Demystifying Membership Inference Attacks in Machine Learning as a Service

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei

Abstract—Membership inference attacks seek to infer membership of individual training instances of a model to which an adversary has black-box access through a machine learning-as-a-service API. In providing an in-depth characterization of membership privacy risks against machine learning models, this paper presents a comprehensive study towards demystifying membership inference attacks from two complimentary perspectives. First, we provide a generalized formulation of the development of a black-box membership inference attack model. Second, we characterize the importance of model choice on model vulnerability through a systematic evaluation of a variety of machine learning models and model combinations using multiple datasets. Through formal analysis and empirical evidence from extensive experimentation, we characterize under what conditions a model may be vulnerable to such black-box membership inference attacks. We show that membership inference vulnerability is data-driven and corresponding attack models are largely transferable. Though different model types display different vulnerabilities to membership inference, so do different datasets. Our empirical results additionally show that (1) using the type of target model under attack within the attack model may not increase attack effectiveness and (2) collaborative learning exposes vulnerabilities to membership inference risks when the adversary is a participant. We also discuss countermeasure and mitigation strategies.

Index Terms—membership inference; federated learning; data privacy

1 Introduction

MACHINE learning-as-a-service has seen an explosion of interest with the development of cloud platform services. Amazon [1], Microsoft [2], IBM [3], and Google [4] have all launched such machine learning-as-a-service platforms. These services allow companies to leverage powerful machine learning and artificial intelligence technologies without requiring in-house domain expertise. Machine learning-as-a-service platforms allow users to upload their data, run various data analytics or model building processes, and deploy trained models to services of their own. Given this landscape, new interest has been given to the potential vulnerabilities of such machine learning services.

One such vulnerability is membership inference. Let us consider a cancer treatment center with a large database of valuable patient data. Let our cancer treatment center then leverage a machine learning-as-a-service platform to develop a predictive model which, when given a patient's data as input, can predict cancer-related health outcomes. The treatment center then utilizes the cloud deployment option to create a service of their own wherein users can log in, provide their own health information, and receive predictions in return. A membership inference attack considers a scenario wherein a user of such a black-box prediction service is an adversary. This adversary can provide the health information of another individual X and, based on the model's output, try to infer if X is a cancer patient at the treatment center.

There are two primary parties who are interested in protecting against such membership inference attacks: patient X and the cancer treatment center. Previous patients of the cancer treatment center, such as patient X, consider their membership private and do not want their patronage to be public knowledge. For example, consider the case of a treatment center patient, Alice. Let Alice be under consideration for a job at Bob's company. Bob can leverage the cancer treatment center's service to infer whether or not Alice is a patient. Upon learning of Alice's inclusion in the cancer treatment center's database, Bob decides not to hire Alice in favor of a candidate who, he believes, will have lower healthcare costs for the company.

In addition to concerns of patients such as Alice, we also must consider the interests of the cancer treatment center, the owner of the training dataset and the trained model under attack. In today's market, across many domains, data is considered an organizational asset [5]. While internal data has always been a driver of decision making for most companies, the role of data has been moving steadily closer to the core of many industries. A company's data therefore holds intrinsic value to the organization. Additionally, this training data is the source of the machine learning model under attack. Training this model not only requires front-end capital, time, and resources but also holds competitive business value for the treatment center. The treatment center may even charge a fee per evaluation. It is therefore essential from the cancer treatment center's perspective to protect their private database.

Membership Inference Risks v.s. Differential Privacy. Membership inference violates the privacy of both the individual participants involved in the model training and the owner of the training dataset. The former involves membership privacy of individuals who are participants in the model training and the latter involves risks of unauthorized

- *S.Truex, L. Liu, M.E. Gursoy, L. Yu, and W. Wei are with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, 30332. Email: {staceytruex, memregursoy, leiyu, wenqiwei}@gatech.edu, ling.liu@cc.gatech.edu*

Manuscript received X; revised Y.

leakages of business value or trade secrets. The ultimate goal of membership privacy is to protect against the risk of membership leakage of individuals in the data used in training a machine learning model.

When a model is secured by differential privacy, there exists some formal guarantee that the model trained on the original dataset D will produce statistically similar predictions as a model trained on D', which differs from D by exactly one instance [6]. Differential privacy therefore protects the content privacy and the output privacy of the model, often with some cost to model accuracy. Membership privacy refers to membership inference against machine learning models and is centered on inferring the membership of the input data, but not the content of input data, from the output result of the model.

Membership Inference v.s. Adversarial Examples. Adversarial learning to-date has been focused on attacking deployed deep learning models. Most existing membership inference attacks similarly attack deep learning models, utilizing deep neural networks (DNNs) for training both the target model under attack and the attack model [7], [8], [9], [10]. However, membership inference attacks are different from adversarial examples with respect to both the attack generation process and the adverse effects of successful attacks and therefore represent a separate class of security and privacy intrusion problems under the general umbrella of adversarial machine learning. Concretely, adversarial deep learning research to date has been centered on the generation of adversarial examples by injecting the minimal amount of perturbation to a benign example required to cause a pre-trained classification model to misclassify the example with high probability. Thus, adversarial example-based attacks aim at altering the output of the model prediction without being visually noticed. On the contrary, a membership inference attack does not alter the prediction output at all and succeeds by simply making membership inference on the prediction output.

Scope and Contributions of the paper. In this paper, we investigate membership inference attacks under a black-box access scenario in which an adversary may only probe the prediction API with input and receive the corresponding prediction output from the privately trained model. Our research results are novel from three perspectives. First, we describe a systematic approach to construct a membership inference attack model with a general formulation of each component of the attack model generation framework. We show that generating a membership inference attack model is a complex and multi-step strategic process. Second, to understand *when* and *how* membership inference attacks work and *why* certain models and datasets are more vulnerable, we take a holistic approach with extensive empirical evidence to study and characterize membership inference attacks across different target model types, different types of training datasets, and different combinations of model types for generating attack training datasets and attack models. Finally, we introduce and investigate a new membership threat, insider membership inference, which is launched by a member of a federated learning system against other participants in an collaborative learning environment. As federated learning systems become more popular with promises of increased accuracy and privacy, highlighting and understanding this risk is an important part of the membership inference mitigation effort.

2 Membership Inference Attacks

In this section, we formalize membership inference attacks against machine learning models as follows: Given an instance $\mathbf{x}$ and black-box access to a classification model F_t trained on a dataset D, can an adversary infer with high confidence that the instance $\mathbf{x}$ was contained in D at the train time of F_t? This definition states that membership inference focuses on the question of the membership of $\mathbf{x}$ in D and *not* about the contents of $\mathbf{x}$. This divergence separates membership inference from existing areas of privacy research, such as differential privacy [11], [12], [13] or secure multiparty computation [14], [15], [16]. Also notable is that membership inference attacks are at the local level: an adversary wishes to know if a particular $\mathbf{x}$ is in D and *not* D in its entirety.

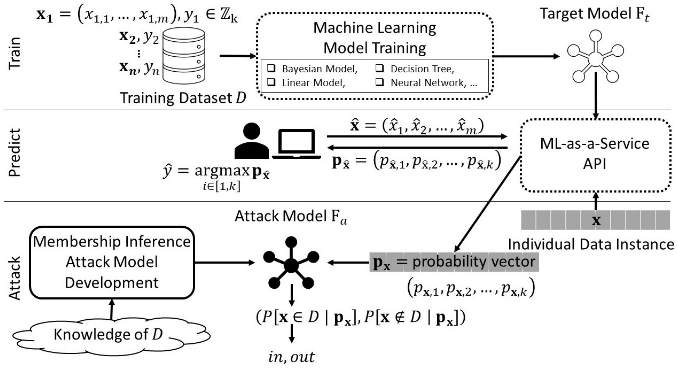
Train
Predict
Attack
$\mathbf{x_1} = (x_{1,1}, \dots, x_{1,m}), y_1 \in \mathbb{Z}_k$
$\mathbf{x_2}, y_2$
$\mathbf{x_n}, y_n$
Training Dataset D
Machine Learning Model Training
❑ Bayesian Model,
❑ Decision Tree,
❑ Linear Model,
❑ Neural Network, ...
Target Model F_t
$\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$
$\mathbf{p}_{\hat{\mathbf{x}}} = (p_{\hat{\mathbf{x}},1}, p_{\hat{\mathbf{x}},2}, \dots, p_{\hat{\mathbf{x}},k})$
$\hat{y} = \underset{i \in [1,k]}{\text{argmax}}\, \mathbf{p}_{\hat{\mathbf{x}}}$
ML-as-a-Service API
Attack Model F_a
$\mathbf{x}$
Individual Data Instance
Membership Inference Attack Model Development
$\mathbf{p_x}$ = probability vector
$(p_{\mathbf{x},1}, p_{\mathbf{x},2}, \dots, p_{\mathbf{x},k})$
$(P[\mathbf{x} \in D \mid \mathbf{p_x}], P[\mathbf{x} \notin D \mid \mathbf{p_x}])$
Knowledge of D
in, *out*

Fig. 1. The workflow of a Membership Inference Attack.

Figure 1 illustrates the workflow of membership inference attack development. Given a training dataset D and a classification model F_t trained on D, the machine learning service provider may provide a classification service through a prediction API. This API offers users black-box access to the model F_t. Users may send prediction queries with their own data to the service and receive classification predictions. An adversary uses such a service to collect information about the private dataset D on which the prediction model F_t was privately trained. By leveraging any public or background knowledge of the training dataset D or the target model F_t, an adversary builds a membership inference attack model F_a to deploy for launching membership inference attacks in real time.

