Your Model Trains on My Data? Protecting Intellectual Property of Training Data via Membership Fingerprint Authentication

Gaoyang Liu, *Member, IEEE*, Tianlong Xu, Xiaoqiang Ma, *Member, IEEE*, Chen Wang, *Senior Member, IEEE*

***Abstract*—In recent years, data has become the new oil that fuels various machine learning (ML) applications. Just as the oil refining, providing data to an ML model is a product of massive costs and expertise efforts. However, how to protect the intellectual property (IP) of the training data in ML remains largely open. In this paper, we present MeFA, a novel framework for detecting training data IP embezzlement via Membership Fingerprint Authentication, which is able to determine whether a suspect ML model is trained on the to be protected target data or not. The key observation is that a part of data has a similar influence on the prediction behavior of different ML models. On this basis, MeFA leverages membership inference techniques to extract these data as the fingerprints of the target data and constructs an authentication model to verify the data's ownership by identifying the obtained membership fingerprints. MeFA has several salient features. It does not assume any knowledge of the suspect model except for its black-box prediction API, through which we can merely get the prediction output of a given input, and also does not require any modification to the dataset or the training process, since it takes advantage of the inherent membership property of the data. As a by-product, MeFA can also serve as a post-protection to verify the ownership of ML models, without modifying the training process of the model. Extensive experiments on three realistic datasets and seven types of ML models validate the effectiveness of MeFA, and demonstrate that it is also robust to scenarios when the training data is partially used or preprocessed with representative membership inference defenses.**

***Index Terms*—Training data authentication, intellectual property protection, membership inference attack, membership fingerprint, machine learning model.**

I. Introduction

In recent years, the explosive growth of data has promoted the application of machine learning (ML) in various fields, ranging from natural language processing [1] to computer vision [2]. As is known to all, it is a non-trivial task to obtain the data that can be used to train ML models from the raw data, especially at an industrial level. Specifically, providing the training data is a product of massive costs and expertise efforts, including data collection, data annotation, and data preprocessing. However, it is reported that attackers can illegally obtain the data through multiple advanced techniques [3]–[6]. With the reduction of technical barriers, they can use the unauthorized data to train an ML model to make illegal profits. Therefore, in order to protect the data owner's legitimate benefits, it is necessary to protect the intellectual property (IP) of the data, i.e., to externally verify the ownership of the data.

Previous IP protections in ML field mainly focus on protecting the IP of the trained deep neural networks (DNN) model and verifying the model creator's identity [7]–[11]. These works consider that the model's IP corresponds to the ownership of training data, cost of computing resources, and experience of ML expertise. However, if the ML model is trained on a dataset that the trainer has no permission to use, the IP and ownership of this dataset will be violated. In this case, existing IP protections can only verify the ownership of an ML model, but cannot establish a clear association between an ML model and the training data that needs to be protected. How to verify the ownership of the target data with respect to a suspect ML model remains largely open.

In this paper, we bridge this gap by presenting MeFA, a novel framework for training data IP verification via Membership Fingerprint Authentication, which is able to determine whether a suspect ML model is trained on the (to be protected) target data. We consider the real-world scenario where the suspect ML model is typically deployed as a black-box to keep the model internals secret, in the way as the widely deployed machine learning as a service (MLaaS) platforms such as Google AI[1], Amazon ML[2] and Microsoft AzureML[3]. Therefore, we have no knowledge about the model (neither the structure or parameters of the model, nor the training algorithm), but only the access to the model's input-output prediction API.

Central to the MeFA's idea is the fact that during the ML model's training process, the data record in the training set influences the prediction behavior of the ML model, in order to fit the model to the whole training data. As a consequence, different training sets naturally bring in the dissonance of the ML models' prediction behaviors, whereas the ML models trained on the same training set should have similar predic-

This work was supported in part by the National Natural Science Foundation of China under Grants 61872416, 62171189, 62002104 and 62071192; by the Fundamental Research Funds for the Central Universities of China under Grant 2019kfyXJJS017; by the Key Research and Development Program of Hubei Province under Grant 2020BAB120; and by the special fund for Wuhan Yellow Crane Talents (Excellent Young Scholar). *(Corresponding author: Chen Wang.)*

G. Liu, T. Xu, X. Ma and C. Wang are with the Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. G. Liu is also with the School of Computing Science, Simon Fraser University, British Columbia, Canada. Email: {liugaoyang, tlxu, maxiaoqiang, chenwang}@hust.edu.cn.

[1]https://cloud.google.com/ai-platform

[2]https://aws.amazon.com/machine-learning

[3]https://azure.microsoft.com/en-us/services/machine-learning/

tion behaviors. By *measuring the similarity of the prediction behaviors between the suspect model and the model trained on the target data*, MeFA can stand a good chance to discriminate whether the target data is involved in the training of the suspect model, which in turn leads to the ownership verification of the target data.

Although the basic idea sounds simple, MeFA confronts with two major technical challenges. The first one is how to measure the similarity of prediction behavior, since it is non-trivial to compare the prediction outputs of all records in the target data, due to limited number of queries and large computing overheads. Motivated by the recent advances in membership inference attacks (MIA) [12], [13], which can determine whether a data record was used for training a given ML model, we find that part of the target data has remarkable and consistent influence (in terms of membership sensitivity) on the predictions of different types and structures of ML models trained on them. Therefore, we propose to select multiple records, which have the above-mentioned influence among different models, as the membership fingerprints of the target data. By comparing the prediction results of the selected membership fingerprints obtained from the suspect model and the model trained on the target data, we can thus measure the similarity of prediction behaviors of these two models.

The second challenge is how to identify the consistent influence from the predictions of different ML models and find the universal membership fingerprints from the target data, as in practice we have only the black-box access to the suspect model with few information about its training settings. Considering the fingerprints are sensitive to the types and structures of ML models [14], directly using the fingerprints generated by existing MIA methods can only identify the influence of the record in the target data on one particular model, but would fail to cross different models. To address this challenge, we propose to first train two sets of reference models with commonly used training algorithms and settings, on the target dataset and an external dataset[4], respectively. Then we construct one universal authentication model with the prediction of all reference models, and select from the target data the records that are most sensitive to all the reference models as the membership fingerprints based on the detection results of the authentication model. The authentication model along with the membership fingerprints can further be used to facilitate the data IP protection.

MeFA has several salient features. First, MeFA does not make any modifications to the target data or the training process but takes advantage of the inherent membership property of the data, which maintains the integrity and utility of the data. Second, MeFA is a model-agnostic framework that can be performed on any type of ML model, without preknowledge of the suspect model's type, structure, parameters, and training process, except for its black-box prediction API. Besides, MeFA is also able to verify the ownership of ML models (not limited to DNN models). We empirically find that membership fingerprints have a great influence on the model's prediction (i.e. the contribution of membership fingerprints to an ML model's prediction involved in the model's training process). By detecting the presence of such an influence, MeFA can facilitate the IP verification of ML models.

Our major contributions are summarized as follows.

- We present a novel framework for detecting training data IP embezzlement, and show for the first time how membership inference techniques can be adopted to construct the fingerprints of the target data for its IP verification.
- We construct an authentication model that can verify the data's ownership by inferring the membership property of the obtained membership fingerprints with respect to a black-box ML model.
- We extensively evaluate MeFA on three realistic datasets and seven types of ML models. The results validate the effectiveness of MeFA, and demonstrate that it is also robust to scenarios when the training data is partially used or preprocessed with representative membership inference defenses.

The remainder of this paper is organized as follows. Section II overviews some related works. Section III introduces the threat model. Section IV describes the design of MeFA, followed by the extensive performance evaluation in Section V. Finally, Section VI concludes this paper. The code of MeFA has been released for reproducibility purposes[5].

II. Related Work

A. DNN IP Protection

DNN models suffer from illegal copy, redistribution, or embezzlement without the model owner's permission, and their IPs need to be protected. Therefore, many DNN IP protections have been proposed [10], [15], which generally fall into the following three classes.

Watermarking-based protection. In these methods, a set of watermarks are embedded into the model at its training or fine-tuning phase [16], [17], and the protections can determine the model's IP by checking the existence of the model watermarks. For example, Uchida et al. [18] propose the first DNN IP protection method, which trains the model with an additional regularization loss to embed the watermark to the model weights. This protection is only applicable to the white-box model that can fully access the model's parameters. Then Rouhani et al. [19] propose a black-box protection technique that embeds the watermark into the probability density function of the activation set of each layer. The embedded watermark can be triggered by a corresponding set of input to remotely verify the IP of DNNs. Besides, Kuribayashi et al. [20] propose a quantifiable watermark embedding approach, which could reduce the magnitude of the weight changes. Different from the above schemes carrying a watermark by model weights, Wu et al. [21] introduce a watermarking framework that embeds the watermark into the model's outputs.

Backdoor-based protection. These methods leverage the backdoor attacks [22], [23] to make the DNNs output a specified label when a specific backdoored input arrives. Then

[4]The external dataset has the same format and value ranges of each feature with the target dataset, detailed in Section IV-B.

