masks (Z is the total number of classes), and therefore $\boldsymbol{x}'_i = \boldsymbol{x}_i + \boldsymbol{m}_k^{y_i}$, $\boldsymbol{m}_k^{y_i} \in M_k = \{\boldsymbol{m}_k^0, \boldsymbol{m}_k^1, ...\boldsymbol{m}_k^{Z-1}\}$. Data samples in the same class share the same $\boldsymbol{m}_k$. In the extended version of this work, we will explore the sample-wise masks for comparisons.

Given an original data sample $\boldsymbol{x}$, we adopt the same mask generation method as in [5]:

$$\begin{aligned} \arg\min_{\theta} \ & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_c} \min_{\boldsymbol{m}_k}[\mathcal{L}(f'(\boldsymbol{x}+\boldsymbol{m}_k), y)] \\ & \mathbf{s.t.} \quad ||\boldsymbol{m}_k|| \leq \epsilon \end{aligned} \tag{3}$$

which is a bi-level optimization problem that both the inner and outer parts minimize the classification loss. The difference is that the outer part tries to find the parameters θ while the inner part tries to find the mask $\boldsymbol{m}_k$ under the condition that its norm is bounded by ϵ. It is worth noting that the optimization step of θ should be limited to enforce the effectiveness in finding $\boldsymbol{m}_k$ due to the fact that the two parts of the bi-level optimization problem share the same objective (loss minimization). We solve the inner optimization problem with the PGD algorithm [37] as follows:

$$\boldsymbol{x}'_{t+1} = \Pi_\epsilon(\boldsymbol{x}'_t - \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}}\mathcal{L}(f'(\boldsymbol{x}'_t), y))) \tag{4}$$

where $\nabla_{\boldsymbol{x}}\mathcal{L}(f'(\boldsymbol{x}'_t), y)$ denotes the gradient with respect to input $\boldsymbol{x}$, t is the iteration number, and Π_ϵ is the clipping function that enforces the norm bound of $\boldsymbol{m}_k$. The detailed masks generation algorithm is summarized in Algorithm 1.

The intuition of the mask generation process is illustrated in Figure 3. Specifically, as the figure illustrated, a machine learning model that trains on the dataset together with a larger mask tends to learn less of the mapping from the original feature space ($\mathcal{X}$) to the label space, as more of the features are hidden by the larger mask. In experiments, we empirically pick several level of the desired accuracy (λ_k) to stabilized the model on, and then output the masks ($\boldsymbol{m}_k$) at corresponding levels.

After the generation of all desired level of masks, the dataset provider can publish $\mathcal{D}_{k_{max}}$ along with the price table for each level of learning outcome λ_k. Once a user queries to purchase $\mathcal{D}_k$, the data provider will send $\boldsymbol{c}_k$ to the user. We explain the computation of $\boldsymbol{c}_k$ with different levels of k in the next subsection.

C. The Seek Phase (Cleaners Generation and Application)

1) Cleaners Generation: In the previous subsection, the data provider gained a list of masks that is able to hide the original dataset to a certain level. In this subsection, the goal is to generate different levels of dataset cleaners that after applied to the unlearnable dataset, $\mathcal{D}_{k_{max}}$, the resulting dataset $\mathcal{D}'_k$ will achieve the corresponding learning outcome λ_k, as designed.

Given the set of masks $\boldsymbol{m}_k$'s with different hiding scales, we define the mask cleaners with the corresponding cleaning levels as:

$$\boldsymbol{c}_k = \boldsymbol{m}_k - \boldsymbol{m}_{k_{max}} \tag{5}$$

where $\boldsymbol{c}_k$ denotes the dataset cleaner with cleaning level k. As will be clear shortly, such definition of the dataset cleaner is capable of cancelling out the unlearnable mask during the

Algorithm 1 Mask Generation

1: Input: λ_k, and **for each** $\boldsymbol{x} \in \mathcal{D}_c$
2: Initialize $\boldsymbol{m}_k$
3: **for** t=0 ... end of training iterations **do**
4: Train the model on $\boldsymbol{x} + \boldsymbol{m}_k$ according to Equation (3);
5: Test on validation dataset $\mathcal{D}_v$;
6: **if** The result of line 5 is consistently within a threshold with λ_k **then**
7: **break**
8: **else**
9: Update $\boldsymbol{m}_k$ according to Equation (4);
10: continue with line 4;
11: **end if**
12: **end for**
13: $\boldsymbol{m}_k = \boldsymbol{x}' - \boldsymbol{x}$
14: Output: $\boldsymbol{m}_k$