To gain an in-depth understanding of the general formulation of the membership inference attack model, we first characterize the types of adversarial knowledge and datasets required to train the attack model as well as the attack cost, the attack value, and their evaluation metrics. We then present a systematic formulation of general attacks in Section 3.

2.1 Threat Model and Assumptions

2.1.1 *Machine Learning-As-A-Service: Black-box Access*

Recall from Figure 1 that the machine learning service provider publishes the trained target model F_t through a black-box access API, which accepts service requests from users in the form of a prediction query for input $\mathbf{x}$ and returns the predicted class and the prediction probability vector. The input and output formats of the API are given by the service provider as part of a service agreement. However, F_t and the dataset D on which F_t is trained remain private. Only the prediction API is exposed to users, thus ensuring only black-box access to F_t.

A question one may ask is: "How can an adversary with only black-box access to the prediction API perform membership inference attack without knowing anything about F_t and the dataset D?" Generally speaking, F_t is trained as an approximation of an ideal function $y = \Gamma(\mathbf{x})$ for a training dataset D, where y is the true class for the sample instance $\mathbf{x} \in D$. Let y_c denote the output of a candidate model F_c. An optimal model is then the one that minimizes the average loss defined by a chosen loss function $L(y, y_c)$ for all samples in the training set D, weighted by their *posterior* probability. The posterior probability, $P_y(y_c|\mathbf{x})$, is defined as the probability of class y_c being the label of sample $\mathbf{x}$. For many application specific problems, $y = \Gamma(\mathbf{x})$ is a non-deterministic function. That is, if $\mathbf{x}$ is sampled repeatedly, different values of y may be given. In this case, the optimal choice of the class for sample object $\mathbf{x}$ among all candidate class labels is the class that minimizes the expected loss for a given sample $\mathbf{x}$. The target model F_t is then assigned to be the optimal model for the training dataset D upon the completion of the training and testing phases.

There are currently a multitude of well-studied algorithms available to determine F_t. Without loss of generality to which algorithm or loss function was chosen to identify F_t, we simply maintain that F_t is a function which maps feature vectors $\mathbf{x} \in \mathbb{R}^m$ to a class $y \in \mathbb{Z}_k$, $k \geq 2$. Our target function therefore creates a decision boundary which separates the feature space $\mathbb{R}^m$ into k sets in which each set is associated with a candidate class value in $\mathbb{Z}_k$. As these sets and corresponding class assignments are chosen to minimize the loss function $L(y, y_c)$ over the training dataset D, the decision boundaries are strongly informed by the training dataset D and will in turn be the core of the trained machine learning model F_t.

2.1.2 Adversarial Knowledge

We characterize the membership inference threat model based on prior adversarial knowledge. We broadly categorize this adversarial knowledge into three categories: black-box, grey-box, and white-box data knowledge.

Black-Box Knowledge. An adversary is said to have black-box knowledge when the adversary does not have any *specialized* knowledge of the training data. However, black-box knowledge may include the input and output of the service API as well as publicly available information about the target prediction model F_t. For example, if the service provider is our cancer treatment center, then the adversary may have access to relevant statistics curated by the government and published for the public good including demographic information, such as the likeliness of different age groups or genders to contract certain cancers, or clinical information, such as the prevalence of co-occurrence of different diseases with various cancer types.

Grey-Box Knowledge. We characterize grey-box knowledge as *specialized* population-level knowledge. This may include population-level statistics that describe the distribution of features in the target model's training data. For example, in addition to publicly available distributions on the average age of cancer patients (black-box knowledge), the adversary may know the average age of a cancer patient seen at the target treatment center (*specific* knowledge).

White-Box Knowledge. White-box knowledge characterizes scenarios where the training data for F_t is sampled from a constrained population or in a skewed fashion such that an adversary has access to some versions of real data in the training data D of the target model F_t or some leaked portion of D but not the complete training set D. For example, a noisy version of the real data may be accessible which resembles D with the addition of some noise or missing values [7]. Adversaries with white-box knowledge can therefore develop or access **true** "`in`" samples and employ active learning techniques on these known samples to develop a very accurate dataset to mirror D.

Adversaries with white-box knowledge are the most powerful whereas adversaries with only black-box knowledge represent the most difficult attack environment wherein adversaries are limited to (i) publicly availbale information, (ii) black-box queries to the prediction API, and (iii) the output of classification prediction from the target model F_t. This is the setting we use to formulate membership inference attacks and to characterize adverse effects and divergence of membership inferences.

2.2 Attack Value vs. Attack Cost

It is generally accepted that systems security should never operate as an all-or-nothing mechanism. Systems must always seek to optimize two sets of factors: cost of defense vs value of assets to the system owner and cost of attack vs value of assets to the adversary. These principles hold true to deployed learning systems with respect to membership inference attacks.

In the context of membership inference attacks, value can be characterized by attack accuracy as an evaluation of what level of leakage is present in F_t or what amount of knowledge an attacker can expect to gain. The cost of attack refers to the knowledge and work necessary for an adversary to launch a successful attack. When characterizing cost, we consider knowledge cost as well as development cost. For example, to launch an effective attack, how much knowledge of D or F_t does an adversary need? Gaining this knowledge should be considered a type of cost for the adversary. Alternatively, cost can also be characterized by how computationally expensive it is to develop an effective attack model F_a.

The accuracy of an attack can be characterized by a number of metrics. For example, an accuracy measure may be defined as the likelihood that F_a correctly identifies $\mathbf{x} \in D$ *or* $\mathbf{x} \notin D$. Alternatively, accuracy can be defined as a precision measure which indicates the fraction of the instances inferred as members that are indeed members of the training dataset D, focusing on the probability $Pr[\mathbf{x} \in D | F_a$ says $\mathbf{x} \in D]$.

3 General Attack Formulation

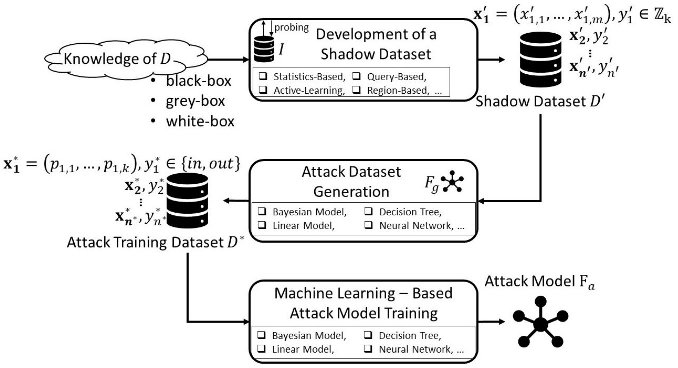
$\mathbf{x}'_1 = (x'_{1,1}, ..., x'_{1,m}), y'_1 \in \mathbb{Z}_k$
probing
I
Development of a Shadow Dataset
Knowledge of D
□ Statistics-Based, □ Query-Based,
□ Active-Learning, □ Region-Based, ...
$\mathbf{x}'_2, y'_2$
$\mathbf{x}'_{n'}, y'_{n'}$
Shadow Dataset D'
• black-box
• grey-box
• white-box
$\mathbf{x}^*_1 = (p_{1,1}, ..., p_{1,k}), y^*_1 \in \{in, out\}$
$\mathbf{x}^*_2, y^*_2$
$\mathbf{x}^*_{n^*}, y^*_{n^*}$
Attack Training Dataset D^*
Attack Dataset Generation
F_g
□ Bayesian Model, □ Decision Tree,
□ Linear Model, □ Neural Network, ...
Attack Model F_a
Machine Learning – Based Attack Model Training
□ Bayesian Model, □ Decision Tree,
□ Linear Model, □ Neural Network, ...

Fig. 2. Membership Attack Model Development.

Using correct identification to denote attack accuracy, the problem of membership inference is defined as follows: given a query input $\mathbf{x}$ and black-box access to the target model F_t, the membership inference attack answers the question of whether $\mathbf{x} \in D$ is true or false. The attack is successful if the attacker can determine with high confidence that $\mathbf{x} \in D$ is true or with high confidence that $\mathbf{x} \notin D$.

At the most abstract level, membership inference attack models are binary classifiers. Given an instance $\mathbf{x}$ and a target model F_t, the goal of a membership inference attack model is to identify whether or not $\mathbf{x}$ was contained within the dataset D used to train F_t.

Let D consist of n training instances $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_n}, y_n)$ where $\mathbf{x_i}$ consists of m features, denoted by $\mathbf{x_i} = (x_{i,1}, x_{i,2}, ..., x_{i,m})$, and $y_i \in \mathbb{Z}_k$, where k is a finite integer value ≥ 2. Let $F_t : \mathbb{R}^m \rightarrow \mathbb{R}^k$ be the target model trained using this dataset D. Given a particular *feature vector* $\mathbf{x} \in \mathbb{R}^m$, F_t will then output a probability vector $\mathbf{p} \in \mathbb{R}^k$ of the form $\mathbf{p} = (p_1, p_2, ..., p_k)$, where $p_i \in [0, 1] \forall i$, and $\sum_{i=1}^{k} p_i = 1$. The prediction class label y for a feature vector $\mathbf{x}$ is the class with highest probability value in $\mathbf{p}$. Therefore $y = \arg\max_{i \in \mathbb{Z}_k} F_t(\mathbf{x})$.