[5]https://www.dropbox.com/s/5j59vbh6qil4hwe/MeFA-Code.zip?dl=0

the predictions on the inputs are compared with the assigned labels to authenticate the model's IP. For instance, Yossi et al. [24] and Zhang et al. [7] leverage the backdoor attack to change the decision boundary of the protected model, which can deliberately output specific (incorrect) labels for a particular set of inputs. Then these specific input-output pairs can be used to verify the protected model's IP. Li et al. [8] use an encoder to combine the ordinary samples and the exclusive logo together to generate multiple modified samples. By infusing these samples into the protected models with specified labels, they can leave a set of backdoors as the basis for the model IP claim. It has been validated that backdoor-based protections would be simply breached through model compression/distillation and model pruning [25], [26]. To mitigate this issue, Li et al. [8] propose a backdoor embedding method that can only be inserted during the initial training of the model, and Jia et al. [27] propose Entangled Watermarks, in which the embedded backdoors are entangled with the legal data of the model.

Fingerprint-based protection. Recent works have demonstrated that the "fingerprint" of the DNNs can also be used for IP protection. Chen et al. [28] propose DeepMarks that allows the owner of DNNs to embed a unique fingerprint within the weights of the model itself. Merrer et al. [29] take advantage of adversarial examples which are very close to the decision boundary of the model as the fingerprint to verify the model's IP. Cao et al. [11] find some data records near the decision boundary of the model, and utilize these records of the protected model to fingerprint the model and track its IP. Maini et al. [30] make use of the distance of multiple training records to the decision boundary as the watermarks to verify the model's IP. By aggregating the signals of multiple records, the verification of model IP can succeed with a high accuracy. A recent work [14] uses MIAs to generate the fingerprint of a model for its IP protection. It trains a membership discriminator along with the protected model simultaneously.

It is observed that the watermark-based and backdoor-based protections require modifying the training and fine-tuning process of the model, while the fingerprint-based protection requires the training data of the protected model, or internal access to the model's parameters. However, the owner of the suspect model often only provides the black-box interface to the users. In the scenario of training data's IP protection, we cannot manipulate the training and fine-tuning process of the suspect model, which will invalidate the watermark-based and backdoor-based protections. In addition, the existing protection methods usually are targeted at protecting the IP of one certain DNN model. In our scenario, the adversary could use multiple algorithms and settings to train his model on the target data, and coupled with the randomness of the training process, the fingerprint-based protection can hardly produce a fingerprint that exactly matches the adversary's model. Against this background, we present MeFA which does not modify the target data or the training process of the protected model and can generate universal fingerprints for multiple ML models.

B. MIA against ML Models

MIA against ML models was first studied by Shokri et al. [12]. The purpose of MIA is to determine whether a given record was used to train an ML model [31]. It can be formulated as:

$$\mathcal{A}(f, \mathbf{x}) \rightarrow \textbf{In}/\textbf{Out}$$

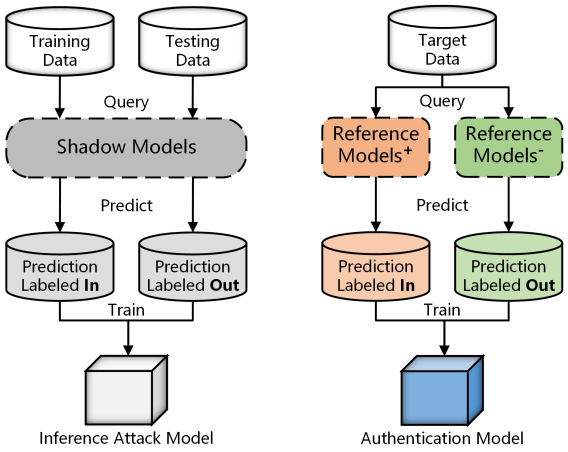
where f is the ML model, $\mathbf{x}$ is the given record, and $\mathcal{A}$ represents the methodology of MIA. The label **In** means that $\mathbf{x}$ belongs to f's training set D_{train} while **Out** means not.

The basic idea of MIA is straightforward: since each record in the training set influences many of the model parameters to minimize its contribution to the training loss, the trained ML model often behaves differently on the data that they were trained on versus the one that they "see" for the first time [12], [13], [32], [33], and such the prediction difference can be reflected in the prediction probability of f. By constructing a binary classifier, the MIA attacker can separate the member records from the non-members.

In the first MIA work [12], Shokri et al. construct multiple shadow models with the same structure as the victim model and derive the shadow model's outputs and the ground truth of membership, to construct multiple attack models. Then Salem et al. [32] propose ML-Leaks and show that it is possible to achieve the resemble attack performance with only one shadow model, rather than multiple shadow models. Instead of training the shadow model, Liu et al. [34] leverage the generative adversarial network (GAN) to train a mimic model. Except for imitating the prediction behavior, some works use other information of the victim model to perform MIAs, including the training loss [35], model parameters [36], model gradients [37], and output distributions [13]. Recently, some researchers focus on performing MIAs with the minimum information of the victim model, and several works have been proposed that only require the predicted label of the victim models [33], [38]. Bo et al. [13] propose BLINDMI which releases the reliance on the shadow models by probing the target model and then inferring the membership directly from the probing results via differential comparisons. Saeidian et al. [39] leverage information theory to prove the risk of membership privacy in ML models theoretically.

Essentially, the problem of data IP protection aims to verify the connection between the target data and the suspect model. Existing MIA studies point out that, if a record takes part in the training process of an ML model, the training algorithms aim to minimize this record's contribution to the training loss. Thus this training record would leave its unique influence on the model's parameters that can be reflected through the model's prediction, especially the prediction of itself. As such, MIA can infer this record's membership property with the model's prediction on this record, regardless of the type and the structure of the model. Motivated by this property, we propose to leverage the MIA results of the target data against different types of models to examine the similarity of prediction behavior for the training data IP protection.

What is worth being emphasized is that, existing MIAs can only breach the membership privacy against one certain model, while we need to screen out the records that have



(a) Existing MIAs (b) MIA in MeFA

Fig. 1. The key difference between existing MIAs and MIA in MeFA. Existing MIAs attempt to find the prediction differences between the training and testing data on the same model, while the MIA in MeFA tries to extract the prediction similarities among different models trained on the target data.

remarkable and consistent influence across multiple types and structures of ML models from the target data. Therefore, we design an MIA that can identify such records based on the prediction results of ML models trained on the target data, without requiring interaction with the suspect model. With the help of our proposed MIA, MeFA can select these records from the target data, which are further used to verify the IP of the suspect model's training data. The key difference between existing MIAs and the MIA in MeFA is shown in Fig. 1.

III. Threat Model

We consider a dataset owner who owns the IP of a dataset (i.e., the target data), and an attacker who can illegally obtain this dataset through multiple ways and construct an ML model on it (c.f. Fig. 2). Our threat model is described as follows.

Attacker's Goal. The attacker's goal is to misappropriate a dataset that he does not have the ownership, and to train an ML model on this dataset to obtain economic benefits or improve the quality of service of his own models.

Attacker's Capability. An attacker may derive and steal the target data through multiple ways. A prominent way is to hack into a data server or trade through the dark web. Perhaps less directly, the attacker could also reconstruct the target data through abusing the legitimate ML models trained on it [40], [41]. Finally, the attacker may directly access the target data. This may happen when the data owners expect to open-source their data for academic purposes but disallow the commercial usage. Then the attacker can train an ML model on the unauthorized dataset on their local devices. Even for the attackers without expertise in ML, they can use the MLaaS to construct the ML model and release its prediction API for economic benefits.

Suspect Model. We consider a practical scenario in which the suspect model $\mathcal{S}$ is deployed as a black-box, and the internal details of the model, including the model structure/parameters, as well as the training algorithm and training

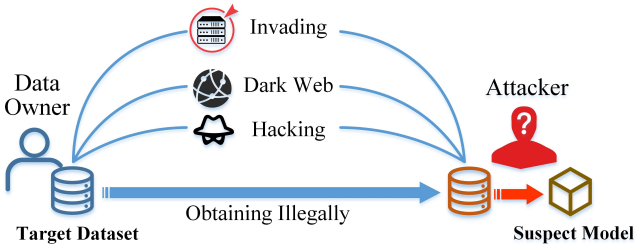


Fig. 2. Illustrative threat of dataset IP in ML.

settings, are kept secret to users. Given an input record $\mathbf{x}$, the suspect model only outputs the prediction probability vector $[\mathcal{S}_1(\mathbf{x}), \mathcal{S}_2(\mathbf{x}), ..., \mathcal{S}_C(\mathbf{x})]$, where $\mathcal{S}_c(\mathbf{x})$ represents the predicted probability of class c, and C is the number of classes that $\mathcal{S}$ can take. Note that in some cases, the suspect model may restrict the output to the top-k probabilities to evade the legality detection.

IV. Design of MeFA

A. Overview

The tasks of MeFA can be briefly formalized as follows: given the target data D_{tgt} whose IP needs protection and a black-box query access to the suspect model $\mathcal{S}$, MeFA first identifies the membership fingerprints of D_{tgt}, then infers the membership property of the fingerprints concerning $\mathcal{S}$, and finally determines whether D_{tgt} is used to train $\mathcal{S}$ according to the inference results. To this end, MeFA mainly involves the following two steps (c.f. Fig. 3).

(1) ***Membership Fingerprint Selection***. In order to extract the membership fingerprints of D_{tgt}, MeFA first trains two sets of reference models on the target data and the external dataset, respectively. Then MeFA utilizes the reference models' prediction on D_{tgt} to train an authentication model $\mathcal{A}$. By leveraging $\mathcal{A}$, MeFA can select from D_{tgt} the records on which all reference models have similar prediction behaviors as the membership fingerprints.