Algorithm 2 Cleaner Generation

1: Input: $\boldsymbol{m}_k$'s trained as in Section III-B, and
2: k from the user.
3: Dataset provider computes $\boldsymbol{c}_k$ with the given k according to Equation 5.
4: Output: $\boldsymbol{c}_k$

cleaner application process from the user's perspective. The cleaner generation algorithm is summarized in Algorithm 2. After receiving the payment from the user, the dataset provider will then send the corresponding $\boldsymbol{c}_k$ to the user, and the user will apply the cleaner on the unlearnable dataset to retrieve the cleaned dataset with the desired learning outcome.

2) Cleaner Application: The user starts the cleaning process by first downloading the publicly available dataset $\mathcal{D}_{k_{max}}$ which is currently unlearnable. The user purchases the dataset cleaner $\boldsymbol{c}_k$ with the desired cleaning scale k, from the dataset provider, and apply it on the unlearnable dataset, $\mathcal{D}_{k_{max}}$ as the following:

$$\begin{aligned} \mathcal{D}'_k &= \mathcal{D}_{k_{max}} + \boldsymbol{c}_k \\ &= \mathcal{D}_c + \boldsymbol{m}_{k_{max}} + \boldsymbol{m}_k - \boldsymbol{m}_{k_{max}} \\ &= \mathcal{D}_c + \boldsymbol{m}_k \\ &= \mathcal{D}_k \end{aligned} \tag{6}$$

where $\mathcal{D}'_k$ is the dataset that the user retrieved after applying the dataset cleaner. From Equation (6) is could be seen that it is equivalent to $\mathcal{D}_k$ the originally masked dataset which achieves the desired level of learning outcome. And the Seek Phase is finished once the user recovers the unlearnable dataset to the desired level of learning outcome.

We show the effectiveness of the mask generation, cleaner generation and cleaner application algorithms in Section IV.

IV. Experiments

A. Experimental Setup

We conduct extensive experiments to verify the effectiveness and generalizability of the proposed Hide-and-Seek data

Algorithm 3 Cleaner Application

1: Input: $\mathcal{D}_{k_{max}}$ published by dataset provider, and
2: $\boldsymbol{c}_k$ purchased.
3: User downloads $\mathcal{D}_{k_{max}}$.
4: User receives $\boldsymbol{c}_k$ and apply it on $\mathcal{D}_{k_{max}}$ according to Equation 6.
5: Output: $\mathcal{D}_k$

sharing framework on the CIFAR-10 [38], Street View House Number (SVHN) [39], and the CIFAR-100 [38] datasets. We train our framework on ResNet [40] on Nvidia RTX 4090 GPUs.

We demonstrate the effectiveness of the Hide (mask generation) Phase and the Seek (cleaner generation and application) Phase in sections IV-B and IV-C, respectively. In section IV-D, we plot the original data sample, together with the corresponding different-level-masked version of the same data samples. Such plots illustrate that our masks are capable of preserving the original visual features on the head of protecting the privacy of the dataset, and hence enables the regular usage of the original data sample, such as the usage as a profile photo.

B. Validation on Effectiveness of Masks

We verify the effectiveness of the masks trained (as described in Section III-B) by testing whether the machine learning model trained on each masked dataset would achieve the corresponding testing accuracy. Figure 4 demonstrates the effectiveness of our masks on CIFAR-10, SVHN and CIFAR-100 datasets. It's worth noting that an ambiguity of Figure 4 is that it seems like each sub-figure is showing a single training process, however, this is not the way it is. Instead, we train each mask in a separate training process, and stop to save (output) the mask until the testing accuracy of the model trained on the masked dataset stabilizes at the desired (pre-selected) level. One example of the sequence of trained masks for different accuracy levels for the "Cat" class in the CIFAR-10 dataset is shown in the first and third rows of Figure 6. We then plot in Figure 4 each level of mask versus the test accuracy achieved on the corresponding level of the masked dataset.