Given the adversary's black-box access to F_t via the prediction service API, an adversary is able to query F_t with any number of instances to receive corresponding probability vectors. The adversary uses this probing access, along with any prior knowledge, to generate I, a representation of adversarial knowledge of D. The first building block for implementing a black-box membership inference attack is to leverage I to generate a synthetic labeled dataset D' to mirror the data in D. This synthetic, labeled dataset D' is artificially simulated and called a *shadow dataset* of D. Although the word "shadow" was borrowed from shadow copying for systems creating back up data copies [17], the shadow dataset in our context should be thought of as a synthetic version of the real training dataset D. D' is then used to generate an attack training dataset D^*, which is required to train the final membership attack model, a binary classifier F_a.

Figure 2 highlights these three primary phases in the development of the membership inference attack: (1) development of a shadow dataset, (2) generation of an attack model training dataset, and (3) training and deployment of the membership inference attack model. In phase (1) the attacker's goal is to develop a dataset denoted as D' which closely emulates, or *shadows*, the dataset D which was used to train the target model. In phase (2) the attacker uses this shadow dataset to develop a shadow model which is considered to emulate the behavior of the target model. The attacker may then observe the behavior of this shadow model in response to instances which the attacker knows were given during training versus those that were not. This behavior is used to develop an attack dataset which captures the difference between output for instances in the training data and those previously unseen by the model. Finally, in phase (3), this attack data is used to generate a binary classifier which provides predictions on whether an instance was previously known to a model based on that model's output from that instance. We now discuss each of these phases in further detail.

3.1 Development of a Shadow Dataset

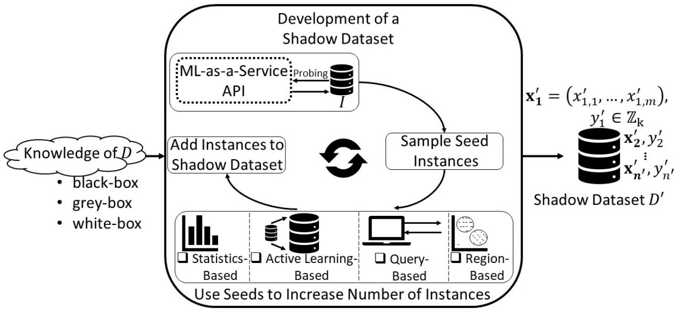

Fig. 3. Development of a Shadow Dataset.

Given a target model F_t, its training dataset D, and black-box adversarial knowledge , the development of a shadow dataset D' is the first step in generating a membership inference attack model. D' consists of n' training instances $(\mathbf{x'}_1, y'_1), (\mathbf{x'}_2, y'_2), ..., (\mathbf{x'_{n'}}, y'_{n'})$ where each $\mathbf{x'_i}$ consists of m features equivalent to those in D and each y'_i is a predicted class label in $\mathbb{Z}_k$. Note that k and m are known via the service API and thus consistent across D and D'. The cardinality of D, however, remains unknown and therefore n and n' are likely to differ. The shadow dataset generation process leverages the prediction service API to manage the creation and control the quality of D', as shown in Figure 3.

3.1.1 API Probing

While the training set D and the cardinality (size) n of D are unknown to an adversary, the adversary can probe the service API to reveal structural information, such as the number of features m, the data types of those features, and the number of classes k. This knowledge can be obtained by the adversary through sending in trial query instances and observing responses.

We refer to the complete set of adversarial knowledge as I. This includes prior knowledge as well as that inferred from this API probing. We do not state any limitations on I except that $D \not\subset I$, as the membership inference attack becomes trivial when $D \subset I$. The cost of launching an effective membership inference attack includes the work of this probing phase by the adversary.

As a result of this API probing, the adversary can construct a skeleton dataset D', which is similar to D in structure and, ideally, any $\mathbf{x}' \in D'$ should be a viable instance that could be included in D.

3.1.2 Shadow Data Generation

There are several ways to generate a quality shadow dataset with a small number of query probing attempts. Below we highlight four categories of techniques: statistics-based, active learning-based, query-based, and region-based generation.

Statistics-Based Generation. In statistics-based generation, the adversary leverages population-level statistics of the features in D to create samples for D'. Given known distributions for features, an adversary may conduct random sampling to construct these new samples. Features may be treated independently where an instance is generated through $m+1$ random samplings of $m+1$ distributions, each distribution corresponding to either a different feature or the class label. Alternatively, sampling may account for feature relationships. This may be done, for example, when our adversary has knowledge of statistics on disease co-occurrence.

Active Learning-Based Generation. Active learning is a technique developed in the semi-supervised machine learning domain [18]. Active learning has been developed to address the problem of a largely unlabeled training dataset when assigning accurate labels is an expensive task. For example, in the development of a spam filter one may have access to a large number of unlabeled emails [19]. It is a very expensive proposal to suggest that a human read millions of emails to provide labels, yet labels are necessary to develop an accurate filter. To address this problem, representative samples are selected and labeled. Given this subset of training instances which now have accurate labels, an automated process takes over and propagates the label logic to other instances. Active learning techniques may be combined with statistics-based generation in black-box or grey-box data knowledge scenarios where a large number of samples may be generated through random sampling of the features but class labels assigned through intervention by the adversary followed by active learning.

Query-Based Generation. When using query-based generation, an adversary will generate a random sample and then query the target model. The target model will then provide a probability vector output. In this case the adversary will want to identify instances for which the machine learning service provides a class label with relatively high confidence. That is, the adversary will search for instances in which the output $\mathbf{p}$ has a $max(\mathbf{p})$ value above some predefined threshold. This, again, may be combined with other techniques. For example, query-based generation may be used to provide a seed for active learning or statistics-based generation may inform the development of the instances sent to the service provider.

Region-Based Generation. Region-based generation follows a clustering-based logic. Given an instance $\mathbf{x}$ with label y, region-based generation will seek to generate instances $\mathbf{x}'$ where $dist(\mathbf{x}, \mathbf{x}')$ is below some pre-determined threshold for a pre-chosen distance function $dist$. The new instance $\mathbf{x}'$ is then assigned the same label y. One way an adversary may use region-based generation is in conjunction with white-box data knowledge. Given knowledge of some instances either in D or very similar to those in D an adversary can use region-based generation to expand this knowledge into a larger number of highly accurate instances to construct D'.

Several factors may determine which concrete technique will be chosen by an adversary, such as the knowledge contained in I or the query probing results. For example, an adversary who has grey-box data knowledge may be more likely to rely on statistics-based generation due to the specificity of the available statistics to D whereas an adversary with black-box knowledge may want to augment statistics-based generation with query-based generation due to lower confidence in their non-specific, population-level knowledge of D. On the other hand, active learning-based or region-based generation would likely be a popular choice for adversaries with white-box data knowledge where the adversary may leverage their insider knowledge of D. These are just some of the considerations in the development of an effective membership inference attack model.

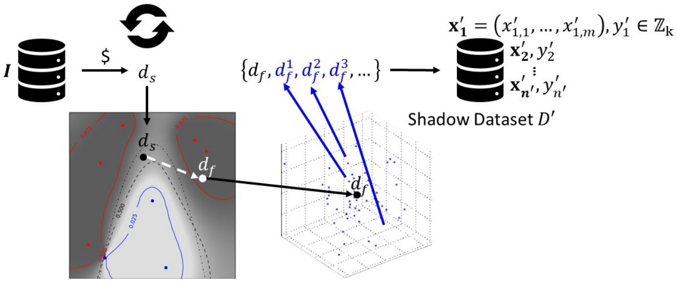
$\mathbf{x}'_1 = (x'_{1,1}, \ldots, x'_{1,m}), y'_1 \in \mathbb{Z}_k$
I
\$
d_s
$\{d_f, d^1_f, d^2_f, d^3_f, \ldots\}$
$\mathbf{x}'_2, y'_2$
$\mathbf{x}'_{n'}, y'_{n'}$
Shadow Dataset D'
d_s
d_f
d_f
0.975
0.975
0.500
0.025

Fig. 4. Shadow dataset development using query-based and region-based techniques with black-box data knowledge. Figures adapted from images in [20] and [21]

In Figure 4 we show an example of the shadow dataset development process using a combination of query-based and region-based techniques for an adversary with black-box data knowledge. The adversary first randomly generates a starting point d_s using distributions in I. This point d_s is then queried to the service provider which provides $\mathbf{p}_{d_s}$ in return. The point d_s is then updated randomly and queried again. This is continued until a confidence threshold or stopping condition is reached. That is, the process stops when either (1) a point d_f is found such that $max(\mathbf{p}_{d_f})$ meets a predefined confidence threshold or (2) the point has been updated and queried without meeting the confidence threshold the maximum number of times as determined by the attacker. If condition (2) is met a new d_s is chosen until the process results in condition (1) and an accepted point d_f.

A hyper-cube is then constructed surrounding d_f. A set of new samples $\{d^1_f, d^2_f, d^3_f, \ldots\}$ are then generated by randomly sampling from the hyper-cube region. Each point d^i_f is assigned the class $\arg\max_{i \in \mathbb{Z}_k} \mathbf{p}_{d_f}$ and added, along with d_f to the shadow dataset D'. The entire process is then repeated beginning with sampling a new starting point from I and ending with a new set of samples added to D'. The adversary will continue this repetition until a satisfactory number of samples n' have been added to D'.