(2) ***Membership Fingerprint Authentication***. In this step, MeFA first queries $\mathcal{S}$ with the membership fingerprints and obtains the corresponding prediction results, which are further inputted to $\mathcal{A}$ to infer the membership property of the fingerprints with respect to $\mathcal{S}$. Finally, according to the inference results, MeFA can determine whether there exists a connection between D_{tgt} and $\mathcal{S}$.

B. Membership Fingerprint Selection

1) Reference Models Construction: Since ML models trained on the same dataset have similar prediction behaviors, the key idea of MeFA is comparing the similarity of prediction behavior of different models to verify the IP of a suspect model's training data. Although the idea is simple, it is difficult to extract and compare the similarity of ML models' prediction behaviors. It seems that directly comparing the prediction probabilities of $\mathcal{S}$ with that of the model trained on D_{tgt} is one feasible solution (we consider this method as a comparing baseline in the evaluation section). However, there are distinct

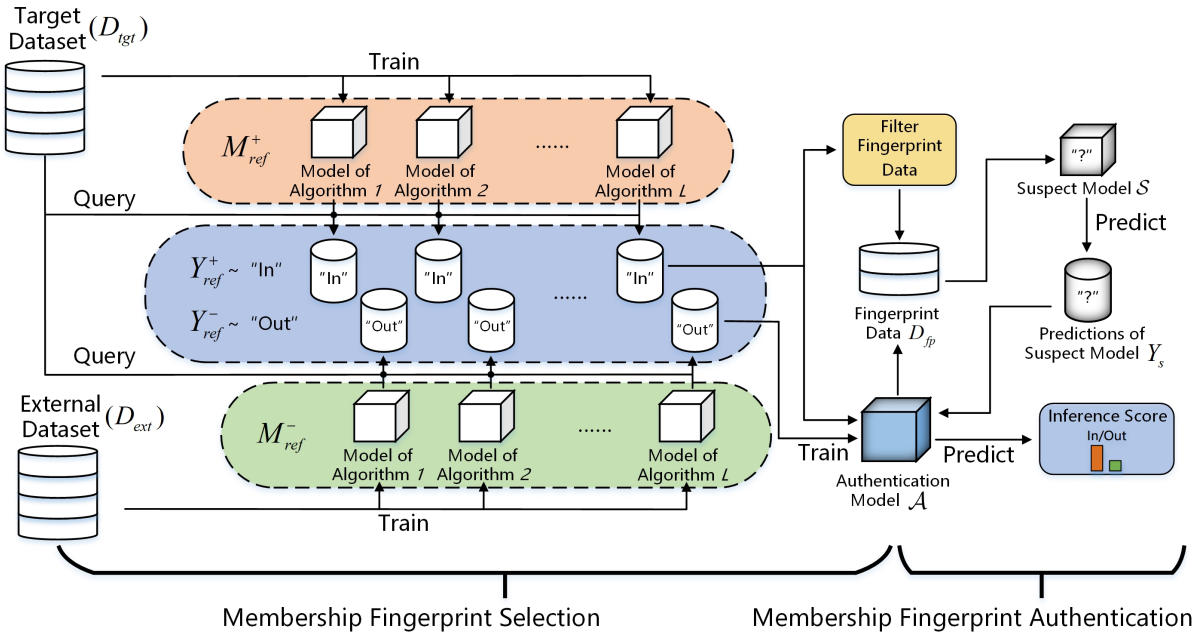


Fig. 3. The framework of MeFA. In membership fingerprint selection step, MeFA first trains a set of reference models $\mathcal{M}^+_{ref}$ (resp. $\mathcal{M}^-_{ref}$) on the target data D_{tgt} (resp. the external dataset D_{ext}). Then MeFA trains an authentication model $\mathcal{A}$ on Y^+_{ref} and Y^-_{ref} that are obtained from the reference models with D_{tgt}. MeFA uses $\mathcal{A}$ to select the fingerprints D_{fp} from D_{tgt} (c.f. Algorithm 1). In membership fingerprint authentication step, MeFA inputs the fingerprints to the suspect model $\mathcal{S}$ and gets the predictions Y_s, based on which MeFA can finally determine whether $\mathcal{S}$ is trained on D_{tgt} with the inference score of $Y_{\mathcal{S}}$ on $\mathcal{A}$.

differences in the prediction probabilities of different types of ML models (e.g. the prediction difference exists between a decision tree model and a linear regression model). Moreover, since the suspect model is usually deployed as a black-box, we cannot train an ML model that has the same type with the suspect model $\mathcal{S}$, and the above solution cannot achieve a high authentication accuracy.

Since each different training record has a different influence on the ML model and the objective of different ML algorithms is to fit the ML model to the training data, the influence of a training record should be similar or consistent across different models. Naturally, a part of the training data will have a significant influence on the prediction behavior of the models trained on it. With or without this part of the data participating in the training process, the prediction output of these models will have a significant difference. As a consequence, we can make use of a part of the target data records that are significant for the model behavior as the fingerprints to protect its ownership. Inspired by the study of MIAs [32] whose goal is to distinguish between the data used and not used to train a given model, *the more influence a training record has on the ML model's behavior, the easier the membership property of this record can be detected by MIAs.* Therefore, we can leverage the MIA technique to help us select the fingerprints from the target data.

To find the universal fingerprints of the target data that are valid for multiple types of models, we design a novel MIA that is targeted at multiple models simultaneously. According to the task of the target data, we select a set of commonly used learning algorithms to train multiple reference models on D_{tgt}, yielding a set of reference models $\mathcal{M}^+_{ref}$ (c.f. the orange dotted block in Fig. 3). For instance, concerning the classification dataset, we choose DNN, random forest and support vector machine to train the reference models.

However, with mere $\mathcal{M}^+_{ref}$, we cannot compare the difference of the model prediction behavior with and without training on D_{tgt}. Therefore, we also need a set of models not trained on D_{tgt}, to help us analyze the influence of each record in the target data on the model's behavior and select the fingerprints for D_{tgt}. To address this issue, we involve an *external dataset* D_{ext} that has the same format and value ranges of each feature with D_{tgt}. We use the same training settings of $\mathcal{M}^+_{ref}$ to construct another set of reference models $\mathcal{M}^-_{ref}$ (c.f. the green dotted block in Fig. 3).

The external dataset D_{ext} can be obtained through various methods. For example, we can add random noise to the target data D_{tgt} and get a noisy version which can be considered as the external dataset D_{ext}. In some situations, we may get the statistical information about the population from which D_{tgt} was drawn. Then we can generate D_{ext} by independently sampling the data records based on the statistics. In addition, with the development of the data generation technique, we can construct an auto-encoder or a generative adversarial network on D_{tgt} to generate a part of synthetic data, as in [42], [43], which can also serve as the external data.

2) Authentication Model Construction: Given the two sets of reference models $\mathcal{M}^+_{ref}$ and $\mathcal{M}^-_{ref}$, we need to select the record in D_{tgt} whose membership property is easy to be determined via MIAs. Then we can construct an authentication model based on the prediction of $\mathcal{M}^+_{ref}$ and $\mathcal{M}^-_{ref}$.

Specifically, the construction process of the authentication model $\mathcal{A}$ is as follows. For all data $\mathbf{x} \in D_{tgt}$, we compute

Algorithm 1 Membership Fingerprint Selection

Input: D_{tgt}, D_{ext}, a set of algorithms alg
Output: $D_{fp}, \mathcal{A}$

$\mathcal{M}^+_{ref}, \mathcal{M}^-_{ref} \leftarrow \emptyset$
$Y^+_{ref}, Y^-_{ref} \leftarrow \emptyset$
for each alg **do**
  $f_{tgt} \leftarrow Train(alg, D_{tgt})$
  $f_{ext} \leftarrow Train(alg, D_{ext})$
  $\mathcal{M}^+_{ref} \leftarrow \mathcal{M}^+_{ref} \cup f_{tgt}$
  $\mathcal{M}^-_{ref} \leftarrow \mathcal{M}^-_{ref} \cup f_{ext}$
  $Y_{tgt} = f_{tgt}(D_{tgt})$
  $Y_{ext} = f_{ext}(D_{tgt})$
  $Y^+_{ref} \leftarrow Y_{tgt} \cup Y^+$
  $Y^-_{ref} \leftarrow Y_{ext} \cup Y^-$
end for
Label $\mathrm{y}^+ \in Y^+_{ref}$ with "In", $\mathrm{y}^- \in Y^-_{ref}$ with "Out"
Train authentication model $\mathcal{A}$ with Y^+_{ref} and Y^-_{ref}
$D_{fp} = FingerprintFiltering(\mathcal{M}^+_{ref}, \mathcal{A}, D_{tgt})$
return $D_{fp}, \mathcal{A}$

Algorithm 2 Fingerprint Filtering

Input: $\mathcal{M}^+_{ref}, \mathcal{A}, D_{tgt}$
Output: D_{fp}

$k = 0$
$D_{fp} \leftarrow \emptyset$
for each $f_{tgt} \in \mathcal{M}^+_{ref}$ **do**
  $k = k + 1$
  $D^k_{fp} \leftarrow \emptyset$
  for each $\mathbf{x} \in D_{tgt}$ **do**
    $y = f_{tgt}(\mathbf{x})$
    if $\mathcal{A}(y) =$ "In" **then**
      $D^k_{fp} \leftarrow D^k_{fp} \cup \mathbf{x}$
    end if
  end for
end for
$D_{fp} = D^1_{fp} \cap D^2_{fp} \cap ... \cap D^k_{fp} \cap ...$
return D_{fp}

the prediction vector $\mathrm{y}+$ (resp. $\mathrm{y}-$) by querying each model in $\mathcal{M}^+_{ref}$ (resp. $\mathcal{M}^-_{ref}$) with $\mathbf{x}$. We label $\mathrm{y}+$ with "In" and $\mathrm{y}-$ with "Out". In MeFA, "In" means that the prediction is derived from the reference model trained on the target data, while "Out" has the opposite meaning. Next, we integrate all $(\mathrm{y}+, \mathrm{In})$ into one dataset Y^+_{ref}, and get Y^-_{ref} in the same way. Thereafter, we combine Y^+_{ref} and Y^-_{ref} together as the training set for our authentication model. Note that our attack is essentially a binary classification task, and thus we can make use of any classification algorithm to construct our authentication model. Our method is independent of the specific method used for authentication model training.

