A Survey of Model Inversion Attacks and Countermeasures*

Junzhe Song, D.E. Namiot

Lomonosov Moscow State University, Moscow, Russia

Abstract. This article provides a detailed overview of the so-called Model Inversion(MI) attacks. These attacks aim at Machine-Learning-as-a-Service (MLaaS) platforms, and the goal is to use some well-prepared adversarial samples to attack target models and gain sensitive information from ML models, such as items from the dataset on which ML model was trained or ML model's parameters. This kind of attack now becomes an enormous threat to ML models, therefore, it is necessary to research this attack, understand how it will affect ML models, and based on this knowledge, we can propose some strategies that may improve the robustness of ML models.

Keywords: *adversarial machine learning, model inversion attack, deep-learning, cyber security.*

DOI: 10.14357/20790279230110

Introduction

With the development of Machine Learning(ML) and the increase of data size, a large number of ML models have been created and utilized in many parts of human life, we define these models as *Machine-Learning as a service(MLaaS)*. Among these models, there is a kind of models that utilize personal privacy-sensitive information as training data and provide services, such as medical applications(analyze one's medical index and provide medical suggestions) and facial recognition applications(analyze given figure and return name & confidence-value).

Since there are many privacy-sensitive values stored in these models, there are also many malicious attackers who try to gain this privacy-sensitive information from ML datasets (usually we call these attackers *Adversaries*). Therefore, privacy risk becomes an important part of ML models, researchers and MLaaS providers start to try to utilize ML to preserve this privacy-sensitive information from adversary attacks, which creates a new section of ML called: Privacy-Preserving Machine Learning(PPML).

Model Inversion(MI) attack is one of the methods in PPML, which inverts the process of training data into an ML model. The threat, in this case, is potentially exposing data from the training set, which can contain private information, to the adversary. Although there are many types of MI attack methods that have been created, the main issue is: MI attack is unaware of the victims.

Unfortunately, the existing countermeasures can only defend against the corresponding attack, which means, to improve the ML model's robustness, we have to apply several countermeasures simultaneously, this is also a passive approach because we can't preserve our model from unrevealed attacks. Since to find out if our model is under attack as soon as possible, maybe we can create a MI attack detector.

For this detector, we hope we can develop one which can detect not only existing attack methods but also unrevealed attack methods or behavior that act like MI attacks. This will be hard work because attack methods can upgrade very quickly and their features are various. In section *Summary and Future Works*, we discuss some attributes that the detector should possess.

This research has been supported by the Interdisciplinary Scientific and Educational School of Moscow University "Brain, Cognitive Systems, Artificial Intelligence".

1. The Model Inversion Attacks and countermeasures

1.1. The Model Inversion(MI) attacks

The Model Inversion(MI) attacks are mainly aimed at the currently popular MLaaS service model, it refers to an attacker extracting information related to training data from the model prediction results.

A simple deployment of a model inversion attack is presented in Fig.1.

To perform a model inversion attack, the adversarial user will be based on his knowledge of the target

* The work was carried out as part of a project to create a trusted environment for the development of artificial intelligence systems [39][40].

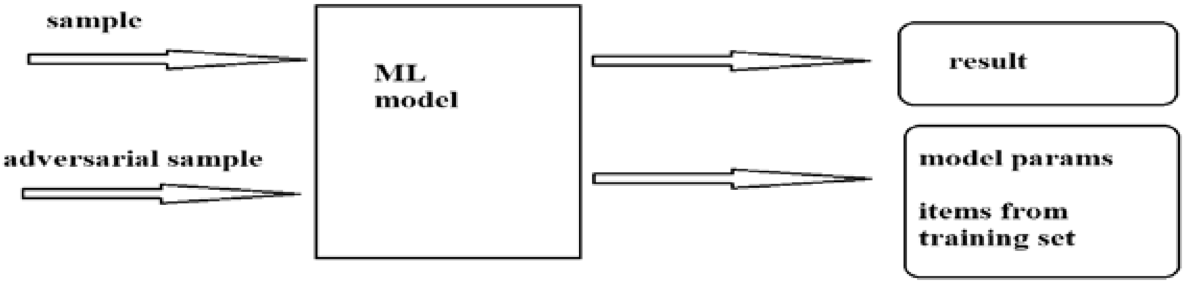


Fig. 1. How adversarial users use ML model

model (0 knowledge - black box, has some knowledge – grey box or white box), and use some well-prepared samples(called adversarial samples) to attack model. As result, adversarial users can get the model's parameters or some samples that are used to train this model.

1.2. Why do we research this kind of attack?

As we said in the Introduction, model inversion attack becomes a huge threat to the ML model, and so far we don't have a mature defensive strategy that can defend against several existing model inversion attacks. So, for the safety and robustness of ML models, we have to learn model inversion attacks, trying to understand how this kind of attack affects the ML model. Then, based on this information and acknowledgment, we might be able to create some effective countermeasures or improve our ML models.

More mathematical definition for MI attack

In [1] has a more mathematical definition of MI attack:

Theorem 1. MI attack is a machine learning problem and specified as a triple (*Z, H, l*), (*Z*, a sample space; *H*, hypothesis space; *l: H×Z -> R*, a loss function) with the following notations:

1) Γ: A training algorithm of the learning problem, which outputs a hypothesis Γ(S) ∊ H on an input training set S.
2) D_S : A distribution over the training set S.
3) τ: The objective function computed by the adversary. For now, one can view it simply as some function that maps *Z* to $\{0, 1\}^*$.
4) gen, sgen: Auxiliary information generators. They map a pair (S, z) to an advice string in $\{0, 1\}^*$.

The MI-attack world is described by a tuple (A, gen, τ, S, D_S, Γ), where the adversary (A) is a probabilistic oracle machine. The following game is played between the Nature and the Adversary A.

$$\text{gain}\,(A, \text{gen}, \tau, S, D_S, \Gamma) = \Pr[A^{\Gamma(S)}(\text{gen}(S, z))= \tau(z)] \quad (1)$$

where the probability is taken over the randomness of z~ D_S, the randomness of gen, and the randomness of A.

The simulated world is described by a tuple (A^*, sgen, τ, S, D_S), where the adversary (A^*) is a non-oracle machine and sgen is the second auxiliary information generator. The game between the Nature and A^* is:

$$\text{sgain}(A^*, \text{sgen}, \tau, S, D_S) = \Pr[A^* (\text{sgen}(S,z))= \tau(z)] \quad (2)$$

where the probability is taken over the randomness of z~ D_S, the randomness of sgen, and the randomness of A^*.

1.3. Classification of MI attacks

For the needs of taxonomy, typically, we can classify client-side access as being either *black-box* or *white-box*. In a *black-box* setting, an adversarial customer will create prediction queries against a model, however not transfer the model description. In a *white-box* setting, an adversarial customer is allowed to transfer an outline of the model.