3.2 Generation of an Attack Model Training Set

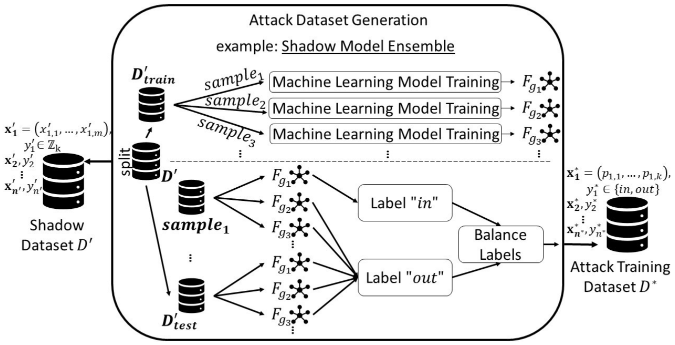
Attack Dataset Generation
example: Shadow Model Ensemble
D'_{train}
$sample_1$
$sample_2$
$sample_3$
Machine Learning Model Training
Machine Learning Model Training
Machine Learning Model Training
F_{g_1}
F_{g_2}
F_{g_3}
split
D'
$\mathbf{x}'_1 = (x'_{1,1}, ..., x'_{1,m})$, $y'_1 \in \mathbb{Z}_k$
$\mathbf{x}'_2, y'_2$
$\mathbf{x}'_{n'}, y'_{n'}$
Shadow Dataset D'
$sample_1$
D'_{test}
Label "in"
Label "out"
Balance Labels
$\mathbf{x}^*_1 = (p_{1,1}, ..., p_{1,k})$, $y^*_1 \in \{in, out\}$
$\mathbf{x}^*_2, y^*_2$
$\mathbf{x}^*_{n^*}, y^*_{n^*}$
Attack Training Dataset D^*

Fig. 5. Generation of an Attack Model Training Set.

Upon the completion of the development of the shadow dataset D', the adversary will proceed to utilize the shadow dataset D' to develop the membership attack dataset for training a binary classifier as the final attack model, as shown in Figure 5. Given that each instance in D' consists of a feature-vector and its known-class, denoted by $(\mathbf{x}', y')$, the adversary can define an attack generation function denoted by $F_g : (\mathbb{R}^m, \mathbb{Z}_k) \to (\mathbb{R}^k, \mathbb{Z}_2)$. F_g takes a feature vector-known class pair $(\mathbf{x}', y')$ as input and outputs an attack training instance, consisting of two pieces of information: a probability vector $\mathbf{p} = (p_1, p_2, ..., p_k)$ and a binary class label, indicating "in" or "out". There are several approaches to generate F_g using D'. For example, the adversary can train a new model over D' which simulates the private target model F_t. In this case, we call F_g a shadow model of F_t. Given that the adversary does not know the original training set D nor the size of D, the adversary may leverage ensemble learning techniques [22], such as data partition-based ensembles, model-based ensembles, or hybrid ensemble models, to improve the quality of the shadow model with the goal of accurately simulating the target model F_t. The attack model training set generation function $F_g : (\mathbb{R}^m, \mathbb{Z}_k) \to (\mathbb{R}^k, \mathbb{Z}_2)$ can therefore be viewed as an ensemble of a set of shadow models. These shadow models seek to characterize the *decision boundary* of the target model. More specifically, the shadow models aim at mirroring the sensitivity of the target decision boundary to individual instances.

Consider a data partition based ensemble approach [7]. The adversary partitions the shadow dataset D' into D'_{train} and D'_{test}. D'_{train} is then divided into q partitions ($q > 1$), one partition for each shadow model. Each partition of D'_{train} will then be used to train a single shadow model F_{g_i}. Here we intentionally do not specify the machine learning model type of F_{g_i} as this is yet another design choice made by an adversary. The decision may be informed if the adversary knows the model type of F_t or chosen using some other criteria, we leave this decision unspecified to remove the constraint that an adversary must know the model type of F_t. Next, D'_{test} will be evaluated against F_{g_i}. The corresponding outputs will then be labeled as "out". Additionally, a sample of size $|D'_{test}|$ is taken from the D'_{train} partition used to train F_{g_i} and evaluated against F_{g_i} with the corresponding outputs labeled as "in". By combining these output-label pairs, we obtain the attack train data D^*.

Figure 5 highlights the workflow of generating an attack model training set. An adversary may choose a single model for efficiency or an ensemble of models to increase the size or generality of D^*. An adversary may diversify model types in an ensemble when F_t's type is unknown. If an ensemble is used, there are choices on size, sampling, and aggregation that must be made and can be informed by the adversary's knowledge of the target in various ways. We stress that while we formulate the membership inference attack using a general model, there exist many implementation variants.

Effect of Ensemble Methods. Combining multiple different models reduces the risk of choosing the wrong hypothesis within the hypothesis space of a particular problem. Also, multiple models allows for more effective local search, which many machine learning algorithms perform in various ways, and limits the impact of the local optima problem. Finally, a combining of chosen hypotheses allows for an expansion of the hypothesis space [22]. Two common ways to accomplish this diversity are bagging and boosting.

Bagging is accomplished by either drawing a sample of training examples from the original dataset randomly and with replacement, or by creating disjoint subsets of the original training data called cross-validated committees [23]. A commonly used implementation of bagging is the Random Forest ensemble model [24].

Using the boosting technique, a set of weights is maintained for each instance within the training dataset. Each model is then trained iteratively to minimize the *weighted* error of the training dataset. The weights of the training instances are updated to put more emphasis on the misclassified examples. The adaboost ensemble model [25] is a popular implementation of boosting.

The use of multiple models via ensemble learning for attack data generation decreases the risk of choosing the wrong hypothesis. Given an adversary who has only black-box knowledge of the target model (regardless of the adversary's knowledge type of the dataset D), there is no guaranteed method to reproducing the target model's behavior on D. A diversity of generation models will minimize the risk that the adversary is only capturing one behavior type or candidate decision boundary shape. This again accentuates the need for boosting or bagging to ensure that the model set is diverse.

3.3 Generating the Membership Attack Model

The attack model training set D^* contains the outputs from the generation function $F_g : (\mathbb{R}^m, \mathbb{Z}_k) \to (\mathbb{R}^k, \mathbb{Z}_2)$. D^* consists of n^* instances, $(\mathbf{x_1^*}, y_1^*), (\mathbf{x_2^*}, y_2^*), ..., (\mathbf{x_{n^*}^*}, y_{n^*}^*)$, and each instance is the output of F_g for some input $(\mathbf{x}', y') \in (\mathbb{R}^m, \mathbb{Z}_k)$. This attack model training dataset D^* will then be used to generate the final attack model $F_a : \mathbb{R}^k \to \{\texttt{in},\texttt{out}\}$, which takes as input a probability vector output for an instance $\mathbf{x}$ and outputs a binary classification of "in" or"out". Similarly, we make no assumptions on how D^* is used to inform F_a but rather say that D^*

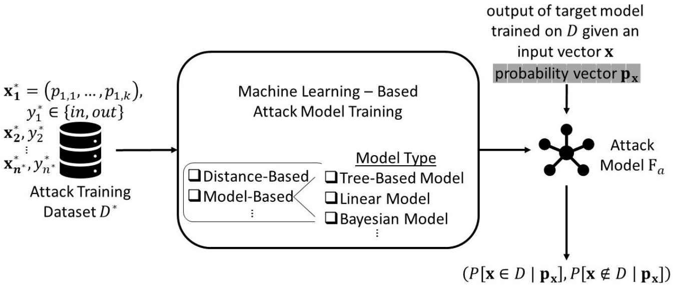
$\mathbf{x}_1^* = (p_{1,1}, \ldots, p_{1,k})$,
$y_1^* \in \{in, out\}$
$\mathbf{x}_2^*, y_2^*$
$\mathbf{x}_{n^*}^*, y_{n^*}^*$
Attack Training Dataset D^*
Machine Learning – Based Attack Model Training
❑Distance-Based
❑Model-Based
Model Type
❑Tree-Based Model
❑Linear Model
❑Bayesian Model
output of target model trained on D given an input vector $\mathbf{x}$
probability vector $\mathbf{p}_\mathbf{x}$
Attack Model F_a
$(P[\mathbf{x} \in D \mid \mathbf{p}_\mathbf{x}], P[\mathbf{x} \notin D \mid \mathbf{p}_\mathbf{x}])$

Fig. 6. Training and Deployment of the Membership Inference Attack Model

is available to an adversary during the generation of the membership attack model F_a. A number of machine learning models and techniques can be leveraged to train a binary classification-based attack model F_a using D^*. F_a can then be deployed against the output of the target model F_t such that, ideally, given an instance $\mathbf{x}$, $F_a(F_t(\mathbf{x})) =$ "in" if the instance $\mathbf{x} \in D$ and $F_a(F_t(\mathbf{x})) =$ "out" if $\mathbf{x} \notin D$. Figure 6 visualizes this final phase of developing a membership inference attack.

Regardless of how complex the chosen training process is, whether it is a distance evaluation or a complex machine learning model, this phase produces the final attack model F_a which will be deployed for membership inference attack against D in real time.

4 General Attack Characterization

We have thus far provided a general formulation of membership inference attacks. In this section, we characterize such attacks through a systematic evaluation of a variety of machine learning models and model combinations using multiple datasets. We show that membership inference vulnerability is data-driven and its attack models are largely transferable. Although the target model is a dominating factor in determining vulnerability, attack data generation techniques need not explicitly mirror the target model. Finally, we show that membership inference attacks can persist as insider attacks in federated systems.