It could be observed that with the increased scales of masks, the testing accuracy of the model trained on the masked dataset is dropped. However, the rate of such dropping of accuracy is not constant. In the beginning and at the end of the plots, such drop rate is smaller, reflecting the smaller accuracy change with the increase of the mask scales. In the middle part, however, the drop rate is steep, suggesting that even a small change of the mask scale could lead to a relatively larger accuracy drop. Due to such a steep change of the learning accuracy, occasionally it's not feasible to find a mask that stabilizes a learning model at certain accuracy. An example of such scenario is that in the SVHN dataset, we aren't able to get the masks that stabilizes the learning outcome around 70% and 80% (as shown in Figure 5 in SVHN). Note that in

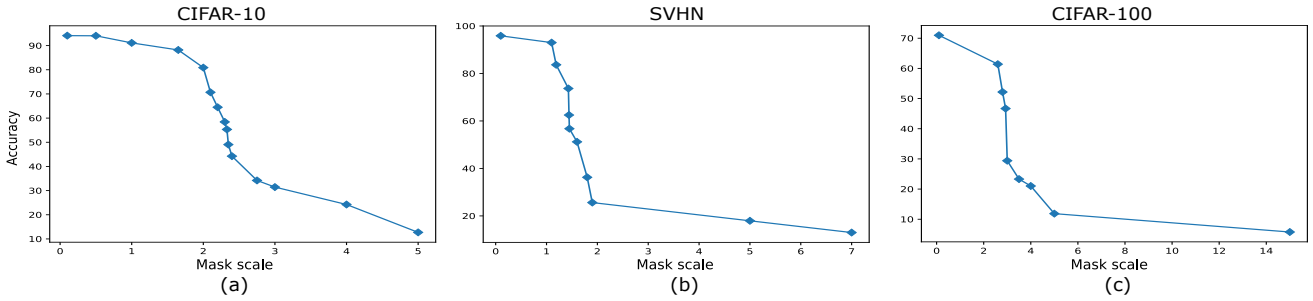

Fig. 4. Validation on the Effectiveness of Masks on the CIFAR-10, SVHN, and CIFAR-100 datasets, respectively. The line graph depicts the accuracy during different scales of mask training.

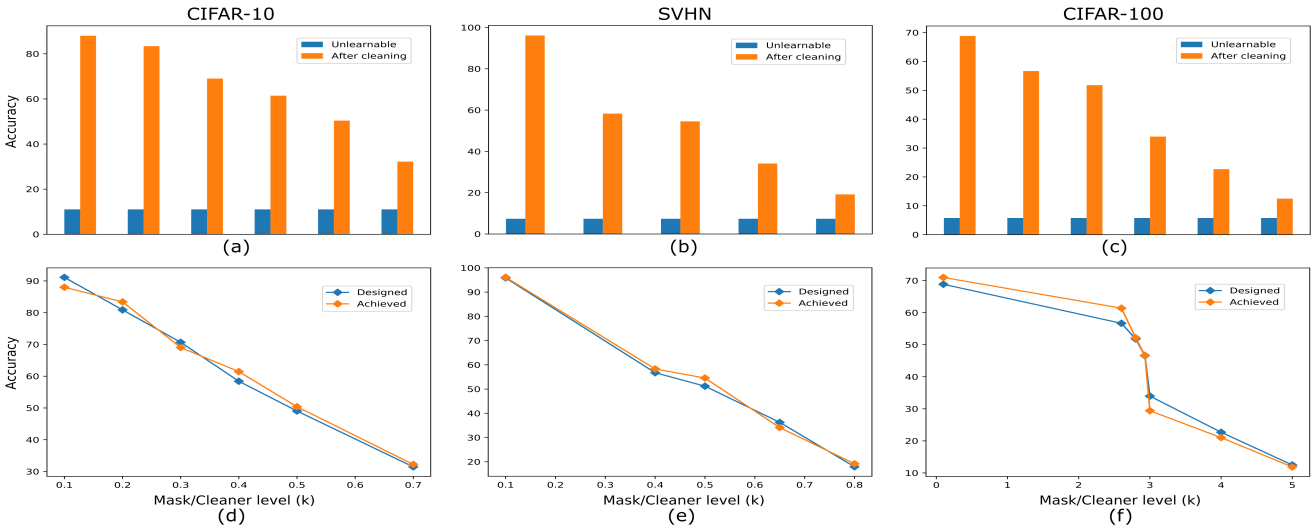

Fig. 5. Validation on the Effectiveness of Cleaners. Top: the model accuracy achieved (orange bars) after each scale of cleaner is applied to the unlearnable dataset (blue bars) on CIFAR-10, SVHN, and CIFAR-100 datasets, respectively. Bottom: The designed level of accuracy that the masks are generated on (blue), and the learning accuracy achieved after training on the corresponding level of cleaned datasets (orange) on CIFAR-10, SVHN, and CIFAR-100 datasets, respectively.

the extensive experiments we have conducted, such scenario only happens for the SVHN dataset between the 70% and 80% accuracy range (a.k.a. black-out range), and hence should be considered as a rare event. Due to the rarity of this scenario, in practice the issue can be simply by-passed by not including the related black-out accuracy range in the price table offered to the data users.