1.4. Attacks & countermeasures

The Fredrikson et al. attack Widely accepted, the first MI attack was realized by *Fredrikson et al.* in [1]. The *Fredrikson et al. attack* is to use auxiliary information and confidence value given by the model prediction to guess the true value of the privacy-sensitive feature. The weakness of the attack is also clearly: 1) adversary knows the target feature(for example, its domain), a large domain means more combinations adversary have to try; 2) adversary have to know as much auxiliary information as he can, same reason as 1), less information about the victim means more combinations to try; 3) adversary has infinite access to the model because he has to correct the guessing value with the help of confidence value, a higher confidence value means our guessing value is closer to the true value. In a word, the *Fredrikson et al. attack* can realize only under many restrictions.

The author gives countermeasures for both decision trees and facial recognition. For decision trees, the level at which the sensitive feature occurs may affect the accuracy of the attack, and it may be possible to design more sophisticated training algorithms that incorporate model inversion metrics into the splitting criteria to achieve resistance to attacks without unduly sacrificing accuracy.

For facial recognition, one possible defense is to degrade the quality or precision of the gradient information retrievable from the model, and also, black-box facial recognition models can produce confidence scores that are useful for many purposes while remaining resistant to reconstruction attacks.

MI attack for deep networks *Fredrikson et al.* established that model inversion attacks include a smart performance on decision tree and face recognition [1]. However, for deep networks, these procedures sometimes cause unidentifiable representations that square measure useless for the opponent [4]. So that they introduced a more realistic definition of model inversion and leveraged properties of generative adversarial networks for constructing a connected lower-dimensional manifold.

MI attack: wherever the opponent is attentive to the final purpose of the attacked model (for instance, whether or not it's an associate degree OCR system or an automatic face recognition system), and the goal is to seek out realistic category representations among the corresponding lower-dimensional manifold (of, separately, general symbols or general faces).

In [4], the approach is based on Generative Adversarial Network. A Generative Adversarial Network(GAN) is a min-max game between two neural networks: generator (G_θ) and discriminator (D_φ). The generator (G_θ) takes random noise *z* as input and generates ($G_\theta(z)$). The discriminator (D_φ) distinguishes between real samples *x* and fake samples coming from (G_θ). The objective function for the min-max game between (G_θ) and (D_φ) is:

$$min_\theta max_\phi \mathrm{E}_{x\sim P(x)}\left[\log\left(D_\phi(x)\right)\right] + \\ +\mathrm{E}_{z\sim P(z)}\left[1-\log\left(D_\phi(G_\theta(z))\right)\right] \qquad (3)$$

where P_x is the real data distribution, and P_z is a noise distribution which is typically a uniform distribution or a normal distribution.

Obviously, different images should belong to their disconnected manifolds without ant paths of "blended" images between them. However, in GAN, the generator function maps from a connected distribution space to all possible outputs, which results in a connected output set of instances. This is an emblematical disadvantage of GANs and various techniques to partition the input into disjoint support sets have been used to address this issue. And [4]'s approach is to leverage this drawback to search in the low-dimensional space P_x (real data distribution) of all possible images.

With some natural knowledge about the underlying target system, an attacker can use this GAN-based approach for retrieving representative and recognizable samples of individual classes. For the countermeasures, the author suggests that a security-based biometric identification system might classify away the larger set of faces so that the faces that are relevant to security verification are effectively hidden sort like a needle in a very rick. The key downside here is to take care of adequate classifier accuracy because the variety of categories will increase.

Also, in the conclusion, the author proposes a prospective research direction is to consider ways to develop a robust defense against model inversion attacks without affecting the model accuracy. This could be difficult since model inversion doesn't involve protecting any specific instance, and the defense should protect all the representative pictures that are part of the manifold used for training.

Generative Model-Inversion Attack For deep neural networks, there is another attack method that uses GAN. In [6], the author presents a novel attack method, termed the generative model-inversion attack, which can reverse deep neural networks with high success rates. Rather than reconstructing private training data from scratch, the author leverage partial public information, which can be generic, to find out a distributional prior via generative adversarial networks (GANs) and use it to guide the inversion method (Fig.2). The author also shows that differential priva-

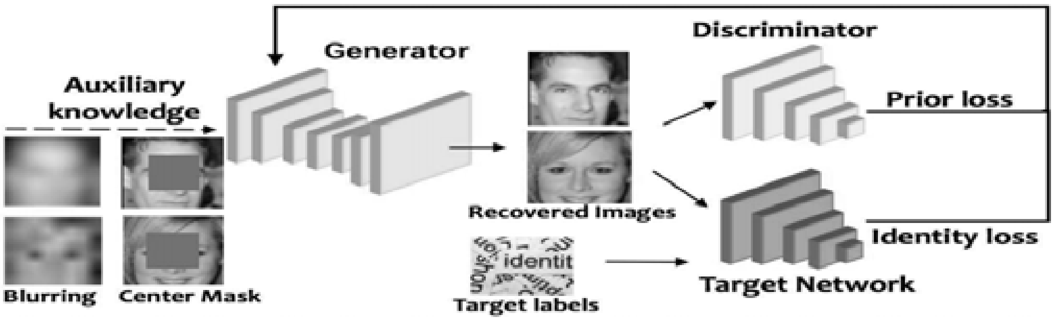


Fig. 2. Overview of the proposed GMI attack method [6]

cy, in its canonical form, is of little avail to defend against their attack.

In this paper, the author focuses on the white-box setting, where the adversary is assumed to have access to the target network *f* and employs some inference technique to discover the options *x* related to a particular label *y*. To recover those images, the reconstruction process consists of two-stage: (1) Public knowledge distillation, which trains the generator and therefore the discriminators on public datasets to encourage the generator to get realistic-looking pictures. The public datasets may be untagged and don't have any identity overlapping with the personal dataset;(2) Secret revelation, in which create use of the generator obtained from the primary stage associated solve an optimization drawback to recover the missing sensitive regions in a picture. In stage 1, the author leverage the canonical Wasserstein-GAN training loss:

$$min_G max_D L_{wgan}(G, D) = \\ E_x[D(x)] - E_z[D(G(z))] \tag{4}$$

when the auxiliary knowledge(blurred or corrupted version of the private image) is available to the attacker, let the generator take the auxiliary knowledge as an additional input. In the second stage, the author solves the following optimization to find the latent vector that generates an image achieving the maximum likelihood under the target network while remaining realistic:

$$\hat{z} = argmin_z L_{prior}(z) + \lambda_i L_{id}(z) \tag{5}$$

where the prior loss L_{prior} (z) penalizes surrealistic pictures and therefore the identity loss L_{id}(z) encourages the generated pictures to own high chance below the targeted network. L_{prior}(z)and L_{id}(z) are defined as follow:

$$L_{prior}(z) = -D(G(z)) \tag{6}$$

$$L_{id}(z) = -log[C(G(z))] \tag{7}$$

where C(G(z)) represents the probability of G(z) output by the target network. The experiments show that this GMI attack has a better performance than the *Fredrikson et al.* attack, which means it is an effective attack method. There are no countermeasures to this GMI from the author.