4.1 Experimental Setup

We conduct a series of experiments to support our characterization of membership inference attacks. In this section we report empirical evidence for four types of machine learning models (logistic regression, k-nearest neighbor, decision tree, and Naïve Bayes). A total of seven datasets are used in these experiments: Adult, MNIST, CIFAR-10, Purchases-10, Purchases-20, Purchases-50, and Purchases-100. Our experiments were conducted using Python [26] and algorithms available with scikit-learn [20]. All reported results are averaged across 10 runs each using 10-fold cross-validation.

In reporting our results we use the accuracy metric. This is to capture the attacker's ability to both identify when $\mathbf{x} \in D$ and $\mathbf{x} \notin D$. Previous work, namely [7], report the precision metric which captures an attacker's ability to correctly identify when $\mathbf{x} \in D$ only. We therefore include this metric in Table 5 to compare the authors' neural network results with those from our experiments with logistic regression, k-nearest neighbor, decision tree, and Naïve Bayes models. Precision is also reported, along side accuracy, in Table 8 to aide in the comparison between insider and outsider attacks. All other results report the accuracy metric.

4.1.1 Datasets

4.1.1.1 Adult: The Adult dataset is available on the UCI Machine Learning Repository [27] and contains 48,842 instances described by 14 different features. The feature set contains both continuous (ex: age, hours per week) and discrete (ex: education, marital status) values. This dataset presents a binary classification problem wherein one wishes to identify if an individual makes $\geq$ \$50K or $<$ \$50K in yearly salary.

4.1.1.2 MNIST: MNIST is a publicly available dataset containing 70,000 images of handwritten digits [28]. Each image is formatted to be 32 x 32 and processed such that the digit is at the center of the image. The MNIST dataset constitutes a 10-class classification problem where the task is to identify which digit between 0 and 9, inclusive, is contained within a given image.

4.1.1.3 CIFAR-10: The CIFAR-10 dataset, also publicly available, contains 60,000 color images [29]. Again, each image is formatted to be 32 x 32. The CIFAR-10 dataset also has 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Each class has 6,000 available images. The problem is therefore a 10-class classification problem where the task is to identify which of the 10 classes is depicted in a given image.

4.1.1.4 Purchases: Finally, we developed a number of purchases datasets similar to what was done in [7]. The purchases datasets were developed from the Kaggle Acquire Valued Shoppers Challenge dataset which contains the shopping history of several thousand individuals. From this dataset we create new datasets wherein each instance represents an individual and each feature represents a particular product. If an individual has purchased this product, there will be a 1 for the feature and otherwise a 0. The instances are then clustered into different shopping profile types. These cluster assignments are treated as the classes. We created different datasets with 10, 20, 50, and 100 shopping profile types. The classification problem then becomes: given a shopper's purchase history, identify their shopping profile type.

We highlight these datasets as each shows fundamentally different results in previous work and the makeup of each is significantly different. For example, the Adult dataset presents a binary classification problem with numeric and factor features and was demonstrated to be resilient to membership inference attacks in previous work with precision results of 50.3%. By comparison, the Purchases-50 dataset has 50 different class values, only binary features, and was much more vulnerable to attack in previous work, with precision results of 86.0% [7].

4.2 Membership Inference: Data-Driven

Recall Section 3. There are many choices an adversary can make during the three phases of developing the membership inference attack model, each of which may impact the model accuracy and consequently the attack success rate. Two elements that the adversary cannot control, however, are the training dataset D and the target model F_t. Our experimental results show that membership inference attacks are data-driven. That is, the make-up of D strongly correlates with the corresponding target model F_t's vulnerability to membership inference attacks. Potential targets of such attacks can therefore use knowledge of their data to evaluate their risk. Table 1 compares the seven different datasets measuring three characteristics: (1) feature distribution, (2) number of classes (size of k), and (3) accuracy of (vulnerability to) membership inference.

Dataset	In-Class Standard Deviation	Number of Classes	Accuracy of Membership Inference
Adult	0.1433	2	59.89
MNIST	0.1586	10	61.75
CIFAR-10	0.2301	10	90.44
Purchases-10	0.3820	10	82.29
Purchases-20	0.3873	20	88.98
Purchases-50	0.3873	50	93.71
Purchases-100	0.3832	100	95.74

TABLE 1
Comparison of datasets versus membership inference attack accuracy using a decision tree model.

The number of classes is important as it characterizes the number of regions into which the input space $\mathbb{R}^m$ is divided. The more classes, the smaller each region. With smaller regions, there is less uninformed space and in fact the regions will more tightly surround the provided training instances in D. This will make any single instance more likely to alter the decision boundary as space is "tighter" between the regions. If an instance is more likely to impact the decision boundary of the target model, then an adversary will be more likely to infer its inclusion in the training dataset.

Another side of this same argument can be seen in the in-class standard deviation metric. This value captures feature distributions by addressing the following: within a dataset, given all instances of the same class, how similar are the feature vectors? The Adult and MNIST datasets, for example, have significantly lower standard deviations than the Purchases datasets despite having more instances of each class. This demonstrates more uniformity within classes of the Adult and MNIST datasets as compared to the Purchases datasets. If an instance is exceedingly similar to other instances of the same class then it will be less likely to noticeably impact the decision boundary during training. If, however, instances within a class are notably different, then the inclusion of each instance may significantly impact the decision boundary. Therefore, the uniformity of the target training data within each class will, in addition to the number of classes, impact an adversary's ability to identify the inclusion of a particular instance.

The variation in attack accuracy results reported in Table 1 from experiments using the same attack development process targeting decision tree models trained on each of the seven datasets demonstrates that factors in the dataset such as the in-class standard deviation and the number of classes contribute to a model's susceptibility to membership inference. We make two interesting observations. First, the impact of in-class standard deviation varies for different types of datasets and different scales of k-class classification problems. Second, membership inference vulnerability is related to a number of factors, many of which are tied to the dataset itself. Concretely, for image datasets the in-class standard deviation increases as the complexity of the dataset increases. For example, MNIST represents images of grey scale for 10 single digits and CIFAR-10 represents color images of 10 different complex entities ranging from flowers to animals. It is observed that the attacks have higher success rate for more complex images which also show larger in-class SD. On the other hand, consider the Purchase datasets where the value of k varies from 2 to 10, 20, 50 and, finally, 100. The attack accuracy increases as k increases without the standard deviation increasing. This shows, from a different perspective, that attack accuracy is higher for more complex classification problems. For example, consider the Purchases-50 and Purchases-100 datasets. The increase in attack accuracy demonstrates that in-class standard deviation may not always be a dominant factor in membership inference vulnerability, other data-driven factors, such as the complexity of the classification problem and the complexity of the dataset itself, can also be important factors in membership inference vulnerability.

Table 1 shows more explicitly the impact on the attack accuracy of two parameters: the variation of in-class standard deviation and the parameter k for the k-class classification problem. While we want to highlight the impact of these important parameters, neither the in-class standard deviation nor the value of k is the sole dominating factor. Rather, the results in Table 1 demonstrate that the membership inference vulnerability is not isolated to the learning process itself but also highly related to several attributes of the training dataset. This leads us to characterize membership inference attacks as data-driven.

4.3 Transferable Attack Models

Recent studies on adversarial machine learning attacks such as evasion attacks or poisoning attacks [30], [31], [32], [33] have shown that maliciously generated adversarial examples tend to transfer from one model to another. This is an important property as it opens the door to adversaries in the black-box scenario. That is, an adversary need not know details of their target model to launch a successful attack. Through extensive experiments, we observe that membership inference attacks are similarly transferable.

For example, the results in Table 2 are the accuracy of membership inference attacks using various attack configurations against a decision tree model trained on the Purchases-20 dataset. The relative consistency seen in Table 2 demonstrates that many attack configurations are viable. An adversary does not need to know the target model configuration to launch an effective attack. This suggests that membership inference attack models are transferable from one target model to another, provided that targets are trained on the same dataset D.

Next, we evaluate the standard deviation of membership inference attack results for a membership inference attack model F_a learned over the output of an attack data generation model F_g and deployed against a target model F_t across the seven datasets for the following three scenarios: (1) vary the model types for F_g and F_a while keeping F_t consistent, (2) varying F_t and F_a while keeping F_g consistent, and (3) varying F_t and F_g while keeping F_a consistent.

Purchases-20	Attack Data Generation Model			
Attack Model	DT	k-NN	LR	NB
DT	**88.98**	**87.49**	72.08	81.84
k-NN	**88.23**	72.57	**84.75**	74.27
LR	**89.02**	**88.11**	**88.99**	83.57
NB	**88.96**	78.60	**89.05**	66.34

TABLE 2
Accuracy of membership inference attack against a decision tree target model trained on the Purchases-20 dataset.

Table 3 shows the results of the experiments on various combinations of model types for the CIFAR-10 dataset. We calculate the average standard deviation for scenario (1) as follows: DT_{sd} denotes the standard deviation of all accuracy values in rows 1, 5, 9, and 13 where the target model is a decision tree (DT). Similarly, $k\text{-}NN_{sd}$ corresponds to rows 2, 6, 10, and 14, LR_{sd} with 3, 7, 11, and 15, and NB_{sd} with 4, 8, 12, and 16. Then we consider the average standard deviation in accuracy for scenario (1) to be the average of DT_{sd}, $k\text{-}NN_{sd}$, LR_{sd}, NB_{st}. For scenario (3) we follow a similar process but with row sets 1-4, 5-8, 9-12, and 13-16. Scenario (2) is calculated by averaging the standard deviations for each of the 4 F_g columns.