Another interesting fact is that with the increase of the difficulty/complexity of the dataset, the difficulty to find the masks that stabilize the learning performance also increases. This is intuitive as the training on a difficult dataset usually has higher variance and will be more sensitive to the perturbations.

C. Validation on Effectiveness of Cleaners

We verify the effectiveness of our multi-scale cleaners computed in the Seek Phase (Section III-C). Figure 5 demonstrates the cleaning results for the CIFAR-10, SVHN, and CIFAR-100 datasets, respectively (from left to right). The sub-figures in the top row (sub-figures (a)-(c)) compare the learning outcome (testing accuracy) between the models trained on the unlearnable dataset (blue bars) and on the cleaned dataset with different cleaning level (scale). It could be observed that in all of the three datasets, our multi-scale cleaners are capable of recovering (or seeking) the unlearnable dataset to different levels.

Furthermore, validating the effectiveness of the dataset cleaners alone is not enough. It is also important for the users (buyers) to verify that the learning outcome on the cleaned dataset matches the original desired performance. To this end, we compare the designed testing accuracy and the achieved accuracy on the bottom row of Figure 5 (sub-figures (d)-(f)). The blue curves denote the testing accuracy of the models trained on the masked datasets, $\mathcal{D}_k$ in different masking level, and the orange curves demonstrates the testing accuracy of the models trained on the cleaned datasets, $\mathcal{D}'_k$, also at the corresponding cleaning levels. It could be observed that the blue and orange curves are very close to each other, indicating that the learning outcomes of the cleaned datasets is very similar to the testing accuracy of the masked datasets. Hence, we complete the validation on the effectiveness of our dataset cleaners, i.e., it could not only recover the unlearnable dataset, but also to the levels as they were designed (or per the requests from the users).

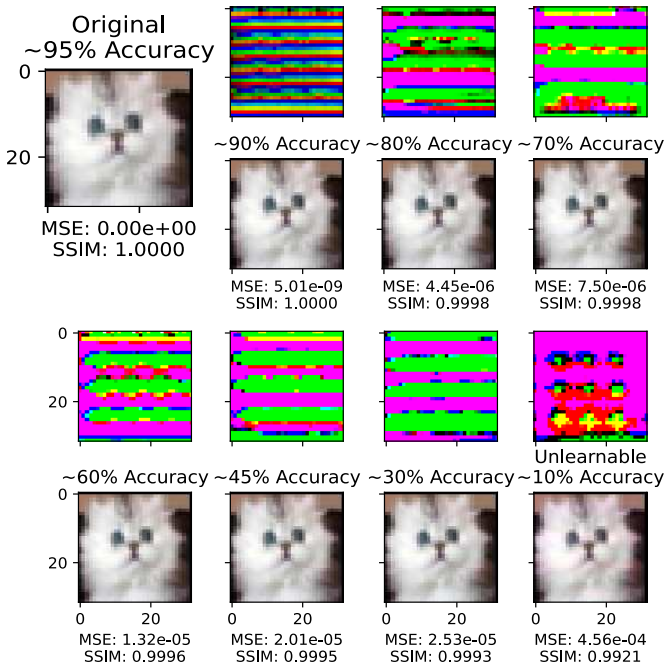

Fig. 6. The Mean Squared Error (MSE) and Structural Similarity Index (SSIM) computed between the original data sample (top left sub-figure), and different levels of masked data samples (2nd and 4th rows) after the corresponding masks (1st and 3rd rows) are applied. The original data sample is picked from class "cat" in the CIFAR-10 Dataset.