For the trained authentication model, we can feed it with a prediction vector of a model and obtain the probability of how it believes the record corresponding to this vector is the training data of the reference models $\mathcal{M}^+_{ref}$. In the next step of MeFA, we use this authentication model to authenticate the ownership of target data with respect to the suspect model.

3) Fingerprint Selection: In this paper, MeFA detects the embezzlement of the target data in accordance with the membership property of the records in D_{tgt} with respect to the suspect model. Therefore, the authentication model $\mathcal{A}$ essentially corresponds to the attack model for MIAs. As required by the MIAs [12], [13], [32], we need to get the prediction results of $\mathcal{S}$ on D_{tgt} which are then used to derive the detection results with the authentication model. However, the owner of $\mathcal{S}$ can easily detect the exception of massive queries. Besides, for the sake of saving our resources, it is not a wise choice to query all the records in D_{tgt}. According to MIAs [12], [32], the larger influence the record has on the model's prediction, the more sensitive to MIA this record is. In addition, when analyzing the prediction of $\mathcal{A}$ on the Y^+_{ref}, we observe that only a small number of records are classified as "In" by $\mathcal{A}$. Therefore, we can select the records that have a significant impact on all the reference models ($\mathcal{M}^+_{ref}$), i.e., the records which satisfy $\mathcal{A}(F(\mathbf{x})) =$ "In" for each model, as the membership fingerprints (denoted as D_{fp}). The selecting flow of the membership fingerprint is outlined in Algorithm 1.

C. Membership Fingerprint Authentication

Now that we have the membership fingerprints D_{fp} of the target data D_{tgt}, the next step is to detect whether the suspect model $\mathcal{S}$ infringes the IP of D_{tgt}.

To measure the membership property for a set of records, we propose a metric *Inference Score* which reflects how possible D_{fp} belongs to the training set of the suspect model $\mathcal{S}$:

$$\textit{Inference Score} = \mathbb{E}_{\mathbf{x} \sim D_{fp}}[\mathcal{A}(\mathcal{S}(\mathbf{x}))]$$

with its range in $[0, 1]$. If $\mathcal{S}$ is trained on the target data D_{tgt}, the inference score derived from $\mathcal{A}$ should be close to 1; otherwise, it will be close to 0.

By comparing the obtained inference score with a pre-determined threshold, we can easily determine whether $\mathcal{S}$ embezzles the target data by comparing with a pre-determined threshold. Therefore, we need to find a proper threshold θ_{thr} for the fingerprint authentication. Since we have two sets of reference models that are respectively trained on the target data and external dataset, we can determine θ_{thr} according to the Inference Score $\mathcal{A}(M^+_{ref}(D_{fp}))$ and $\mathcal{A}(M^-_{ref}(D_{fp}))$

By comparing $\mathcal{A}(M^+_{ref}(D_{fp}))$ with $\mathcal{A}(M^-_{ref}(D_{fp}))$, we find that a hard threshold 0.5 is not effective, and from the experiment results, we notice that θ_{thr} fluctuates with the similarity and complexity of D_{tgt} and D_{ext}. This indicates that θ_{thr} is not a fixed value, and should be determined by $\mathcal{A}(M^+_{ref}(D_{fp}))$ and $\mathcal{A}(M^-_{ref}(D_{fp}))$ when confronting different datasets. Here, we provide two feasible methods to determine θ_{thr}:

1) Traversing θ_{thr} to find a value that maximizes the metric:

$$\arg\max_{\theta_{thr}} \textbf{Metric}(\mathcal{A}(M_{ref}(D_{fp})), Y_{true})$$

where $\mathcal{A}(M_{ref}(D_{fp}))$ is the membership prediction ("In" or "Out") and y_{true} is the true membership label.

Here, the metric can be the accuracy, F1 score, etc, which goes up when $\mathcal{A}(M_{ref}(D_{fp}))$ and y_{true} are more approaching.

2) Simply setting θ_{thr} at the average of $\mathcal{A}(M^{+}_{ref}(D_{fp}))$ and $\mathcal{A}(M^{-}_{ref}(D_{fp}))$. According to MIA, a model predicts more confidently on its training data. This results in a higher $\mathcal{A}(M^{+}_{ref}(D_{fp}))$ and a lower $\mathcal{A}(M^{-}_{ref}(D_{fp}))$ since D_{fp} is the training data of M^{+}_{ref}. Hence, the average of them is a boundary that separates M^{+}_{ref} and M^{-}_{ref}.

At the end of MeFA, by comparing the determined θ_{thr} and the inference score of membership fingerprints, we can identify the ownership of the suspect model's training data. If the inference score is higher than the pre-determined threshold, MeFA will determine the suspect model has embezzled the target data. Otherwise, we will regard the suspect model as an innocent model.

V. Performance Evaluation

A. Experiment Setup

Datasets. We use three datasets commonly used in the previous works of MIAs [12], [32], [37] to evaluate the performance of MeFA.

UCI Adult[6]. The task of Adult dataset is to predict whether a person's income is over \$50K a year. It includes $48,842$ records with 14 features, such as age, education, and gender.

MNIST[7]. It is a handwritten recognition dataset that contains 10 classes of handwritten digits from 0 to 9. MNIST contains $70,000$ digits formatted as 28×28 gray images. The value of each pixel in the image is limited to $0 \sim 255$.

Purchases[8]. This dataset is based on Kaggle's "acquire valued shoppers" challenge dataset that contains shopping histories for thousands of individuals. Following [12], [32], [37], we use K-Means algorithm to cluster the dataset into $\{2, 10, 20, 50, 100\}$ classes. We use Purchase dataset mainly to evaluate the impact of the number of classes on the performance of our framework.

For each dataset, we randomly select $10,000$ records as the target data D_{tgt}, while randomly selected $10,000$ records from the rest serve as the external dataset D_{ext} and the remaining as the second external dataset $D^{'}_{ext}$ for training a part of suspect models. It is worth noting that there is no overlap between D_{tgt} and D_{ext}; however, the datasets used for training reference models in M^{+}_{ref} and M^{-}_{ref} can overlap with each other. In addition, for every dataset, we perform the data splitting operation multiple times and obtain different target/external splits. Then in our experiment section, we evaluate the performance of MeFA with respect to different data splits and report the averaged evaluation results.

Suspect Models. To demonstrate the universality of our framework, we evaluate MeFA on seven types of suspect models: *logistic regression (LR) with the penalty of L1 and L2, support vector machine (SVM), random forest (RF), XGBoost (XGB), decision tree (DT) and DNN*. For DNN models, we set the model architecture as a combination of one input layer, two hidden layers, and one output layer. It is worth noting that for different datasets, we set different numbers of hidden units in the two hidden layers. Specifically, for Adult dataset, the numbers of hidden units are 150 and 32, respectively, while for Purchase dataset, the numbers are 840 and 180, respectively. As for MNIST dataset, we follow the model architecture shown in its original source[9], where we use the architecture with two hidden layers (500 and 150 hidden units, respectively) to construct suspect models. For all deep models, we use Cross Entropy to calculate the training loss and set the learning rate of stochastic gradient descent (SGD) to 0.01. For each type of models, we use 10 different training settings to construct 10 sets of suspect models, and each set contains 2 models trained on the target data D_{tgt} and the second external dataset $D^{'}_{ext}$, respectively. In our experiments, we regard the suspect models trained on D_{tgt}, which accounts for half of all suspect models, as the models that indeed embezzle the target data. As a consequence, the baseline accuracy of the authentication is 0.5 which equals the effect of the *random guess*.

Reference Models. We also choose the above seven types of ML algorithms to train our reference models. For each type of algorithms, we use the same training settings to construct 2 models on the target data D_{tgt} and the external data D_{ext}, yielding the reference models corresponding to $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$, respectively. Then we leverage these 7×20 models to identify the membership fingerprints for the target data.

Metrics. The task of MeFA essentially is a binary classification problem. Thus we use the standard *Precision* and *Recall* metrics to measure the performance of MeFA. Precision represents the fraction of suspect models detected by MeFA that indeed embezzle the target data. Recall represents the fraction of the models that embezzle the target data that are correctly detected.

Baselines. We present two baselines for comparisons. The first baseline directly leverages all records in the target data D_{tgt} as the membership fingerprints; the second one simply compares the prediction probabilities of the suspect model $\mathcal{S}$ with those of the reference models. We use KL-divergence to measure the prediction similarity between the suspect model and the reference models. Then we determine whether $\mathcal{S}$ embezzles the target data according to the training data's source of the reference model which has the most similar prediction with the suspect model.