Attack Model	Target Model	Attack Data Generation Model			
		DT	k-NN	LR	NB
DT	DT	**90.44**%	85.64%	60.48%	65.78%
	k-NN	54.92%	69.32%	55.01%	51.38%
	LR	53.84%	61.06%	61.10%	50.02%
	NB	50.46%	50.58%	49.98%	50.20%
k-NN	DT	**89.96**%	81.55%	89.07%	61.10%
	k-NN	55.33%	68.32%	62.45%	50.89%
	LR	51.34%	59.58%	64.78%	50.09%
	NB	50.12%	50.61%	50.46%	50.11%
LR	DT	**90.37**%	**90.11**%	88.81%	**66.98**%
	k-NN	51.72%	69.90%	65.29%	55.64%
	LR	50.01%	64.34%	67.40%	54.49%
	NB	50.54%	50.63%	50.60%	50.29%
NB	DT	**90.42**%	89.86%	**90.52**%	63.71%
	k-NN	50.33%	68.31%	57.65%	53.08%
	LR	50.00%	64.22%	67.63%	53.54%
	NB	50.58%	50.44%	50.58%	50.01%

TABLE 3
Accuracy for CIFAR-10 dataset across experiments with various attack, data generation, and target models.

We follow this process for each of the seven datasets and summarize the results in Table 4. This allows us to compare the impact of model variation for each F_t, F_g, and F_a. We observe that the standard deviation of membership inference attack results is relatively small against a fixed target model when compared to a fixed attack data generation model or fixed attack model. A smaller standard deviation is indicative of a larger impact. That is, the accuracy is stable when the standard deviation is small. A small standard deviation would indicate that the fixed model has more influence over attack accuracy than the varied models.

Table 4 clearly shows that for all datasets the deviation is minimized when F_t is fixed. This indicates that variation in F_g and F_a have a comparatively low impact on attack success rates. This supports the evidence in Table 2 that an adversary need not have particularly informed choices in F_g or F_a to develop an attack model. We therefore characterize membership inference attacks, like others in the adversarial learning domain, as transferable.

Dataset	Standard Deviation in Accuracy Results		
	Fixed F_t	Fixed F_g	Fixed F_a
Adult	**0.0093**	0.0335	0.0328
MNIST	**0.0126**	0.0347	0.0351
CIFAR-10	**0.0643**	0.1233	0.1366
Purchases-10	**0.0396**	0.1069	0.1074
Purchases-20	**0.0545**	0.1336	0.1352
Purchases-50	**0.0705**	0.1468	0.1482
Purchases-100	**0.0849**	0.1468	0.1452

TABLE 4
Standard deviation between accuracy results with (1) fixed F_t type and varying F_g and F_a types, (2) fixed F_g type and varying F_t and F_a types, and (3) fixed F_a type and varying F_t and F_g types.

In summary, Table 2 highlights the role of the attack data generation model using the Purchases-20 dataset. It articulates that a variety of attack model structures may be used against a single target model. Tables 3 and 4 cover combinations of target models, attack models, and attack data generation models with the CIFAR-10 dataset to illustrate a broader, more comprehensive picture that both the target model and the target dataset are the most important factors for membership inference vulnerability. For each of the four attack models (DT, k-NN, LR, NB), the results show the impact in varying target models on attack accuracy under one of the four given attack data generation models. Table 4 shows the derivation of accuracy results for *each* of the seven datasets under a fixed target, a fixed generation model, or a fixed attack model. These results indicate that an adversary may be able to develop an attack model without knowing “best attack model or the “best attack data generation model. To further understand this observation, we next study the impact of different choices of attack model on attack accuracy.

4.4 Attacks Across Model Types

Many works in adversarial machine learning focus on the vulnerability of deep learning models whose uses range from image classification [34], [35] to speech recognition [36], [37] to natural language processing [38]. By generating adversarial examples which are tweaked in ways that are unnoticeable to humans, adversaries can exploit the models' complexities to force misclassification [39], [40]. This trend has also influenced the study of membership inference problems.

Most existing efforts on membership inference [8], [9], [10], [7] have been focused on deep learning models. We argue that the complexity exploited in traditional adversarial learning attacks is not explicitly leveraged in membership inference attacks. Additionally, areas where membership inference would be most alarming, such as healthcare [41], e-commerce, banking, and government often deploy simpler model types, such as decision trees as model understanding is prioritized for safe use of predictive services. This leads to a natural question: are model types outside of deep learning methods susceptible to membership inference? Our empirical study on all seven datasets and four types of models demonstrates that not only are other model types vulnerable to membership inference attacks but that the model type is in fact very influential in determining the extent of that vulnerability.

The basic hypothesis of membership inference attack is it that models respond differently to instances which they have "seen" versus those they have not. In this hypothesis it is clear that model behavior and sensitivity are likely to impact vulnerability. We therefore consider a variety of model types outside of neural networks. We consider models from 4 different major categories: linear models (logistic regression), Bayesian models (Naïve Bayes), cluster models (k-nearest neighbor with k=5), and tree models (CART decision trees).

For equal comparison to the previous work in [7], we use the shadow model implementation of the membership inference attack. All results from our experiments use 10-fold cross-validation and are averaged across 10 runs for a randomly selected sample of 10,000 instances.

Dataset	LR	k-NN	DT	NB	NN
Adult	*50.13*	51.39	**55.49**	50.22	50.30
MNIST	53.25	*50.44*	**56.66**	50.48	51.70
CIFAR-10	70.25	65.99	**83.94**	*50.03*	78.00
Purchases-10	64.56	53.53	**73.85**	*50.61*	55.00
Purchases-20	75.85	55.36	**81.94**	*50.79*	59.00
Purchases-50	81.61	58.19	**88.88**	*52.08*	86.00
Purchases-100	83.78	60.11	92.19	*54.93*	**93.50**

TABLE 5
Precision of membership inference attack across 5 model types.

Dataset	LR	k-NN	DT	NB
Adult	*50.17*	51.22	**59.89**	50.18
MNIST	54.38	50.59	**61.75**	*50.81*
CIFAR-10	67.40	68.32	**90.37**	*50.01*
Purchases-10	66.82	53.78	**82.29**	*51.00*
Purchases-20	80.50	55.92	**88.98**	*51.29*
Purchases-50	88.60	59.57	**93.71**	*53.49*
Purchases-100	90.23	62.19	**95.74**	*57.61*

TABLE 6
Accuracy of membership inference attack across 4 model types.

We can clearly see in Table 5 that other models are in fact vulnerable to membership inference and that both the training data (as discussed in Section 4.2) and the model type play an important role in understanding a particular model's risk. As shown in Table 5, despite the variety in our datasets, the highest precision is seen with the decision tree model for all datasets except Purchases-100 while the Naïve Bayes models consistently show exceedingly low precision across all datasets.

In general, a target model whose decision boundary is unlikely to be drastically impacted by a particular instance will be more resilient to membership inference attacks. For example, the Naïve Bayes algorithm independently considers the probability of a given class for each feature. Therefore, given significant training samples, a single instance only marginally affects these probabilities. This explains the low numbers continuously seen when attacking a Naïve Bayes model. By contrast, a decision tree leaf node will consider a unique feature combination to determine class rather than each feature in isolation. The introduction of a single instance, if that instance displays a unique feature set-class combination, may cause a decision tree to grow and entire new branch. This sensitivity to single instances makes membership inference attacks more successful when targeting decision tree models.

Consequently, it is important to understand that while different datasets display different vulnerabilities to membership inference, so do different model types. This also indicates that machine learning-as-a-service providers who are wary of membership inference attacks against their deployed models may also be able to use model choice to help mitigate vulnerability.

We have studied the impact of various target models on attack accuracy (Section 3.3) and the impact of various attack models on attack accuracy (Section 4.4), we next study the impact of the attack generation models on the accuracy of the membership inference attack.

4.5 Variation in Generation Model

Instinctually, one may believe that the model used to generate attack data must be of the same type as the target model. We briefly discuss why previous research has made this same assumption and then investigate its veracity and seek to explain why it does not strictly hold.

In [7], the authors claim that the shadow model implementation of a membership inference attack requires the shadow models be trained in a similar way to the target model, an assumption followed by later work such as [42]. Consistent with our formalization of membership inference attacks, this claim is equivalent to saying that the attack data generation technique F_g must mirror the behavior of the target model F_t under attack.

The reasoning behind this assumption is intuitive. Let us say the target model F_t is an approximation of an ideal function Γ for a dataset D. Then, given a data point $\mathbf{x}$ for which the adversary aims to identify membership in the training data D, the output provided to the adversary will be $F_t(\mathbf{x})$. This output will then be provided to the attack model to determine classification of $\mathbf{x}$ as "`in`" or "`out`". Let the binary classifier which serves as the attack model be trained from the output of a generation model F_g. That is, the output $F_g(\mathbf{x}')$ for all $\mathbf{x}' \in D'$ make up the attack model training data D^*.

We can clearly see that our attack model is therefore trained on the output of a function F_g and deployed against the output of a function F_t. It is understandable then that previous work would seek to have the behavior of F_g mirror the behavior of F_t. This assumption naturally extends to say that F_g, in an attempt to mirror F_t, must be, or intuitively should be, of the same model type as F_t in a successful membership inference attack model.