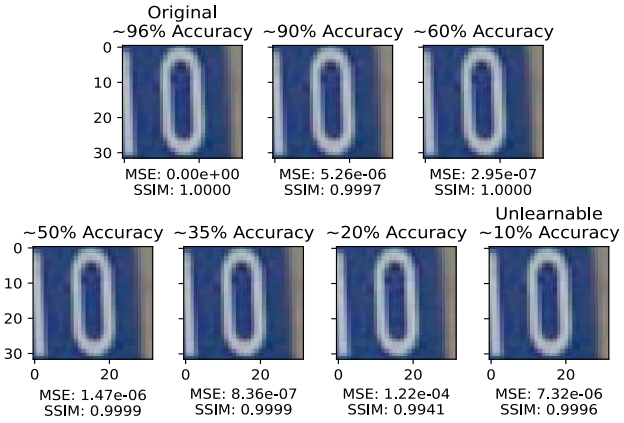

Fig. 7. The MSE and SSIM computed between the original data sample (top left sub-figure), and different levels of masked data samples (all other sub-figures). The original data sample is picked from class "0" in the SVHN Dataset.

D. Validation on the Consistency of Visual Patterns

In this subsection, we show that the masked dataset still preserves the visual patterns of the original dataset. We pick a sample from each of the CIFAR-10, SVHN and the CIFAR-100 dataset, and plot the original samples and their corresponding multi-scaled masked versions in Figures 6, 7 and 8, respectively. It could be seen by human eyes that all masked versions (although with different masking levels) are capable of preserving the original visual patterns (i.e., all masked cat images from the CIFAR-10 dataset are still the same cat, and same for the number 0 image in the SVHN dataset and the

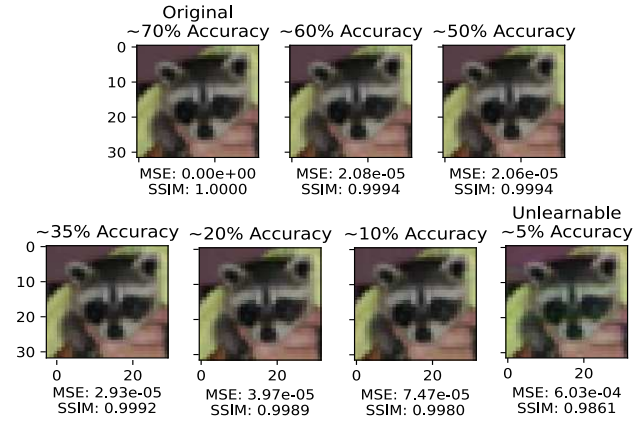

Fig. 8. The MSE and SSIM computed between the original data sample (top left sub-figure), and different levels of masked data samples (all other sub-figures). The original data sample is picked from class "caccoon" in the CIFAR-100 Dataset.

caccoon image in the CIFAR-100 dataset).

To empirically verify the observation, we calculate the Mean Squared Error (MSE) and the Structural Similarity Index (SSIM) scores between the original sample and each masked samples, as these two metrics are well-known to measure the similarities between two images. The MSE and SSIM scores are listed under the corresponding sub-figure, respectively. It could be observed that all the MSE scores are very small ($\leq 10e^{-3}$ for all three datasets), indicating that all the masked data samples are very similar with the original data. It could also be observed that with the increase of the masking level, the MSE increases by a small scale. This suggests that stronger masks does cause a bigger distortion of the original data sample, but such distortion is too small to be concerned.

Furthermore, the SSIM scores indicate the same observation. The SSIM scores are all very high (≥ 0.99 for all three datasets), indicating that all masked samples resembles the structural patterns of the original data. And similarly, although the SSIM score decreases with the increase of mask level, such decrease is too small to change the visual features of the original data. And hence we finish the verification that our masks are capable of preserving the visual patterns of the original datasets.

V. Conclusion

In this paper, we propose a novel Hide-and-Seek data sharing framework that is able to protect the data privacy for data providers and enables the desired level (defined by the user) of sharing for the training purposes in the context of machine learning. Such flexible framework bridges the gap between the current "either completely share or not share at all" extreme decisions in the current machine learning field. Following the Hide-and-Seek framework, the data provider first publishes the unlearnable (strongest-level-masked) dataset, and sends the corresponding dataset cleaners only to the users (buyers) who query with the payment. Upon receiving the dataset cleaner, the user can apply it on the downloaded unlearnable dataset, and retrieve the cleaned dataset at the corresponding cleaning

level. The machine learning task, after being trained on the cleaned dataset, would achieve the desired level of learning outcome on the testing dataset.

ACKNOWLEDGEMENT

This work is supported in part by the United States National Science Foundation (NSF) under grants CNS-2308761 and CNS-2006998. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[2] V. U. Prabhu and A. Birhane, "Large image datasets: A pyrrhic win for computer vision?," *arXiv preprint arXiv:2006.16923*, 2020.