MI attack that using explanations As the ML models are widely used, people need not only answers from the ML model but also explanations. Explainable artificial intelligence (XAI) provides additional info to assist users to grasp model selections, however further information exposes additional risks for privacy attacks [ref5]. In this paper, the author studies this risk for image-based model inversion attacks and identified several attack architectures with increasing performance to reconstruct private image data from model explanations. And these XAI-aware inversion models were designed to use spatial data in image explanations.

Fig. 3 presents architectures of inversion attack models:

Here, a) Baseline threat model with target CNN model M_t to predict emotion $\tilde{y}_t$ from face *x*, and inversion attack model to reconstruct face $\hat{x}_r$ from emotion. Emotion prediction confidences are input to a transposed CNN (TCNN) for inversion attack (d). b) Threat model with explainable target model that also provides instance explanation $\widetilde{E_t}$ of the target prediction, and XAI-aware multi-modal inversion attack model that inputs $\widetilde{E_t}$ via different input architectures: e) Flattened $\widetilde{E_t}$ concatenated with $\tilde{y}_t$, f) U-Net for dimensionality reduction and spatial knowledge, g) combined Flatten and U-Net. c) Threat model with non-explainable target model and inversion attack model that predicts a reconstructed surrogate explanation $\widetilde{E_r}$ from target prediction $\tilde{y}_t$ and uses $\widetilde{E_r}$ for multi-modal image inversion (e-g). Flattened $\widetilde{E_t}$ concatenated with $\tilde{y}_t$, f) U-Net for dimensionality reduction and spatial knowledge, g) combined Flatten and U-Net. c) Threat model with non-explainable target model and inversion attack model that predicts a reconstructed surrogate explanation $\widetilde{E_r}$ from target prediction $\tilde{y}_t$ and uses $\widetilde{E_r}$ for multi-modal image inversion (e-g).

The author divided MI attacks into 3 types: 1) model inversion with Target Explanations; 2) model inversion with Multiple Explanations; 3) model inversion with Surrogate Explanations. For type 1, the author trained the inversion attack model as a Transposed CNN(TCNN) to predict a 2D image from the 1D target prediction vector as input to the attack model. The model is trained with MSE. For type 2, the author exploits Alternative CAMs(Σ-CAM) by concatenating explanations for |*C*| classes into a 3D tensor and training the inversion models on this instead of the 2D matrix of a single explanation. There is no information about countermeasures.

An inversion-specific GAN for MI attack In the paper [7], the author presents a novel inversion-specific GAN that can better distill knowledge useful for performing attacks on private models from public data. In particular, the discriminator is trained to differentiate not only the real and fake samples but the soft labels provided by the target model. Experiments show that the combination of these techniques can significantly boost the success rate of the state-of-the-art MI attacks by 150%, and generalize better to a variety of datasets and models.

Author focus on white-box setting MI attack. The goal of the attacker is to discover a representative

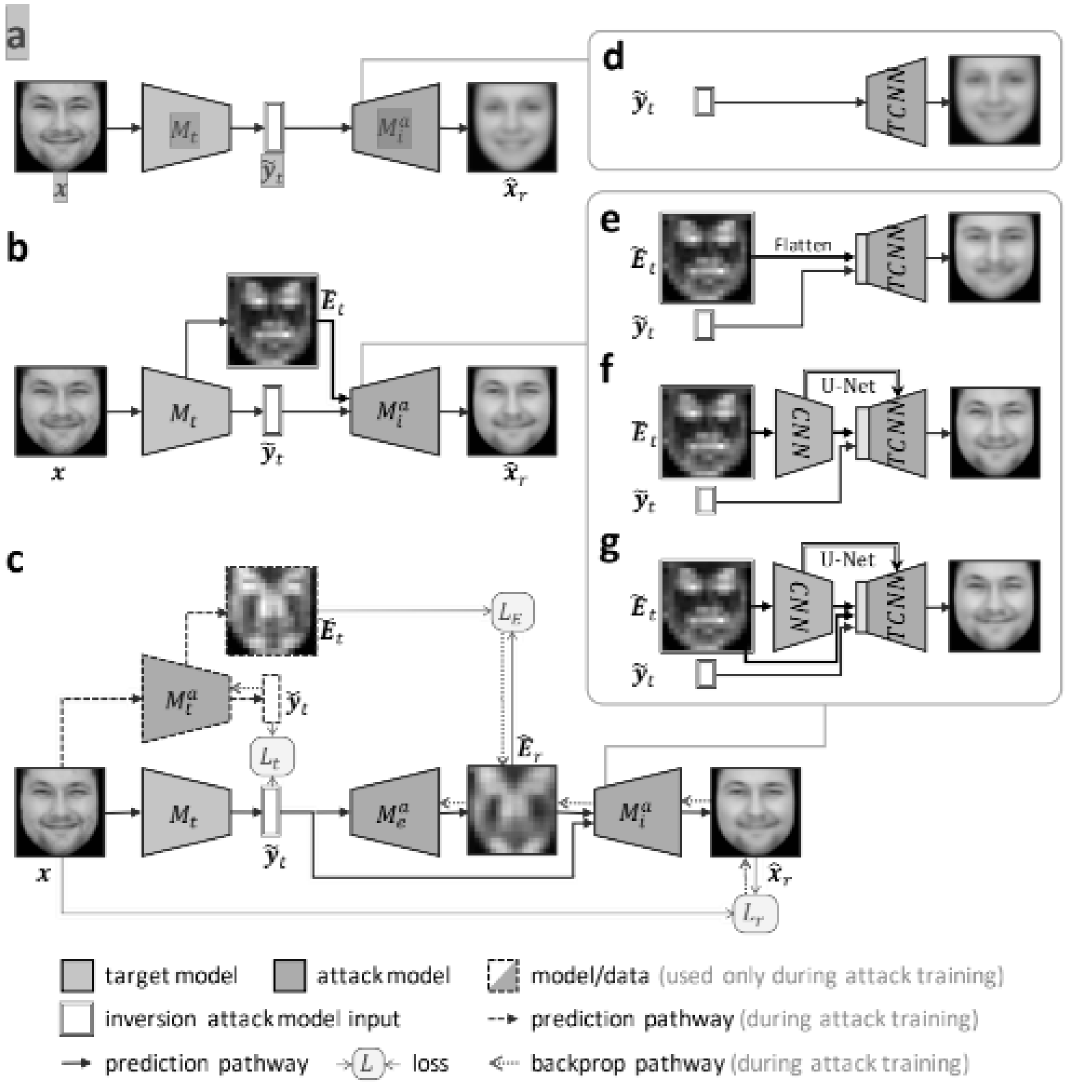


Fig. 3. Architectures of inversion attack models [5]

input feature x associated with a specific label *y*. The author uses facial recognition as a running example for the target network. The proposed attack algorithm consists of two steps. The first step is to train a GAN to have information concerning the personal categories of the target model from public knowledge. Rather than training a generic GAN, the author customizes the training objective for each generator and discriminator thus on higher distill the private-domain data concerning the target model from public knowledge. In the second step, the author makes use of the generator learned in the first step to estimate the parameters of the private data distribution.