However, this assumption is not necessary to launch an effective membership inference attack. We conducted a set of experiments varying model type combinations of F_t, F_g, and F_a considering the four candidate model types.The combination with the highest accuracy for each dataset is reported in Table 7. Let $type(F_t^{max}), type(F_g^{max}), type(F_a^{max})$ be the target model, attack data generation model, and attack model types respectively for the membership inference attack which reported the highest accuracy. These types will naturally vary for different datasets. We also report

Dataset	Model Types for $(F_t^{max}, F_g^{max}, F_a^{max})$	Accuracy $(F_t^{max}, F_g^{max}, F_a^{max})$	$type(F_t^{max})$	Accuracy All $type(F_t^{max})$	$type(F_g^{max})$	Accuracy All $type(F_g^{max})$	$type(F_a^{max})$	Accuracy All $type(F_a^{max})$
Adult	(DT, DT, NB)	59.91%	DT	59.89%	DT	59.89%	NB	50.18%
MNIST	(DT, DT, LR)	61.80%	DT	61.75%	DT	61.75%	LR	54.38%
CIFAR-10	(**DT**, **LR**, NB)	90.52%	**DT**	90.44%	**LR**	67.40%	NB	50.01%
Purchases-10	(**DT**, **k-NN**, DT)	82.45%	**DT**	82.29%	**k-NN**	53.78%	DT	82.29%
Purchases-20	(**DT**, **LR**, NB)	89.05%	**DT**	88.98%	**LR**	80.50%	NB	51.29%
Purchases-50	(**DT**, **LR**, LR)	93.77%	**DT**	93.71%	**LR**	88.60%	LR	88.60%
Purchases-100	(**k-NN**, **LR**, DT)	95.86%	**k-NN**	95.74%	**LR**	90.23%	DT	62.19%

TABLE 7

Model set up with maximum accuracy averaged across 10 runs using 10-fold cross validation. Maximum configuration is then compared to configurations where model type is consistent across the target, generation, and attack models using each model type represented in the maximum configuration.

accuracy for scenarios when all three model types are set to $type(F_t^{max})$. We similarly report when all three types are equivalent to $type(F_g^{max})$ and $type(F_a^{max})$.

For example, we recall that in Table 3 the highest accuracy was reported when the target model was a decision tree, the data generation model was a logistic regression model, and the attack model was a Naïve Bayes model. Therefore, $type(F_t^{max})$ = DT, $type(F_g^{max})$ = LR, and $type(F_a^{max})$ = NB, as shown in column 2 of Table 7 along with the attack accuracy under these settings. For the CIFAR-10 dataset, we then also report the attack accuracy when all the models are a decision tree (i.e. equivalent to $type(F_t^{max})$), when all models are logistic regression models ($type(F_g^{max})$), and finally when all models are Bayesian ($type(F_a^{max})$). We repeat this process for all seven datasets.

We observe that, across datasets, it is not necessary for all models to be of the same type, as no maximally accurate combination contained the same model type across all three phases of the membership attack development. Additionally, the attack data generation model, as previously assumed, does not need to strictly mirror the target model for a successful membership inference attack. In fact, for 5 out of the 7 datasets the highest accuracy was reported when the attack data generation model was of a different type than the target model, i.e. $type(F_t^{max}) \neq type(F_g^{max})$. The reason behind this non-intuitive phenomenon lies in a more precise understanding of the role of the generation model. Although it was previously assumed that the role was to mirror the behavior of the target model, we assert that the generation model's role is to characterize how the target model may be impacted by the inclusion of a particular instance. That is, how the decision boundary of the target model may reveal the inclusion of an instance.

The generation algorithm is therefore trying to characterize probability distributions related to a decision boundary which either has or has not been informed by the instance. From this perspective, it is now more clear that the vulnerability is more closely related to two elements: distribution of the data and sensitivity of the decision boundary. If a decision boundary for a given dataset is notably impacted by the inclusion of a given instance then membership inference attacks are likely to be more successful.

In summary, we have demonstrated that, despite natural intuition, it is not strictly necessary for the attack data generation technique to be of the same type as the target model for a membership inference attack to be successful. Additionally, we again see the dominating factors are the target model type and the training dataset. We note that when all model types are equivalent to F_t^{max} the resulting accuracy is within 0.16% of the maximum reported accuracy. By comparison, setting all model types to F_g^{max} decreases attack accuracy by up to 28.67%. Setting all model types to F_a^{max} demonstrates an even larger decrease in many cases as F_a^{max} is reported to be a Naïve Bayes model for multiple datasets. As we previously identified the Naïve Bayes model as robust against membership inference attack, it is not surprising that attack accuracy will decrease significantly in these cases.

While success is close to maximal in all scenarios when the target model type is known, we re-accentuate the attack success seen in Table 7 with a mixture of model types. This again supports the conclusion that an adversary need not have this level of insider knowledge to launch a successful attack. Rather, when a model and its training dataset are particularly vulnerable, a variety of attack scenarios are likely to demonstrate success.

4.6 Impact of Attacker Knowledge

Finally, we demonstrate the impact of attacker knowledge on their ability to launch successful membership inference attacks. We evaluate attacker knowledge along two separate vectors: (1) accurate target data and (2) accurate shadow data.

In Figure 7, we demonstrate the impact of attacker knowledge of the target data points. That is, if the attacker does not accurately know each feature within $\mathbf{x}$, is the attacker still able to identify if $\mathbf{x} \in D$ or if $\mathbf{x} \notin D$? To simulate this uncertainty we add noise to the target data points prior to querying. Given features normalized within $[0, 1]$ we add noise uniformly sampled from $[0, \sigma]$. In Figure 7, we gradually increase σ from 0 to 1, recording the impact on the membership inference attack accuracy. We report results from 4 datasets using logistic regression models for target, generation, and attack models.

While it is unsurprising that the noisier the target instance the less accurate the attack, it is also interesting to note that accurate attacks may still be launched when small amounts of noise are added. That is, if the attacker knows 90% of the data for the target instance but perhaps has to make an educated guess for the final 10%, then that attacker is able to identify whether or not the target instance was within the training data with similar accuracy as was seen when 100% of the target data was known.

We conducted similar experimentation with respect to the shadow data. That is, we began by selecting a shadow dataset disjoint from the target training data but from the same distribution. We then added noise to each instance

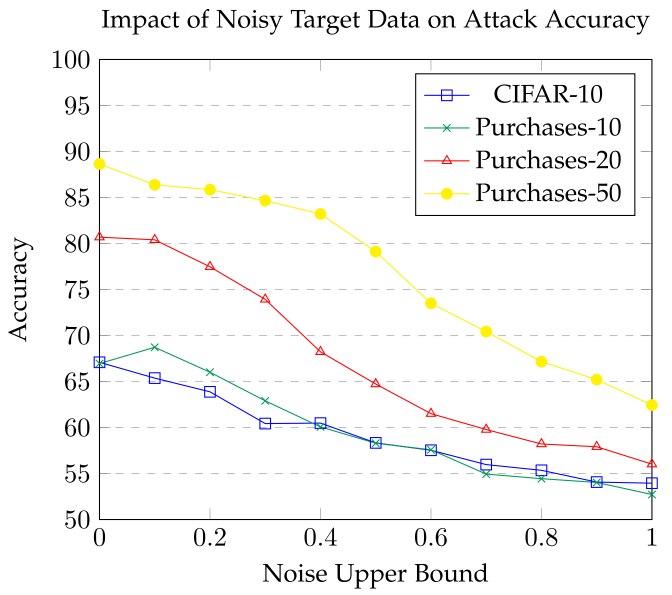

Fig. 7. Impact of Noisy Target Data on Attack Accuracy with Logistic Regression models.

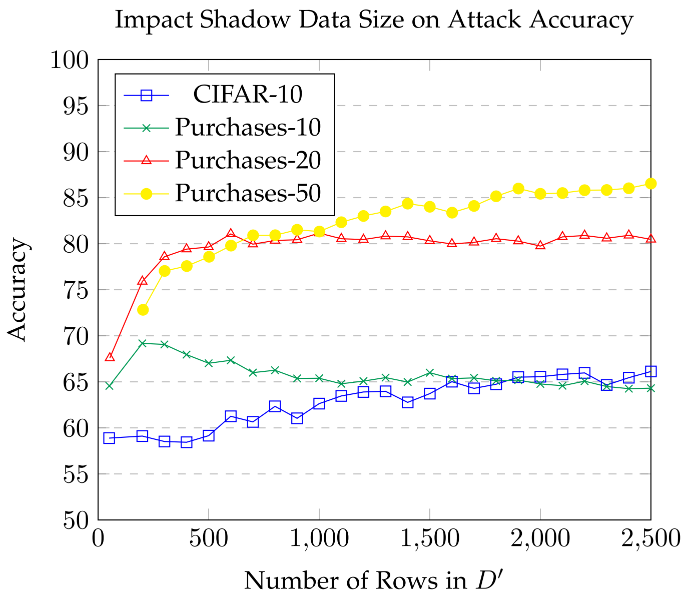

Fig. 9. Impact Shadow Data Size on Attack Accuracy with Logistic Regression models.