[3] K. Hill, "The secretive company that might end privacy as we know it," *https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html*, 2020.

[4] F. T. Commission, "Ftc imposes $5 billion penalty and sweeping new privacy restrictions on facebook," *https://www.ftc.gov/news-events/news/press-releases/2019/07/ftc-imposes-5-billion-penalty-sweeping-new-privacy-restrictions-facebook*, 2019.

[5] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," *arXiv preprint arXiv:2101.04898*, 2021.

[6] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting privacy against unauthorized deep learning models," in *29th USENIX security symposium (USENIX Security 20)*, pp. 1589–1604, 2020.

[7] V. Sistla and S. Chiplunkar, "Patterns for enterprise data sharing at scale," *https://aws.amazon.com/blogs/big-data/patterns-for-enterprise-data-sharing-at-scale/*, 2023.

[8] R. Caldelli, F. Filippini, and R. Becarelli, "Reversible watermarking techniques: An overview and a classification," *EURASIP Journal on Information Security*, vol. 2010, pp. 1–19, 2010.

[9] F. Peng, Y.-z. Lei, M. Long, and X.-m. Sun, "A reversible watermarking scheme for two-dimensional cad engineering graphics based on improved difference expansion," *Computer-Aided Design*, vol. 43, no. 8, pp. 1018–1024, 2011.

[10] G. Xuan, Y. Q. Shi, Z. Ni, P. Chai, X. Cui, and X. Tong, "Reversible data hiding for jpeg images based on histogram pairs," in *Image Analysis and Recognition: 4th International Conference, ICIAR 2007, Montreal, Canada, August 22-24, 2007. Proceedings 4*, pp. 715–727, Springer, 2007.

[11] H. Sakai, M. Kuribayashi, and M. Morii, "Adaptive reversible data hiding for jpeg images," in *2008 International Symposium on Information Theory and Its Applications*, pp. 1–6, 2008.

[12] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication watermark for jpeg images," in *Proceedings International Conference on Information Technology: Coding and Computing*, pp. 223–227, 2001.

[13] G. Song, Z. Li, J. Zhao, J. Hu, and H. Tu, "A reversible video steganography algorithm for mvc based on motion vector," *Multimedia Tools and Applications*, vol. 74, pp. 3759–3782, 2015.

[14] J. Zhao, Z.-T. Li, and B. Feng, "A novel two-dimensional histogram modification for reversible data embedding into stereo h. 264 video," *Multimedia Tools and Applications*, vol. 75, pp. 5959–5980, 2016.

[15] C. Vural and B. Baraklı, "Reversible video watermarking using motion-compensated frame interpolation error expansion," *Signal, Image and Video Processing*, vol. 9, pp. 1613–1623, 2015.

[16] J. Shen, H. Yang, P. Vijayakumar, and N. Kumar, "A privacy-preserving and untraceable group data sharing scheme in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2198–2210, 2021.

[17] W. Tang, J. Ren, K. Zhang, D. Zhang, Y. Zhang, and X. Shen, "Efficient and privacy-preserving fog-assisted health data sharing scheme," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–23, 2019.

[18] H. Deng, Z. Qin, L. Sha, and H. Yin, "A flexible privacy-preserving data sharing scheme in cloud-assisted iot," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11601–11611, 2020.

[19] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.

[20] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pp. 587–601, 2017.

[21] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[22] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

[23] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

[24] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18, IEEE, 2017.

[25] A. Friedman, R. Wolff, and A. Schuster, "Providing k-anonymity in data mining," *The VLDB Journal*, vol. 17, pp. 789–804, 2008.

[26] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pp. 217–228, 2006.

[27] M. Yang, L. Song, J. Xu, C. Li, and G. Tan, "The tradeoff between privacy and accuracy in anomaly detection using federated xgboost," *arXiv preprint arXiv:1907.07157*, 2019.

[28] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, IEEE, 2010.

[29] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*, pp. 201–210, PMLR, 2016.

[31] K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi, "Towards deep neural network training on encrypted data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[32] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

[33] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[34] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)*, pp. 19–38, IEEE, 2017.

[35] S. Tan, B. Knott, Y. Tian, and D. J. Wu, "Cryptgpu: Fast privacy-preserving machine learning on the gpu," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1021–1038, IEEE, 2021.

[36] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[37] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[38] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.