Author tested this threat model on several datasets with the baseline. Experiments show that this approach can significantly improve the performance of GMI on all target models. Countermeasures are not mentioned in this paper.

MI attack against collaborative inference Most studies solely targeted knowledge privacy throughout training and neglected privacy throughout illation. During this paper [8], the author devises a brand new set of attacks to compromise inference data privacy in cooperative deep learning systems. Specifically, once a deep neural network and also the corresponding illation task are split and distributed to completely different participants, one malicious participant has the ability to accurately recover any input fed into this system, although he has no access to different participants' information or computations, or to prediction APIs to query this system.

Author considers a collaborative inference system between two participants, P_1 and P_2. The target model is split into two parts: $f_\theta = f_{\theta 2} \cdot f_{\theta 1}$. P_1 performs earlier layers $f_{\theta 1}$, and P_2 performs $f_{\theta 2}$. P_1 is trusted and P_2 is untrusted.

Algorithm 1 White-box model inversion attack

```
function WhiteBoxAttack ($f_{01}, f_{01}(x_0), T, \lambda, \varepsilon$)
/* $f_{01}$ – the target model */
/* $f_{01}(x_0)$ – the intermediate output of sensitive input x0 */
/* $T$ - maximum number of iterations */
/* $\lambda$ – tradeoff between prior and posterior information */
/* $\varepsilon$ – step size if GD */

$L(x) = \|f_{01}(x) - f_{01}(x_0)\|_2^2 + \lambda TV(x)$
$t$=0
$x^{(0)}$ = ConstantInit()
    while (t<T) do
        $x^{t+1} = x^t - \varepsilon * \frac{\partial L(x^t)}{\partial x^t}$
        t = t+1
    end while
  return $x^{(T)}$
end function
```

In the experiments, the results show that different split points can yield different attack effects, so the question is: how to split the neural network in the collaborative system, to make the inference data more secure? Generally, it is observed that the quality of recovered images decreases when the split layer goes deeper. This is straightforward as the relationship between input and output becomes more complicated and harder to revert when there are more layers. Besides, it is also observed that the image quality drops significantly, both qualitatively and quantitatively, on the fully-connected layer (fc1), indicating that model inversion with fully-connected layers is much harder than for convolutional layers. The reason is that a convolutional layer only operates on local elements (the locality depends on the kernel size), while a fully-connected layer entirely mixes up the patterns from the previous layer. Besides, the number of output neurons in a fully-connected layer is typically much smaller than input neurons. So it is relatively harder to find the reversed relationship from the output of the fully-connected layer to the input. And the first defense method is running fully-connected layers before sending out results.

Other possible defenses are making client-side networks deeper, trusted execution on untrusted participants differential privacy, and homomorphic encryption.

Improving robustness to MI attack In the paper [9], the author proposed the Mutual Information Regularization based Defense (MID) against MI attacks. The key idea is to limit the information about the model input contained in the prediction, thereby limiting the ability of an adversary to infer the private training attributes from the model prediction.

The author limits the dependency between X and $\hat{Y}$ to prevent the adversary from inferring the training data distribution associated with a specific label. The author's idea is to quantify the dependence between X and $\hat{Y}$ using their mutual information $I(X;\hat{Y})$ and incorporate it into the training objective as a regularizer. This defense, which is called MID, trains the target model via the loss function:

$$min_{f \in H} E_{(x,y)\sim p_{X,Y}}(x,y)$$
$$[L(y,f(x))] + \lambda I(X,\hat{Y}) \tag{8}$$

where

$$I(X;\hat{Y}) = \int_X \int_Y p_{X,Y}(x,y) log\left(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\right) \tag{9}$$

$L(y,f(x))$ is the loss function for the main prediction task, and λ is the weight coefficient that controls the tradeoff between privacy and utility on the main prediction task.

To deconstruct the proposed regularizer, mutual information is as follows:

$$I(X;\hat{Y}) = H(\hat{Y}) - H(\hat{Y}|X) \tag{10}$$

When f is a deterministic model, $H(\hat{Y}|X) = 0$ and introducing the mutual information regularizer effectively reduces the entropy of the model output, i.e., $H(\hat{Y})$. When f is stochastic, the regularizer will additionally promote the uncertainty of the model output for a fixed input, i.e., $H(\hat{Y}|X)$.

For Linear Regression, due to the deterministic nature of the model, the mutual information regularizer is reduced to $H(\hat{Y})$. Approximation of $\hat{Y}$ by a Gaussian mixture is:

$$p(\hat{y}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(\hat{Y}|Ax_i;\sigma^2) \tag{11}$$

where $x_{i}{}_{i=1}^{N}$ is the training set and σ is a free parameter. The author utilizes a Taylor-expansion based approximation for the entropy of Gaussian mixtures described in *Hu et al.* and derive the following approximation to $I(X,\hat{Y})$:

$$\widetilde{I_{lin}}(X,\hat{Y}) =$$

$$= -\frac{1}{N}\sum_{i=1}^{N} log\left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}exp\left(-\frac{1}{2}\left(\frac{Ax_i - Ax_j}{\sigma}\right)^2\right)\right) \tag{12}$$

For the decision tree, the author modifies ID3 [31] to incorporate the mutual information regularizer. Since decision trees trained with ID3 are deterministic, the mutual information regularizer again reduces to $H(\hat{Y})$. To defend against the MI attacks, the author adds $-H(\hat{Y})$ into the splitting criterion.

For networks, the author gets inspiration from the line of work on information bottleneck [32] [33] and regard the neural network as a Markov chain $Y - X - Z - \hat{Y}$, where X is the feature, Y is the ground truth label, Z is a stochastic encoding of the input X at some intermediate layer and defined by $P(Z|X;\sigma)$, and $\hat{Y}$ is the prediction. The author replaces $I(X,\hat{Y})$ with upper bound $I(X,Z)$ in the training object and train the neural network with loss function:

$$min_\theta - I(Z;Y) + \lambda I(Z,X) \tag{13}$$

The experiments show that this MID approach can significantly reduce the attack accuracy and effectively protect the ML model from MI attacks.

A simple additive noise method to defend MI attack In [10], the author demonstrates that the attack can be successfully performed with limited knowledge of the data distribution by the attacker, and show that NoPeekNN, an existing defensive methodology, protects completely different info from exposure, suggesting that a combined defense is important to completely shield personal user information.