B. Threshold Identification

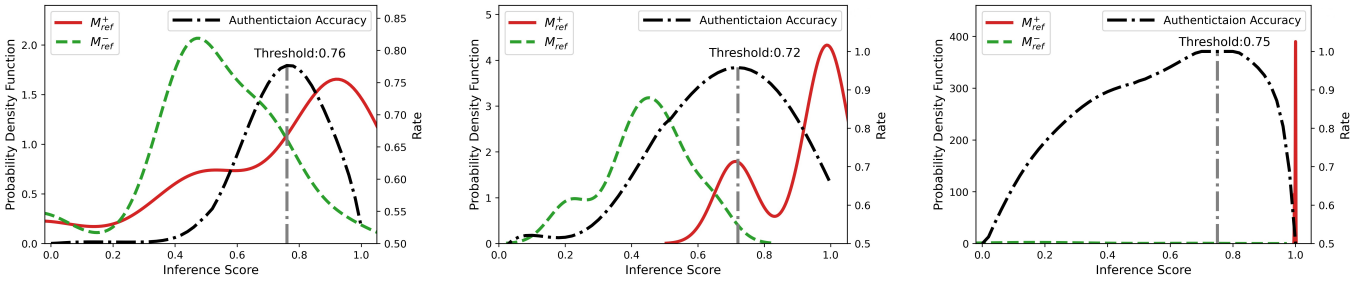
The threshold θ_{thr} is a vital parameter that MeFA relies on to determine the final authentication results, so we first show how to set θ_{thr} according to the difference of inference score derived from the reference models trained on the target and external data. In our experiments, we assume that the target data embezzlement occurred if *Inference Score* $> \theta_{thr}$. To find a preferable θ_{thr}, we choose an increasing rate of 0.01 and vary it from 0 to 1 thus getting a batch of authentication

[6]http://archive.ics.uci.edu/ml/machine-learning-databases/adult/

[7]http://yann.lecun.com/exdb/mnist/

[8]https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

[9]http://yann.lecun.com/exdb/mnist/



(a) Adult (b) MNIST (c) Purchase(50)

Fig. 4. Probability density function (PDF) of the inference score with respect to different datasets. For better observation, we use Gaussian kernel to process the original distribution.

TABLE I
COMPARISONS OF AUTHENTICATION PRECISION

Dataset	Algorithm	XGB	DT	LR-L1	LR-L2	SVM	DNN	RF	Average
Adult	Baseline-1	0.0%	100.0%	0.0%	0.0%	100.0%	100.0%	0.0%	42.86%
	Baseline-2	50.0%	50.0%	0.0%	0.50%	55.56%	33.33%	65.50%	43.06%
	MeFA	100.0%	50.0%	100.0%	100.0%	100.0%	50.0%	100.0%	**85.70%**
MNIST	Baseline-1	100.0%	60.0%	40.0%	100.0%	100.0%	0.0%	0.0%	57.14%
	Baseline-2	100.0%	100.0%	50.0%	50.0%	100.0%	50.0%	100.0%	78.57%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**
Purchase (50)	Baseline-1	100.0%	100.0%	0.0%	100.0%	100.0%	0.0%	100.0%	71.43%
	Baseline-2	100.0%	100.0%	100.0%	100.0%	100.0%	50.0%	100.0%	92.86%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**
Purchase (100)	Baseline-1	100.0%	100.0%	0.0%	100.0%	100.0%	100.0%	100.0%	85.71%
	Baseline-2	100.0%	100.0%	100.0%	100.0%	100.0%	50.0%	100.0%	92.86%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**

accuracy among the reference models with different θ_{thr}. Among these values of authentication accuracy, the one that brings the maximal value is the final identified θ_{thr}.

From Fig. 4 we can see that for each dataset, the inference score of membership fingerprints derived from $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$ are quite different. The average inference score derived from $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$ are 0.836 and 0.574 for Adult, 0.912 and 0.559 for MNIST, 0.995 and 0.256 for Purchase(50), and 0.998 and 0.242 for Purchase(100). To precisely separate the inference scores of $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$, we traverse the possible values of the threshold θ_{thr} and choose θ_{thr} according to the corresponding authentication accuracy on the reference models. From the experiment results shown in Fig. 4 we can see that, with the threshold increasing from 0, the authentication accuracy also gradually increases. When the threshold reaches a certain point, MeFA can achieve the highest authentication accuracy at this time. Correspondingly, we choose this threshold as θ_{thr}. In this way, we obtain θ_{thr} at 0.76 for Adult, 0.72 for MNIST, 0.75 for Purchase(50), and 0.81 for Purchase(100) (note that similar θ_{thr} can be obtained by simply setting θ_{thr} at the average of $\mathcal{A}(M^{+}_{ref}(D_{fp}))$ and $\mathcal{A}(M^{-}_{ref}(D_{fp}))$).

It can be also observed that as for the authentication accuracy, the dataset with more classes will get a more significant distinction difference between the target data and the external dataset. This result can explain why MeFA is more precise and robust in complex classification tasks, as will be shown in the following results.

C. Performance of MeFA

We next start to evaluate the performance of MeFA. We compare MeFA with the aforementioned two baselines, and the authentication precisions and recalls are shown in Tables I and II, respectively.

Taking MNIST dataset for example, it is observed that MeFA could achieve a mean precision and recall of 100% and 94.29%, respectively, which are significantly better than the baselines. The performance of the first baseline is close to the random guess which has a mean precision of 50%. This demonstrates that not all data in D_{tgt} are suitable to be the input of the authentication and the membership fingerprints selected by MeFA is effective. As for the second baseline, it can achieve a mean precision and recall of 78.57% and 87.14%, respectively, which stay between the first baseline and MeFA. Regarding to the performance on different types of suspect models, we can see that there exists an apparent gap among them. For the authentication precision and recall of the first baseline, three types of models including XGB, LR-L2, and SVM can reach 100%, while DNN and RF models only get 0. Even for MeFA, the authentication recall of RF is only

TABLE II
COMPARISONS OF AUTHENTICATION RECALL

Dataset	Algorithm	XGB	DT	LR-L1	LR-L2	SVM	DNN	RF	Average
Adult	Baseline-1	0.0%	100.0%	0.0%	0.0%	100.0%	100.0%	0.0%	42.86%
	Baseline-2	100.0%	100.0%	0.0%	100.0%	100.0%	40.0%	100.0%	77.14%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**
MNIST	Baseline-1	100.0%	37.5%	57.14%	100.0%	100.0%	0.0%	0.0%	56.38%
	Baseline-2	100.0%	100.0%	60.0%	50.0%	100.0%	100.0%	100.0%	87.14%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	60.0%	**94.29%**
Purchase (50)	Baseline-1	100.0%	100.0%	0.0%	100.0%	100.0%	0.0%	100.0%	71.43%
	Baseline-2	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**
Purchase (100)	Baseline-1	100.0%	100.0%	0.0%	100.0%	100.0%	100.0%	100.0%	85.71%
	Baseline-2	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	MeFA	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	**100.0%**

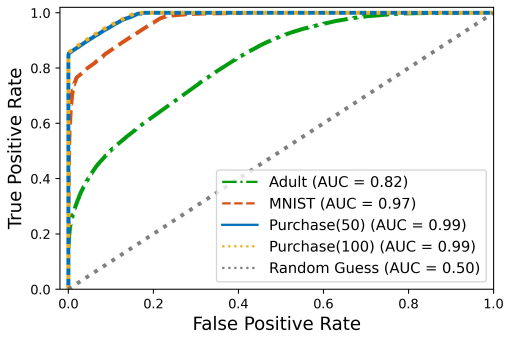
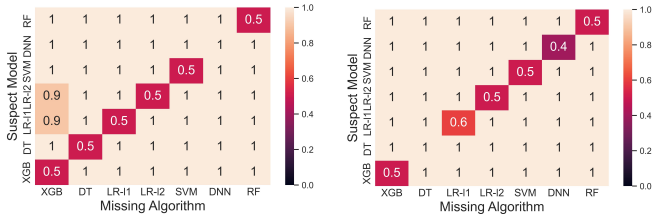


Fig. 5. ROC curves for different datasets.



(a) MNIST (b) Purchase(50)

Fig. 6. The impact of missing an algorithm that the reference models can take. The horizontal axis represents the algorithm that the suspect model uses, and the vertical axis represents the missing algorithm in reference models.

60%, which is 40% lower than other algorithms. One possible reason is that the prediction distribution of the fingerprints obtained from RF models is different from those of other algorithms. This indicates that there exists an incompatibility among algorithms. For the deficiency of the second baseline, it can be ascribed to the fact that the predictions of KL-divergence between suspect models are more affected by algorithms instead of membership property.

Similar performance can be observed in other three datasets. In general, the performance of the dataset which has more features and output dimensions can achieve a higher authentication precision. It can also be observed that MeFA exhibits steady robustness no matter how the suspect model performs on its original classification task. Especially for the suspect model trained on Purchase(100) dataset, even though the accuracy of the training accuracy varies from 20.86% to 100%, MeFA can always achieve an authentication precision of 100%.