NoPeekNN may be a technique for limiting knowledge reconstruction in SplitNNs by minimizing the gap correlation between the input data and the intermediate tensors throughout model training [34]. NoPeekNN optimizes the model by a weighted combination of the task's loss and a distance correlation loss, that measures the similarity between the input data and the intermediate data. NoPeekNN's loss weighting is governed by a hyperparameter $\alpha \in [0,\infty)$. While NoPeekNN was shown to cut back autoencoder's ability to reconstruct input information, it's not been applied to adversarial model inversion attack.

Similar to this work, to defend against model inversion attack on one-dimensional ECG data, [35] utilizes noise to the intermediate tensors in a SplitNN. The authors pack this defense as a differential privacy mechanism [36]. However, in that work, the addition of noise greatly impacts the model's accuracy for even modest epsilon values (98.9% to roughly 90% at $\epsilon = 10$). There is also a similar method introduced by [37] called Shredder. To minimize mutual information between input and intermediate data, this method will adaptively generate a noise mask.

In this work, the author considers an honest-but-curious computation server and an arbitrary number of data owners who run the correct computations during training and inference. At least one party attempts to steal input data from alternative parties by employing a model inversion attack. The attack method is as follows: 1) The attackers collect a dataset of inputs (raw data) and intermediate data made by the first model phase. 2) To convert the intermediate information into raw input data, they train an attack model. 3) They collect intermediate information made by some information owners and run it through the trained attack model to reconstruct the raw input information. This attack is considered a "black-box" since the internal parameters of the data owner model segment are not used in the attack. The author assumes that the model training method has been orchestrated by a third party in which there's just one computational server.

MI attack for large language models In [11], the author tries to extract text data from a language model trained on scrapes of the public Internet called GPT-2. Training data extraction attacks are usually seen as theoretical or academic and therefore unlikely to be exploitable in application. This can be even by the prevailing intuition that privacy leakage is correlated with overfitting, and since advanced LMs trained on massive (near terabyte-sized) datasets for a few epochs, they tend to not overfit. This paper proved that training data extraction attacks are viable.

First, is the definition of committal to memory. The author defines eidetic memorization as a special type of memorization. Unofficially, eidetic memorization is data information that has been memorized by a model despite solely showing during a tiny set of training instances. The fewer training samples that contain the information, the stronger the eidetic memorization is.

Theorem 2. A string s is extractable from an *LM* f_θ if there exists a prefix *c* such that:

$$s \leftarrow argmax_{s':|s'|=N} f_\theta(s'|c) \tag{14}$$

Fig. 4 presents the structure of extraction attack:

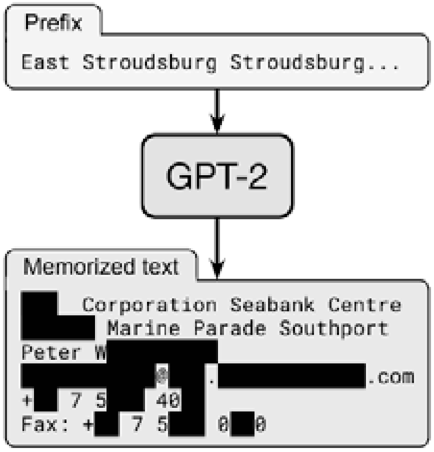


Fig. 4. Extraction attack. Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy [11]

Theorem 3. A string s is k-eidetic memorized (for k ≥ 1) by an $LM\ \ f_\theta$ if s is extractable from f_θ and s appears in at most k examples in the training data $X\colon |x \in X\colon s \subseteq x| \leq k$. This threat model is extremely realistic as several LMs are available through black-box APIs. For example, the GPT-3 model created by OpenAI is available through black-box API access. Auto-complete models trained on actual user information have conjointly been created public, though they reportedly use privacy-protection measures throughout training.

The extraction of training data from a language model has two-step: 1) Generate text. Generate a large quantity of data by unconditionally sampling from the model; 2) Predict which outputs contain memorized text. We next remove the generated samples that are unlikely to contain memorized text using a membership inference attack. These 2 steps correspond on to extracting model information (Theorem 2), then predicting that strings may be k-eidetic memorization (Theorem 3).

Fig. 5 presents the workflow of extraction attack and evaluation:

In total across all strategies, the author identify 604 unique memorized training examples from among the 1,800 possible candidates, for an aggregate true positive rate of 33.5% (the best variant has a true positive rate of 67%).

For countermeasures, the author suggests that needs to be techniques developed to specifically address their attacks. Training with differentially private techniques is one method for mitigating privacy leakage, however, the author believes that it will be necessary to develop new methods that can train models at this extreme scale (e.g., billions of parameters) without sacrificing model accuracy or training time.

Defending MI attack via prediction purification In [12], the author proposes a unified approach, namely a purification framework, to defend data inference attacks. It purifies the confidence score vectors foretold by the target classifier by reducing their dispersion. The setup may be additional specialized in defensive a selected attack via adversarial learning.

The model owner trains a machine learning classifier F on its training dataset D_{train} and test F on validation dataset D_{val}. Both D_{train} and D_{val} are drawn from the same underlying data distribution $p_r(X)$. The attacker aims at performing data inference attacks against the target classifier F. Consider that the classifier F works as a black-box “oracle” to the attacker, i.e., the attacker can only query F with its data sample x and obtain the prediction scores $F(x)$. The attacker is also assumed to have auxiliary information $\mathcal{A}$. Given a prediction vector $F(x)$ on some victim data point x, the attacker wants to find an attack function $A(F(x), O(F), \mathcal{A})$ for membership inference:

$$A(F(x), O(F), \mathcal{A}) = m \in \{0{,}1\} \tag{15}$$

for model inversion:

$$A(F(x), O(F), \mathcal{A}) = \hat{x} \tag{16}$$

where $O(F)$ represents the attacker’s blackbox access to the oracle classifier F.

For purification, the base of purification is purifier G. The author designed G as an autoencoder and is used to reduce the dispersion of the confidence scores as well as to preserve the utility of the classifier. G is trained on the confidence scores predicted by F on the defender’s reference dataset D_{ref}. The author trained G to also produced the label predicted by F by adding a cross-entropy loss function. G is trained to minimize the function:

$$\mathcal{L}(G) =$$

$$= E_{x\sim p_r(x)}\left[\mathcal{R}\left(G(F(x)), F(x)\right) + \lambda\mathcal{L}\left(G\left(F(x), argmaxF(x)\right)\right)\right] \tag{17}$$

The author also provides specialized G for MI attack.

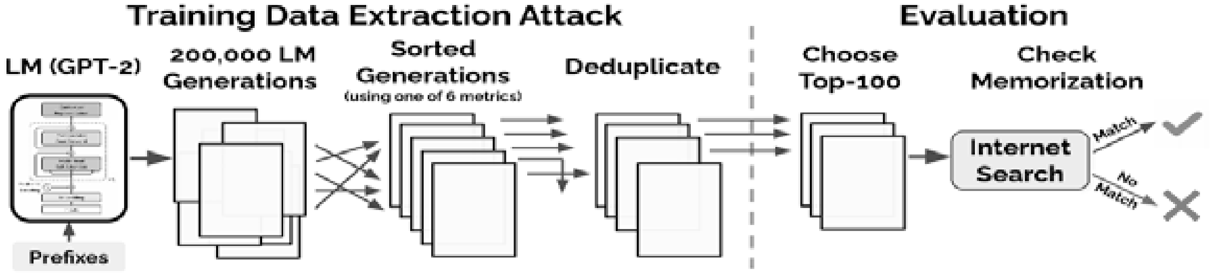


Fig.5. 1) Attack. We begin by generating many samples from GPT-2 when the model is conditioned on (potentially empty) prefixes. We then sort each generation according to one of six metrics and remove the duplicates. This gives us a set of potentially memorized training examples. 2) Evaluation. We manually inspect 100 of the top-1000 generations for each metric. We mark each generation as either memorized or not-memorized by manually searching online, and we confirm these findings by working with OpenAI to query the original training data [11]

For each classification task, we can see that the single purifier is able to concurrently decrease the inference accuracy and increase the inversion error as well as preserve the classification accuracy. The purifier has almost no defense effect against the Label attack where the confidence score information is not leveraged.

MLPrivacyGuard In [18], the author presents MLPrivacyGuard, a countermeasure against black-box MI attack. In this countermeasure, they add controlled noise to the output of the confidence function. The author demonstrates that when noise is injected with a long-tailed distribution, the objectives of low misclassification error with a strong defense can be attained as model inversion attacks are neutralized because numerical approximation of gradient ascent is unable to converge.

MLPrivacyGuard is a measure to forestall model inversion attacks that don't need retraining or any type of modification to the ML system's inner workings. MLPrivacyGuard relies on the injection of long-tailed distributed errors to the output $\tilde{f}$ of the model, so that a model inversion attack can not converge, whilst maintaining the functionality of the ML system reliable for legitimate users. The idea behind MLPrivacyGuard is that when the confidential information has some level of randomness the model inversion attack is unable to converge in black-box systems. The reason for this is straightforward: since a black-box model inversion has to rely on numerical approximations of the gradient $\tilde{f}(x)$, which are obtained by numerical differentiation of $\tilde{f}$ on each of the features, the attack is unable to converge if the value of $\tilde{f}$ is nondeterministic.

To preserve correct classifications for legitimate users, the author guaranteed this with the distributions picked for the random errors. In the experiments, the error size has exponential distribution, i.e. the absolute value of the error injected is x with probability $\lambda e^{-\lambda x}$.

The result of experiments shows that this MLPrivacyGuard approach increases the classification error rate at most by 2% while defeating adversarial model inversion attacks.

MI attack without knowledge of non-sensitive attributes In [15], the author proposes a General Model Inversion (GMI) framework to capture the scenario where knowledge of the non-sensitive attributes is not necessarily provided. This framework also captures the scenario of *Fredrikson et al.*, notably, it enables a new type of model inversion attack that infers sensitive attributes without the knowledge of non-sensitive attributes by modifying the ML model into a target ML model via data poisoning. The GMI attack is defined by a tuple of three algorithms: Setup, Poisoning and ModelInversion [15]. Fig. 6 presents the workflow of GMI attack:

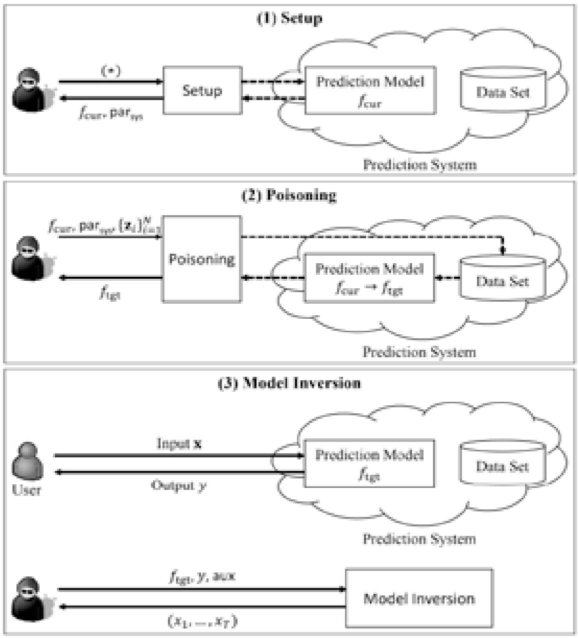


Fig. 6. Workflow of GMI attack [15]

Leverage the functional mechanism to prevent MI attack In [16], the author develops a novel approach that leverages the functional mechanism to perturb coefficients of the polynomial representation of the objective function but effectively balances the privacy budget for sensitive and non-sensitive attributes in learning the differential privacy preserving regression model.

This approach leverages the functional mechanism proposed in [38] but perturbs the polynomial coefficients of the objective function with different magnitudes of noise. This approach can effectively weaken the correlation between the sensitive attributes with the output to prevent model inversion attacks whereas retaining the utility of the released model by decreasing the perturbation effect on non-sensitive attributes.