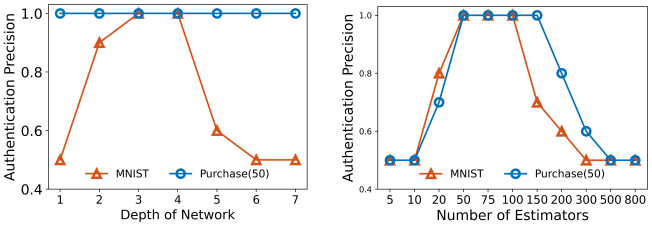
We further depict the ROC curves of MeFA for the four datasets, as shown in Fig. 5. From the results, we can see that these ROC curves of MeFA are close to the coordinate of $(0, 1)$, indicating MeFA performs much better than the random guess. With increasing complexity of the dataset, the AUC score also raises up accordingly. Especially for MNIST and Purchase, MeFA can achieve an AUC score of 0.97 and 0.99, respectively.

D. Impact of Different Factors

1) Impact of the Type of Reference Models: In MeFA, we construct multiple types of reference models for the authentication model. To evaluate the impact of the type of reference models, we successively remove one type of reference models and then reconstruct the authentication model.

Fig. 6 depicts the authentication precision of MeFA when missing one certain type of ML models in reference models. We can find that in MNIST and Purchase(50), MeFA only achieves the (lowest) precision 40% or 50% on the missing model types. As for the rest types, MeFA can achieve an authentication precision of 100%. Compared with the impact of other factors, the experiment results reveal that the considered types of reference models play a more important role in MeFA. To enhance the robustness of MeFA, we need to add more model types to train our reference models, in order to increase the probability that the reference models can cover the types of suspect models. To achieve this point, we should consider the domain information hidden behind the target data during the adding process, and thus limit the chosen range of added reference model types.

There is also an interesting observation: the inference score of the target data mainly concentrates around 1, but that of the external dataset is around 0.45 rather than 0. One possible reason is that, although we build quantities of suspect models based on target and external datasets respectively, the two datasets are sampled from one source dataset and their



(a) DNN (b) RF

Fig. 7. The relationship between the decisive hyper-parameter and the authenticate accuracy. We control the hyper-parameter within an appropriate range and measure how it affects MeFA.

TABLE III
PERFORMANCE FOR DIFFERENT NUMBERS OF CLASSES

No. of Classes	2	10	20	50	100
Authent. Precision	85.7%	100%	100.0%	100%	100%

distributions are almost the same. If these two datasets differ more significantly, the inference score of the external dataset may become much lower than 0.45.

2) Impact of the Structure of Suspect Model: From the perspective of the suspect model, the model structure's influence on the prediction becomes an uncertain factor for MeFA. In this part, we vary DNN and RF suspect models' hyper-parameters which control the model structure to evaluate the impact of the structure of the suspect models. We leave the training settings of the reference models unchanged, and then measure the authentication accuracy of MeFA. Fig. 7 shows how decisive hyper-parameter affects MeFA. For DNN model, when varying the depth of the network, the authenticate accuracy of Purchase(50) is stubbornly staying at 100%, while MNIST dataset is perfectly authenticated at the range of 2 to 4 (the depth of DNN model in $\mathcal{M}^{+}_{ref}$ is 3). Meanwhile, when we set the number of estimators of the RF model in $\mathcal{M}^{+}_{ref}$ at 100, the accuracy of 100% is also limited to a range around 70 and 100, respectively. These results indicate that as the structure of the suspect model and that of the same type of reference model get closer, the authentication performance of MeFA becomes better.

3) Impact of the Number of Classes: The number of classes of the protected dataset contributes to how much information we can use to select the membership fingerprints. In this part, we measure how MeFA performs when Purchase(100) dataset is split into $2, 10, 20, 50, 100$ classes. The results shown in Table III illustrate that as the number of classes increases from 2 to 100, our authentication precision increases from 85.7% to 100%. One reason is that with more numbers of classes, the reference models can provide more information that we can utilize to select the membership fingerprints, and the authentication model can catch more membership information that the suspect model leaks.

In addition, for each algorithm and dataset with a different number of classes, we train multiple models with D_{tgt}, and derive the mean prediction uncertainty of D_{tgt} and D_{ext}, respectively, as shown in Fig. 8. It can be seen that the prediction uncertainty difference increases as the number of classes grows. Consequently, with the increasing number of classes, the membership attributes of fingerprints become easier to be distinguished, in which case MeFA could achieve a higher authentication accuracy.

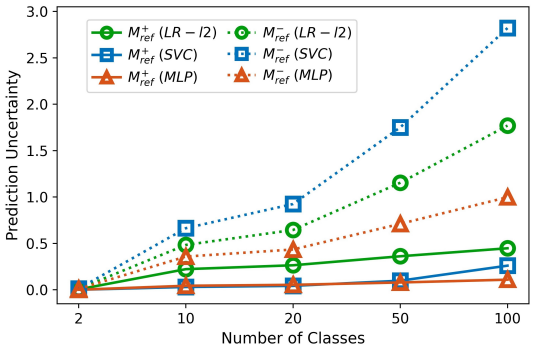


Fig. 8. Relation between the number of classes and the prediction uncertainty (information entropy).

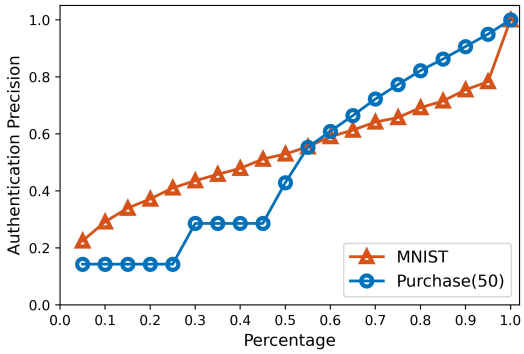


Fig. 9. The authentication precision as a function of percentage for MNIST and Purchase(50) datasets. The red line (MNIST) starts (when using only 5% of the dataset) at 22.44% and the blue line (Purchase(50)) at 14.29%, both of which are far lower than 100% (when using the whole dataset).

4) Impact of the Proportion of Dataset Used: In some cases, the attacker may not utilize the entire but only part of the target data. In this case, the subsequently trained model should be also determined as "has stolen the target data" under the definition of authentication in our setting. In this experiment, we use different percentage of the target data to train the suspect models, while leaving the training settings of the reference models unchanged. Specifically, we vary the percentage of the data used by the suspect model from 5% (500 records) to 100% with an inverval of 5%. For different percentage of data, we train 4×7 suspect models (4 for each algorithm) and evaluate the authentication results.

From the results in Fig. 9, we can see that the curve of MNIST rises steadily with the increasing percentage, while the curve of Purchase(50) stays constant in some interval. The result shows that a percentage above half is needed when MeFA surpasses the baseline (50%). This is because with the decreasing percentage, the originally valid fingerprint data once it becomes out of the selected proportion by the attacker, would be invalid. As a result, the authentication precision is positively correlated with the proportion of data used.

5) Impact of Model Type on the Inference Score: More concretely, the behavioral pattern reflects in the prediction of

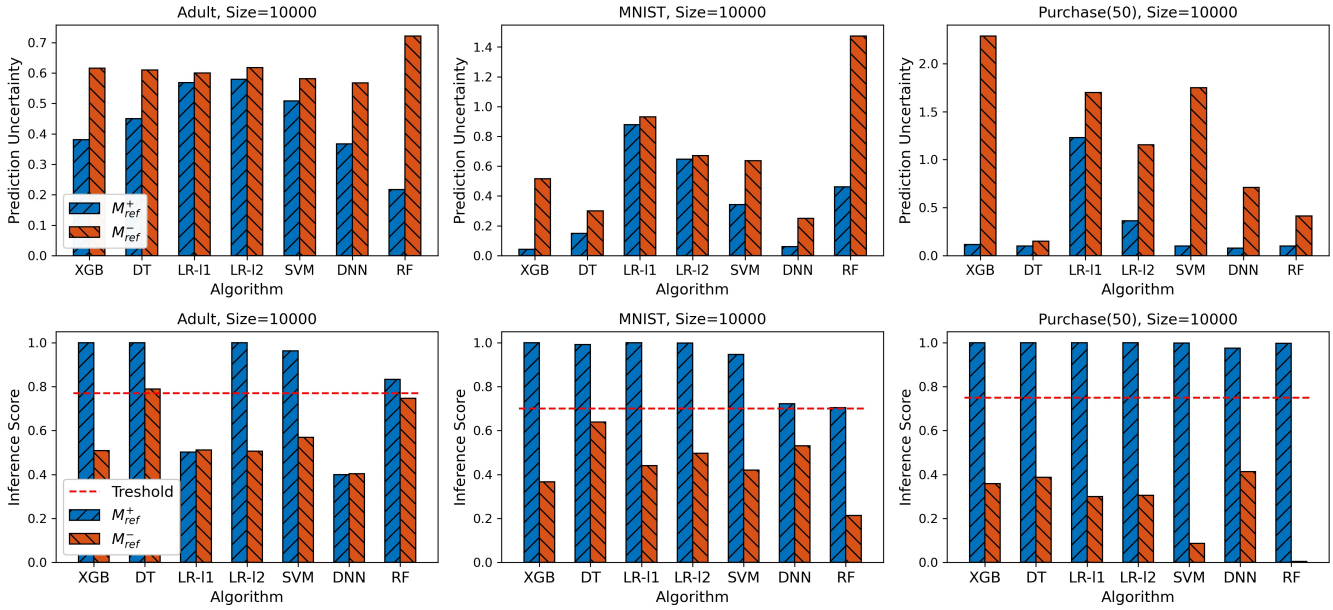


Fig. 10. The prediction uncertainty (top row) and inference score (bottom row) for suspect models of seven learning algorithms. The prediction uncertainty is the average information entropy for each data record. In all the three datasets, the prediction uncertainty distribution for each algorithm differs significantly, which, to some extent, increases the difficulty in our authentication for multiple learning algorithms.

the models. Therefore, we check the prediction uncertainty and inference score of all algorithms, and the results are shown in Fig. 10. The prediction uncertainty can be regarded as the cause of the inference score: the former measures the input of the authentication model and the latter is the output. From the top row, the difference of $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$ increases with the growing complexity of the dataset. Also, the different value of prediction uncertainty between algorithms explains why the factor— algorithm is so important in data authentication. Combined with the bottom row, we find that $\mathcal{M}^{+}_{ref}$ and $\mathcal{M}^{-}_{ref}$'s difference of prediction uncertainty determines that of inference scores. Comparing the three datasets, we can find that the large difference of prediction uncertainty causes the large difference of inference score, resulting in the higher authentication accuracy. It is therefore important to involve algorithms that are diverse enough in our framework to make our MeFA more robust and widely applicable.