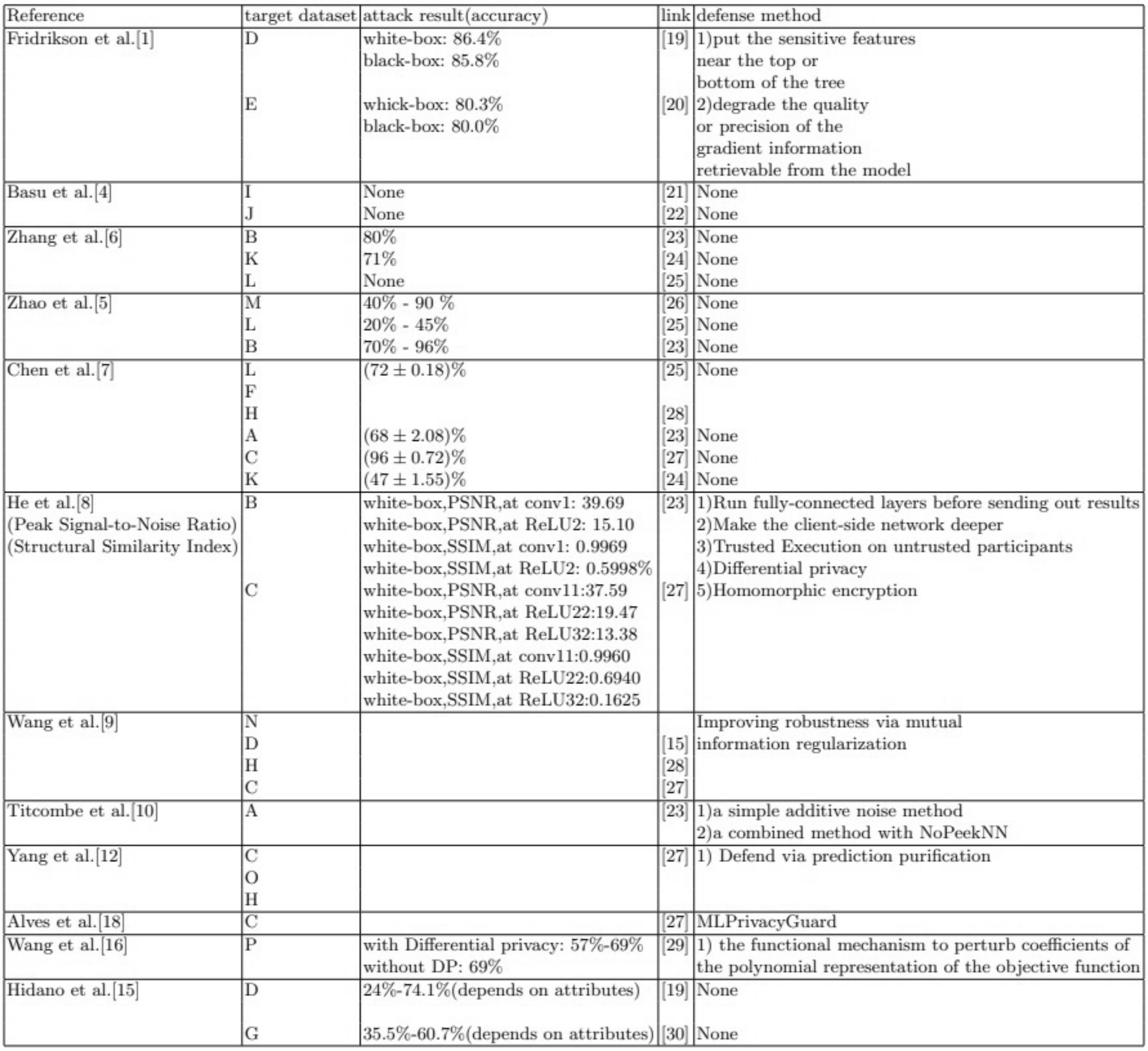
2. Table of MI attacks with their attributes

In this section, we summarize the results of our review (Table 1). Presented datasets are: A - MNIST, B - MNIST handwritten digit, C - CIFAR-10, D - FiveThirtyEight survey (How americans like their steak), E - GSS marital happiness survey, F - Flickr-Faces-HQ(FFHQ), G - MovieLens 1M Dataset, H - FaceScrub, I - Numeric MNIST, J- Fashion MNIST, K - ChestX-ray8, L - CelebA, M - iCV-MEFED, N - IPWC, O - Purchase100, P - Adult dataset.

As can be seen from this table, attacks clearly prevail over defenses. In reality, only the practical feasibility of attacks really protects existing systems. All attacks require multiple polling of models. If this is not

Table 1

Summary of MI attacks



Reference	target dataset	attack result(accuracy)	link	defense method
Fridrikson et al.[1]	D	white-box: 86.4% black-box: 85.8%	[19]	1)put the sensitive features near the top or bottom of the tree
	E	whick-box: 80.3% black-box: 80.0%	[20]	2)degrade the quality or precision of the gradient information retrievable from the model
Basu et al.[4]	I	None	[21]	None
	J	None	[22]	None
Zhang et al.[6]	B	80%	[23]	None
	K	71%	[24]	None
	L	None	[25]	None
Zhao et al.[5]	M	40% - 90 %	[26]	None
	L	20% - 45%	[25]	None
	B	70% - 96%	[23]	None
Chen et al.[7]	L	$(72 \pm 0.18)\%$	[25]	None
	F			
	H		[28]	
	A	$(68 \pm 2.08)\%$	[23]	None
	C	$(96 \pm 0.72)\%$	[27]	None
	K	$(47 \pm 1.55)\%$	[24]	None
He et al.[8] (Peak Signal-to-Noise Ratio) (Structural Similarity Index)	B	white-box,PSNR,at conv1: 39.69 white-box,PSNR,at ReLU2: 15.10 white-box,SSIM,at conv1: 0.9969 white-box,SSIM,at ReLU2: 0.5998%	[23]	1)Run fully-connected layers before sending out results 2)Make the client-side network deeper 3)Trusted Execution on untrusted participants 4)Differential privacy
	C	white-box,PSNR,at conv11:37.59 white-box,PSNR,at ReLU22:19.47 white-box,PSNR,at ReLU32:13.38 white-box,SSIM,at conv11:0.9960 white-box,SSIM,at ReLU22:0.6940 white-box,SSIM,at ReLU32:0.1625	[27]	5)Homomorphic encryption
Wang et al.[9]	N			Improving robustness via mutual information regularization
	D		[15]	
	H		[28]	
	C		[27]	
Titcombe et al.[10]	A		[23]	1)a simple additive noise method 2)a combined method with NoPeekNN
Yang et al.[12]	C		[27]	1) Defend via prediction purification
	O			
	H			
Alves et al.[18]	C		[27]	MLPrivacyGuard
Wang et al.[16]	P	with Differential privacy: 57%-69% without DP: 69%	[29]	1) the functional mechanism to perturb coefficients of the polynomial representation of the objective function
Hidano et al.[15]	D	24%-74.1%(depends on attributes)	[19]	None
	G	35.5%-60.7%(depends on attributes)	[30]	None

an MLaaS system, then it will be impossible to carry out an attack directly.

3. Summary and future works

From all this information above we find that several types of MI attacks have been created and successful test on various datasets like CIFAR-10, MNIST, FiveThirtyEight, and so on. But the problem is, those existing countermeasures are passive counter, which means these countermeasures are just been applied in the ML model, and each countermeasure can only defend a specific attack method. Considering there are many attack methods and these methods can also be iterated, if one ML model wants to survive under those attacks, it has to apply many countermeasures simultaneously. We think that this approach may lower not only the efficiency of the ML model but also the accuracy.

So, if we can build a MI attack detector, and this detector can immediately cut off the connection between user and model when it detects a MI attack(or some action similar to an MI attack), it will be great, and it can save much cost for MLaaS provider. In our opinion, this is a promising direction.

For the detector, we want to start from GAN. In GAN there is a generator G and discriminator D, we can use G to simulate existing attack methods and let D discriminate whether one is a MI attack or not. Also, if possible, we can import CNN to our detector. For images, CNN can learn its features; and for MI attacks, maybe we can transform MI attack into a type that can let CNN learn its features.

References

1. *Fredrikson, M., Jha, S., & Ristenpart, T.* (2015, October). 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).
2. *Wu, X., Fredrikson, M., Jha, S., & Naughton, J.F.* (2016, June). A methodology for formalizing model-inversion attacks. In 2016 IEEE 29th Computer Security Foundations Symposium (CSF) (pp. 355-370). IEEE.
3. *Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S.* (2018, July). Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF) (pp. 268-282). IEEE.
4. *Basu, S., Izmailov, R., & Mesterharm, C.* (2019). Membership model inversion attacks for deep networks. arXiv preprint arXiv:1910.04257.
5. *Zhao, X., Zhang, W., Xiao, X., & Lim, B.Y.* (2021). Exploiting Explanations for Model Inversion Attacks. arXiv preprint arXiv:2104.12669.
6. *Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., & Song, D.* (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 253-261).
7. *Chen, S., Kahla, M., Jia, R., & Qi, G.J.* (2021). Knowledge-Enriched Distributional Model Inversion Attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 16178-16187).
8. *He, Z., Zhang, T., & Lee, R.B.* (2019, December). Model inversion attacks against collaborative inference. In Proceedings of the 35th Annual Computer Security Applications Conference (pp. 148-162).
9. *Wang, T., Zhang, Y., & Jia, R.* (2020). Improving robustness to model inversion attacks via mutual information regularization. arXiv preprint arXiv:2009.05241.
10. *Titcombe, T., Hall, A. J., Papadopoulos, P., & Romanini, D.* (2021). Practical Defences Against Model Inversion Attacks for Split Neural Networks. arXiv preprint arXiv:2104.05743.
11. *Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K. & Raffel, C.* (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 2633-2650).
12. *Yang, Z., Shao, B., Xuan, B., Chang, E. C., & Zhang, F.* (2020). Defending model inversion and membership inference attacks via prediction purification. arXiv preprint arXiv:2005.03915.
13. *Ateniese, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., & Felici, G.* (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. International Journal of Security and Networks, 10(3), 137-150.
14. *Xu, R., Baracaldo, N., & Joshi, J.* (2021). Privacy-Preserving Machine Learning: Methods, Challenges and Directions. arXiv preprint arXiv:2108.04417.
15. *Hidano, S., Murakami, T., Katsumata, S., Kiyomoto, S., & Hanaoka, G.* (2017, August). Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In 2017 15th Annual Conference on Privacy, Security and Trust (PST) (pp. 115-11509). IEEE.
16. *Wang, Y., Si, C., & Wu, X.* (2015, June). Regression model fitting under differential privacy and model inversion attack. In Twenty-Fourth International Joint Conference on Artificial Intelligence.
17. *Wang, K. C., Fu, Y., Li, K., Khisti, A. J., Zemel, R., & Makhzani, A.* (2021, May). Variational Model Inversion Attacks. In Thirty-Fifth Conference on Neural Information Processing Systems.
18. *Alves, T. A., França, F. M., & Kundu, S.* (2019, May). MLPrivacyGuard: Defeating Confidence Information based Model Inversion Attacks on Machine Learning Systems. In Proceedings of the 2019 on Great Lakes Symposium on VLSI (pp. 411-415).
19. *W. Hickey.* FiveThirtyEight.com DataLab: How americans like their steak. http://fivethirtyeight.com/datalab/how-americans-like-their-steak/, May 2014.
20. *J. Prince.* Social science research on pornography. http://byuresearch.org/ssrp/downloads/GSShappiness.pdf.
21. *Deng, L.* (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.
22. *H. Xiao, K. Rasul, and R. Vollgraf.* Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR, abs/1708.07747, 2017.
23. *Yann Lecun, Leon Bottou, Y Bengio, and Patrick Haffner.* Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86:2278 – 2324, 12 1998.
24. *Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers.* Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax

diseases. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2097–2106,2017.

25. *Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang.* Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, pages 3730–3738, 2015.
26. *Christer Loob, Pejman Rasti, Iiris Lusi, Julio CS Jacques, Xavier Baro, Sergio Escalera, Tomasz Sapinski, Dorota Kaminska, and Gholamreza Anbarjafari.* Dominant and complementary multi-emotional facial expression recognition using c-support vector classification. In 2017 12th IEEE International Conference on Automatic Face \& Gesture Recognition (FG 2017), pages 833–838. IEEE, 2017.
27. *Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.* Cifar-10(canadian institute for advanced research).
28. *H.-W. Ng, S. Winkler.* A data-driven approach to cleaning large face datasets. Proc. IEEE International Conference on Image Processing (ICIP), Paris, France, Oct. 27-30, 2014.
29. *Bache, K. & Lichman, M.* (2013). UCI Machine Learning Repository (Technical report, University of California, Irvine, School of Information and Computer Sciences)
30. GroupLens Research, “MovieLens 1M Dataset,” http://grouplens.org/datasets/movielens/, 2003.
31. *Quinlan, J.R.* (1986) Induction of Decision Trees. Machine Learning, 1, 81-106. http://dx.doi.org/10.1007/BF00116251
32. *Shwartz-Ziv, R., & Tishby, N.* (2017). Opening the Black Box of Deep Neural Networks via Information. ArXiv, abs/1703.00810.
33. *Alemi, Alexander A., et al.* “Deep variational information bottleneck.” arXiv preprint arXiv:1612.00410 (2016).
34. *Vepakomma, P., Gupta, O., Dubey, A., & Raskar, R.* (2019). Reducing leakage in distributed deep learning for sensitive health data.
35. *Sharif Abuadbba, Kyuyeon Kim, Minki Kim, Chandra Thapa, Seyit A Camtepe, Yansong Gao, Hyoungshick Kim, and Surya Nepal.* Can we use split learning on 1d cnn models for privacy preserving training? arXiv preprint arXiv:2003.12365, 2020.
36. *Cynthia Dwork.* Differential privacy: A survey of results. In International conference on theory and applications of models of computation, pp. 1–19. Springer, 2008.
37. Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Ali Jalali, Dean Tullsen, and Hadi Esmaeilzadeh. Shredder: Learning noise distributions to protect inference privacy. In Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 3–18, 2020.
38. *Zhang, Jun & Zhang, Zhenjie & Xiao, Xiaokui & Yang, Yin & Winslett, Marianne.* (2012). Functional Mechanism: Regression Analysis under Differential Privacy. Proc. VLDB Endowment. 5.
39. *Namiot, D., Ilyushin, E., & Pilipenko, O.* (2022). On Trusted AI Platforms. International Journal of Open Information Technologies, 10(7), 119-127 (in Russian).
40. *Namiot, D., Ilyushin, E., & Chizhov, I.* (2022). On a formal verification of machine learning systems. International Journal of Open Information Technologies, 10(5), 30-34.

Junzhe Song. Student of the magistracy of the faculty of CMC of Lomonosov Moscow State University, MSU, Faculty of Computational Mathematics and Cybernetics, Russia, 119991, Moscow, GSP-1, 1-52, Leninskiye Gory, e-mail: songjz@smbu.edu.cn

D.E. Namiot. Dr. of Sci., senior researcher of the faculty of CMC of Lomonosov Moscow State University, MSU, Faculty of Computational Mathematics and Cybernetics, Russia, 119991, Moscow, GSP-1, 1-52, Leninskiye Gory, e-mail: dnamiot@gmail.com (correspondent author)