E. By-Product: Model IP Protection

Since each ML model has its own uniqueness in the prediction behavior difference between its training and testing data, and the membership fingerprints extracted by MeFA for this model will be significantly different from the fingerprints of other models. Therefore, our membership fingerprints of a certain ML model can reflect the model uniqueness and be intuitively used to authenticate the IP of a black-box ML model. If most of the fingerprints are determined to belong to the given model's training set, then we can verify this model's IP to a large extent.

Therefore, we conduct extensive experiments and compare MeFA with a Blind-Watermark based model IP protection [8]. Table IV shows the comparison results. It can be seen that MeFA outperforms Blind-Watermark in complicated learning tasks. For the datasets whose dimension of features and labels is large, such as Purchase(50) and Purchase(100) with 600 features, the authentication precision of MeFA reaches 100%. Especially for the models trained on Purchase(100), the performance of MeFA is nearly twice as accurate as that of Blind-Watermark. In addition, since MeFA is a post-protection that does not modify the training process of ML models, it will not affect the performance of the protected model. The experiment results in Table IV also verify this point of view.

TABLE IV
COMPARISON OF MEFA AND BLIND-WATERMARK.

Dataset	MeFA		Blind-Watermark	
	Authenti. Precision	*Accuracy Decline*	*Authenti. Precision*	*Accuracy Decline*
Adult	85.70%	⁄	100.0%	0.41%
MNIST	94.29%	⁄	100.0%	2.73%
Purchase(50)	100.0%	⁄	62.60%	3.71%
Purchase(100)	100.0%	⁄	55.0%	3.0%

⁄: No accuracy decline on original classification task.

F. Robustness against MIA Defenses

The authentication process of MeFA essentially is to verify the membership property of the membership fingerprints with respect to the suspect model. Therefore, the attack may employ MIA defenses on the suspect model to escape our authentication. In order to evaluate the robustness of MeFA against MIA defenses, we implement the following three MIA defenses on the suspect model and evaluate the authentication performance of MeFA.

Differential Privacy. Differential privacy (DP) is a solution for publicly sharing information about a dataset by describing

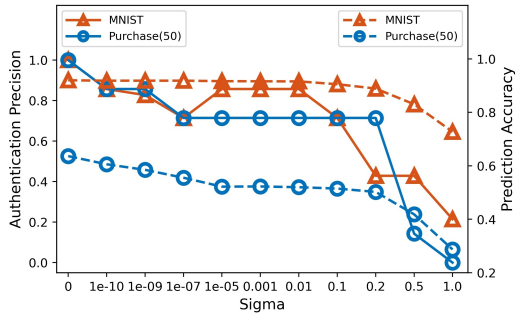


Fig. 11. Authentication accuracy and prediction accuracy for different noise levels on MNIST and Purchase(50) datasets. The parameter σ in Laplace noise ranges from 1×10^{-10} to 1.0.

the patterns of groups within the dataset while withholding information about individuals in the dataset [44], [45]. Due to its characteristics, differential privacy has been widely recognized as an effective defending technique against MIAs. Typical differential privacy based defenses often add differential noise to the training data or the prediction outputs of an ML model to evade the MIAs [46], [47].

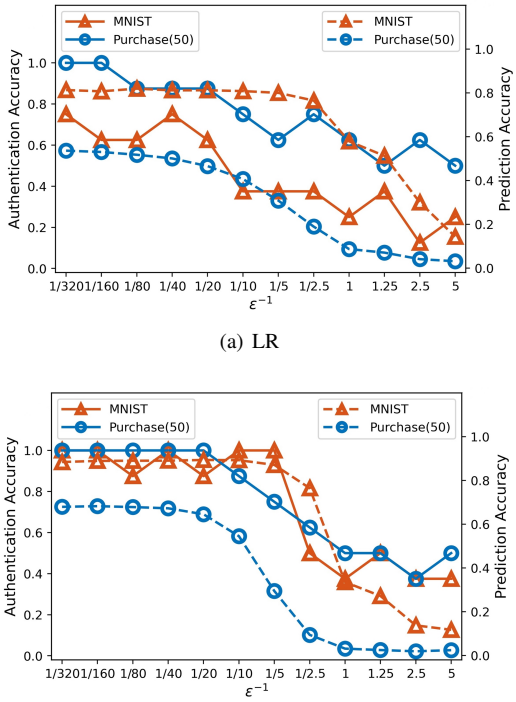
Restrict Prediction Output. The number of output classes of an ML model contributes to how much the model leaks. The more classes, the more information about the internal state of the model would be available to the attackers and the higher accuracy of MIAs can achieve. Therefore, restricting the prediction outputs is supposed to be an effective MIA defense method. For example, Shokri et al. [12] propose to add a filter to the model's output, making it merely outputs the probabilities of the most likely k classes. The smaller k is, the less information the model leaks.

Reduce Overfitting. Overfittinge is one major reason for the existence of the risk of MIAs [12], [32], [37]. Therefore, to defend against MIAs, many researchers have explored to reduce overfitting using L2-regularization [12], dropout [32], or model stacking [32] when training an ML model.

1) Robustness against DP: DP guarantees that any single data record in a dataset has limited impact on the output. Previous works have shown that DP mechanism can effectively prevent MIAs [12], [48], [49]. To validate whether DP can make the suspect models evade our authentication, we evaluate MeFA's performance against two different DP mechanisms.

For the first mechanism, we add Laplace noise on every attribute of the target data, as described in [47], [50], and then use the noised target data to train a suspect model. Specifically, we first scale the values in the target data to $[-1, 1]$ and then add 10 magnitudes of Laplace noise $S \sim La(0, \sigma)$ while varying σ from 1×10^{-10} to 1.0. For each value of σ, we test 7×6 models (6 for each kind of the training algorithm) which are all trained on target data.

Fig. 11 shows the authentication precision of MeFA and the prediction accuracy of suspect models concerning to different DP noise scales. It can be seen that the both authentication precision and prediction accuracy go down with σ increasing. Especially when σ grows from 0 to 1.0, there is a sharp dip to 78.58% (MNIST) and 100% (Purchase(50)) for authentication precision. As for the performance of suspect models,



(a) LR

(b) DNN

Fig. 12. Authentication accuracy of MeFA and prediction accuracy of suspect models for different privacy budget of DP-SGD.

the prediction accuracy dramatically decreases by 19.26% in respect of MNIST and 34.93% in respect of Purchase(50) respectively.

For the second mechanism, we follow the DP setting in the recent MIA studies [51], [52] and use DP-SGD [53], the most representative DP mechanism for training ML models. The core idea of DP-SGD is to add Gaussian noise to the gradients of an ML model and then use the noised gradients to update the model. In this section, we set the privacy budget ϵ of DP-SGD varying from 0.2 to 320. In addition, since DP-SGD can only be applied to the models that encounter gradient updating in the training process, we only report the results for the LR and DNN models trained on MNIST and Purchase (50) datasets. For each value of ϵ, we test 2×6 models as described above. The experiment results are shown in Fig 12. The results show that DP-SGD can effectively prevent our membership inference attack. It worth noting that DP-SGD can inevitably degrade the target model's accuracy.

Overall, a large DP noise, no matter it is added to the target data or to the model gradient, would reduce the utility of the trained ML models significantly. Therefore, the owner of the suspect models needs carefully tune the privacy budget parameters of DP mechanism to achieve a trade-off between privacy and model utility in practice. In this sense, the performance of MeFA is satisfactory in terms of the robustness against differential privacy techniques.

2) Robustness against Restriction of Prediction to Top-k Classes: For a more severe situation, the query access of the suspect model would not return the whole prediction vector to the inquirer. Instead, it may only show the top k

classes with their probabilities. To evaluate the impact of the restriction of prediction to top-k classes on the performance of MeFA, we perform experiments on MNIST and Purchase (50) datasets. Specifically, we train 70 suspect models (10 models for each algorithm) on the target data and external dataset respectively, while leaving the training settings of reference models unchanged. Then we restrict the prediction vector of the suspect models to top k classes to test if this meathod can evade the authentication of MeFA. For MNIST (resp. Purchase (50)) dataset, we set k to $\{1, 2, \cdots, 10\}$ (resp. $\{1, 2, \cdots, 50\}$).

According to the experiment results, we find that our authentication precision has an average decline of 1.43% only when k is set to 1. For the other choices of k, MeFA can achieve an authentication precision of 100% with more than top 2 (resp. top 1) prediction for MNIST (resp. Purchase(50)) dataset. It is easy to understand that the number k determines the input dimension of the authentication model. The larger the input dimension is, the more information leakage from the suspect model can be exploited.

3) Robustness against Overfitting: In this section, we adopt the dropout technique to the suspect models and obtain a series of models with different overfitting levels. More exactly, we design 160 models, each of which is different from others in the depth of the network, the rate of dropout or the number of neurons in a layer. Then we measure the performance of MeFA with respect to these suspect models.

Fig. 13 presents how overfitting affects our authentication result. It can be seen that those correctly authenticated models are restricted to a small scope, while the falsely authenticated models scatter more separately. On the other hand, the correctly authenticated models are also far from two groups of reference models. This may suggest that the authentication model implements its function with a new pattern rather than simply grasping the overfitting level of the suspect model. From the experiment results, we can see that the authentication performance of MeFA has a high tolerance to the overfitting level of the suspect model. Even there is a significant difference between the training and testing accuracy of the suspect model, MeFA can still verify its training data's IP precisely.

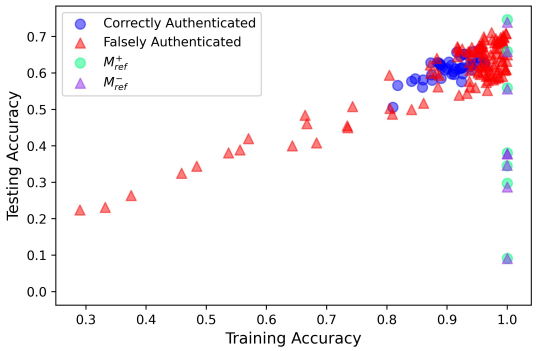


Fig. 13. The relationship between the authentication result of MeFA and the model overfitting level.

VI. Conclusion

In this paper, we have presented MeFA, a novel framework for detecting training data IP embezzlement in ML field. MeFA leverages MIA techniques to extract the membership fingerprints of the target data, which are then used to verify the ownership of the data. Extensive experiments on different datasets show that: 1) the membership fingerprints extracted by MeFA can be used to effectively measure the similarity of the prediction behavior for different models; 2) MeFA can be robust against different types of ML models regardless of the structure or setting the suspect model takes for training, and still works even when the training data is partially used or preproccessed with representative membership inference defenses; and 3) MeFA can also be used to verify the ownership of ML models (not limited to DNN models) without modifying the training process of the model. We believe that MeFA may deepen the understanding of the connection between the training data and the ML model, and opens a new pathway on data IP protection in the booming ML field.

In our future work, we plan to take into consideration ML models that only output predicted labels [35], [38], which are widely deployed in practical scenarios, and explore techniques that can verify the training data IP of such models. Another potential line of IP protection is to leverage internal characteristics of the target data. According to recent works about the unfairness of ML applications, the inherent bias existing in the training data will cause the ML models to output unfair results with respect to some records [54], [55]. By identifying the prediction unfairness, we may verify the IP of a suspect model's training data with a smaller cost.

References

[1] L. Deng and Y. Liu, *Deep learning in natural language processing*. Springer, 2018.
[2] M. Hassaballah and A. I. Awad, *Deep learning in computer vision: principles and applications*. CRC Press, 2020.
[3] S. Lee, C. Yoon, H. Kang, Y. Kim, Y. Kim, D. Han, S. Son, and S. Shin, "Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web," in *Proceedings of NDSS*, 2019.
[4] X. Fu, Y. Gao, B. Luo, X. Du, and M. Guizani, "Security threats to hadoop: data leakage attacks and investigation," *IEEE Network*, vol. 31, no. 2, pp. 67–71, 2017.
[5] J. Ning, J. Xu, K. Liang, F. Zhang, and E.-C. Chang, "Passive attacks against searchable encryption," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 789–802, 2018.
[6] M. Sun and W. P. Tay, "On the relationship between inference and data privacy in decentralized iot networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 852–866, 2019.
[7] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of ACM ASIACCS*, 2018, pp. 159–172.
[8] Z. Li, C. Hu, Y. Zhang, and S. Guo, "How to prove your model belongs to you: a blind-watermark based framework to protect intellectual property of DNN," in *Proceedings of ACM ACSAC*, 2019, pp. 126–137.
[9] X. Jia, X. Wei, X. Cao, and X. Han, "Adv-watermark: A novel watermark perturbation for adversarial examples," in *Proceedings of ACM Multimedia*, 2020, pp. 1579–1587.
[10] M. Xue, C. He, J. Wang, and W. Liu, "DNN intellectual property protection: Taxonomy, methods, attack resistance, and evaluations," *CoRR, arXiv: 2011.13564*, 2020.
[11] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of ACM ASIACCS*, 2021.
[12] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of IEEE S&P*, 2017, pp. 3–18.
[13] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," in *Proceedings of NDSS*, 2021.

[14] C. Song and R. Shokri, "Membership encoding for deep learning," in *Proceedings of ACM ASIACCS*, 2020, pp. 344–356.

[15] J. Guo and M. Potkonjak, "Watermarking deep neural networks for embedded systems," in *Proceedings of IEEE ICCAD*. IEEE, 2018, pp. 1–8.

[16] H. Chen, B. D. Rouhani, and F. Koushanfar, "Blackmarks: Blackbox multibit watermarking for deep neural networks," *CoRR, arXiv: 1904.00344*, 2019.

[17] X. Chen, W. Wang, Y. Ding, C. Bender, R. Jia, B. Li, and D. Song, "Leveraging unlabeled data for watermark removal of deep neural networks," in *Proceedings of ICML Workshop on Security and Privacy of Machine Learning*, 2019.

[18] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of ACM on International Conference on Multimedia Retrieval*, 2017, pp. 269–277.

[19] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks," in *Proceedings of ASPLOS*, 2019, pp. 485–497.

[20] M. Kuribayashi, T. Tanaka, and N. Funabiki, "DeepWatermark: Embedding watermark into DNN model," in *Proceedings of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2020, pp. 1340–1346.

[21] H. Wu, G. Liu, Y. Yao, and X. Zhang, "Watermarking neural networks with watermarked images," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[22] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Computer Science Review*, vol. 34, p. 100199, 2019.

[23] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," in *Proceedings of ACM AsiaCCS*, 2021, pp. 363–377.

[24] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proceedings of USENIX Symposium*, 2018, pp. 1615–1631.

[25] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proceedings of ICLR*, 2020.

[26] F. Boenisch, "A survey on model watermarking neural networks," *CoRR, arXiv: 2009.12153*, 2020.

[27] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in *Proceedings of USENIX Symposium*, 2021.

[28] H. Chen, B. D. Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proceedings of ICMR*, 2019, pp. 105–113.

[29] E. Le Merrer, P. Perez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020.

[30] P. Maini, M. Yaghini, and N. Papernot, "Dataset inference: Ownership resolution in machine learning," in *Proceedings of ICLR*, 2021.

[31] H. Hu, Z. Salcic, G. Dobbie, and X. Zhang, "Membership inference attacks on machine learning: A survey," *CoRR, arXiv:2103.07853*, 2021.

[32] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proceedings of NDSS*, 2019.

[33] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of ACM CCS*, 2021.

[34] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, "SocInf: Membership inference attacks on social media health data with machine learning," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 907–921, 2019.

[35] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proceedings of IEEE CSF*, 2018, pp. 268–282.

[36] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proceedings of IEEE S&P*, 2019, pp. 691–706.

[37] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proceedings of IEEE S&P*, 2019, pp. 739–753.

[38] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proceedings of ICML*, 2021, pp. 1964–1974.

[39] S. Saeidian, G. Cervia, T. J. Oechtering, and M. Skoglund, "Quantifying membership privacy via information leakage," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3096–3108, 2021.

[40] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of ACM CCS*, 2015, pp. 1322–1333.

[41] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proceedings of NeurIPS*, 2019, pp. 14 774–14 784.

[42] Q. Yu and W. Lam, "Data augmentation based on adversarial autoencoder handling imbalance for learning to rank," in *Proceedings of AAAI*, 2019, pp. 411–418.

[43] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis," in *Proceedings of IEEE/CVF CVPR*, 2021, pp. 5799–5809.

[44] C. Dwork, "Differential privacy," in *Proceedings of International Colloquium on Automata, Languages and Programming*, 2006, pp. 1–12.

[45] C. Dwork, F. Mcsherry, and K. Nissim, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of Theory of Cryptography Conference*, pp. 265–284.

[46] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of ACM CCS*, 2016, pp. 308–318.

[47] B. Zhang, R. Yu, H. Sun, Y. Li, J. Xu, and H. Wang, "Privacy for all: Demystify vulnerability disparity of differential privacy against membership inference attack," *CoRR, arXiv:2001.08855*, 2020.

[48] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *Proceedings of USENIX Security*, 2019, pp. 1895–1912.

[49] C. Wang, G. Liu, H. Huang, W. Feng, K. Peng, and L. Wang, "MIASec: Enabling data indistinguishability against membership inference attacks in MLaaS," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 3, pp. 365–373, 2020.

[50] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[51] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," *Proceedings of ACM CCS*, 2021.

[52] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," *Proceedings of ACM CCS*, 2021.

[53] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of ACM CCS*, 2016, pp. 308–318.

[54] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, 2021.

[55] M. Wang and W. Deng, "Mitigating bias in face recognition using skewness-aware reinforcement learning," in *Proceedings of IEEE/CVF CVPR*, 2020, pp. 9322–9331.