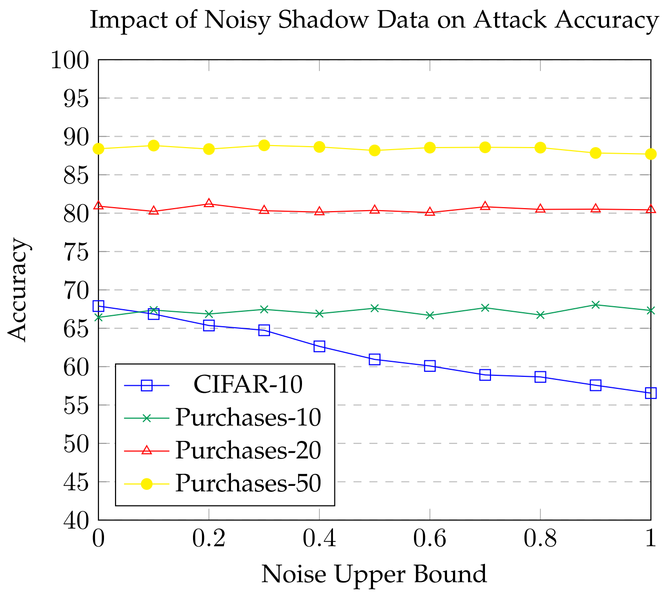

Fig. 8. Impact of Noisy Shadow Data on Attack Accuracy with Linear Regression Models.

within the shadow dataset as was done in the previous experiment. Shadow models were then trained and evaluated during the attack data generation process using this noisy shadow dataset. Figure 8 reports the impact of σ on the resulting membership inference attack accuracy. Experiments were again run for 4 datasets with the logistic regression model used in all phases.

We can see in Figure 8 that the attack accuracy is relatively resilient to noisy shadow data. While the CIFAR-10 dataset shows a slight decline in accuracy as noise increases, this decline is minor when compared to the inclusion of noise into the target data. Additionally, the various Purchases datasets show strong resilience to the introduction of noise into the shadow dataset. When comparing Figures 7 and 8 it is clear that the attacker is more likely to be successful if resources are allocated to developing strong, accurate target instances compared to perfectly representative shadow data.

Finally, in Figure 9 we report the effect that the size of the shadow dataset has on attack accuracy. For these experiments we select shadow datasets of various sizes which are again disjoint from the target training data but from the same distribution. No noise is added for these experiments to isolate the impact of the shadow data size. The experiments are evaluated on the same four datasets again with logistic regression models.

Each dataset in these experiments appears to result in different behavior. The CIFAR-10 dataset continues to show increased accuracy as the shadow dataset size increases. The Purchases-10 dataset however shows an initial accuracy increase before decreasing again and leveling out by the time the shadow dataset reaches 1,000 instances. The Purchases-20 dataset shows a fast initial trend of accuracy improvement before leveling off at a shadow dataset size around 700 instances. Finally, the Purchases-50 dataset results show a similar trend to that seen with the CIFAR-10 dataset where the accuracy continues to increase with the shadow dataset size.

Overall through Figures 7, 8, and 9 it is clear that accuracy in the target instances is the most important attacker knowledge factor in determining attack success. With limited resources an attacker should therefore focus on acquiring sufficient information with respect to target instances. The next most deterministic element is the size of the shadow dataset. Rather than allocating resources to ensuring completely accurate instances, attackers should instead put the focus on quantity in the development of their shadow datasets.

4.7 Federated Learning and Insider Attacks

4.7.1 Insider Attack Model

To this point we have exclusively discussed "outsider" membership inference attacks. That is, membership inference attacks which are launched by an adversary who is only a user of the target model through black-box access to the target service prediction API. We now introduce the threat of insider membership inference attacks. We define these attacks to be those launched by a participant in a federated learning system.

In recent years there has been an increased interest in the role of federated learning systems in addressing privacy concerns in data mining. The intuition behind federated systems to protect against membership inference is as follows: if an adversary is able to identify that a certain instance is contained within the training data of a model and that model is the result of a federated learning system, then any individual participant will have plausible deniability with respect to their individual dataset. However, such federations open the door to a new risk through insider attacks.

The difference between an attack on federated systems and outsider membership inference attacks is that, in federated systems, the training dataset D is divided amongst multiple parties who engage in collaborative learning to provide predictions to the machine learning service. We consider the following loosely federated system: given parties $P_1, P_2, ..., P_\ell$ there exist ℓ independent dataset $D_1, D_2, ..., D_\ell$, one belonging to each party. Each party P_i will then train a model F_{t_i} using D_i as the corresponding training data.

Within this environment, new instances will be evaluated as follows. On input of $\mathbf{x}$, each model F_{t_i} will output a probability vector $\mathbf{p_i} = F_{t_i}(\mathbf{x})$. The individual parties will then share their output either openly with one another or with some aggregation service to compute the final output $\mathbf{p} = \overline{ave}(\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p}_\ell)$ where $\overline{ave}$ refers to point-wise averaging of the probability vectors. Any outside adversary using this service will only have access to the final probability vector $\mathbf{p}$. Any individual party P_i will therefore have plausible deniability because, if an instance is identified as a member of the training data, the adversary is unlikely to identify which training set specifically.

However we must also consider adversaries who are members of the federated learning systems. That is, the aggregation service or a participating party. Under this scenario the "insider" will have access to the individual probability vectors $\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p}_\ell$. The insider membership inference attack then becomes: given these probability vectors, is the adversary able to identify which dataset a training instance belongs to.

We now consider *when* parties may participate in a federated system. Let P_1, P_2, and P_3 represent three candidate parties with training datasets D_1, D_2, and D_3 respectively. Let us assume the extreme case that D_1, D_2, and D_3 are statistically equivalent. Then, the trained models F_{t_1}, F_{t_2}, and F_{t_3} will approximate the same ideal function Γ. Given this set up, it is then likely that, on any input $\mathbf{x}$, the outputs $F_{t_1}(\mathbf{x})$, $F_{t_2}(\mathbf{x})$, and $F_{t_3}(\mathbf{x})$ will also be statistically equivalent. Here, there is no accuracy gain for P_1, P_2, or P_3 through collaboration. They are therefore unlikely to be motivated to create a federated learning system.

Alternatively, consider such significantly different D_1, D_2, and D_3 such that F_{t_1}, F_{t_2}, and F_{t_3} may be considered independent. Let us now assume accuracies of 75%, 80%, and 70% and a majority voting aggregation scheme. Such a federation, on input $\mathbf{x}$, has an 84.5% chance of accurately classifying $\mathbf{x}$, an accuracy higher than any individual model. Under these conditions, P_1, P_2, and P_3 are much more likely to form a federation.

It is therefore reasonable to assume parties are likely to form federated learning systems when their individual datasets are sufficiently different. Unfortunately, this leads to sufficiently different decision boundaries for different parties. These diverging decision boundaries open the door to effective insider membership inference attacks as an adversary will notice differences in $\mathbf{p_1}$, $\mathbf{p_2}$, ..., $\mathbf{p}_\ell$.

4.7.2 Insider Attack Risk in Federated Systems

In Table 8 we see that even datasets showing resilience to outsider membership inference are vulnerable to an insider membership inference attack. We created a federated system where $n = 3$ for datasets with $k \leq 10$ so that each party has sufficient instances of each class to learn a meaningful decision boundary. Given a federation where $n = 3$, any party behaving as an adversary will have an attack precision baseline of 50%. This allows for comparison with the outsider inference attacks. Both the Adult and MNIST dataset showed minimal vulnerability in the outsider membership inference attacks and experienced significant jumps in vulnerability in the insider attack scenario while the CIFAR-10 and Purchases-10 datasets show similar precision results, notably outperforming the baseline.

Dataset	Outsider Inference Precision (Accuracy)	Insider Inference Precision (Accuracy)
Adult	55.49 (59.89)	**73.26** (69.33)
MNIST	56.66 (61.75)	**68.47** (68.17)
CIFAR-10	83.94 (90.44)	82.02 (82.05)
Purchases-10	73.85 (82.29)	**74.42** (74.30)

TABLE 8
Insider inference precision in federated systems with 3 parties. Baseline is 50%. Model type is set to decision tree.

In Figure 10, we plot the decision boundaries created by three different decision tree models trained on disjoint subsets of the Adult dataset to articulate how the variation in decision boundaries informs insider attacks. We plot the decision boundaries relative to the *capital loss* and *education number* features. The section enclosed within the blue box highlights a portion of the decision boundary which notably differs between each plot. The second level of Figure 10 is a zoomed-in view of this section for all three plots. On the third level we then plot the positive training instances that informed each decision boundary in this region. It is clear that the long region identified as the positive class in the third plot was informed by significantly more positive instances than the other two decision plots. It is decision boundary differences such as those demonstrated here, and what they reveal of the underlying training data, that reveals ownership in the insider membership inference attack.

More specifically, we know that a participant who returns prediction vectors consistent with the third plot (i.e. higher probabilities for the positive class within the long horizontal region), is more likely to have seen a participant with a capital loss value just under 2,000 and a salary above $50,000. The probabilities returned from the participant with a model whose decision boundary is more consistent with the first plot however will return higher probabilities for the negative class. It is therefore much less likely that this participant has an individual with a capital loss value within this range and a salary greater than $50,000. An attacker, therefore, will know that such an individual (i.e. one who falls within this range with a higher salary) is much more likely to be in the dataset of the third participant.

This is supported by the accuracy seen for federations with different data distributions characteristics. When constructing our federated learning systems for these experiments we first sampled target class distributions at random to create different scenarios which may be seen in deployed federated learning environments.

In Figure 11 we show the relationship between the accuracy of the insider membership inference attack and the distance between the two targeted parties' data. That is, let a 3-party federation be formed by parties P_1, P_2, and P_3 wherein P_3 behaves maliciously and launches an insider membership inference attack against P_1 and P_2. Then, we look at how similar all the instances of class k_i in D_1 are to those of class k_i in D_2. This similarity is averaged across all $k_i \in [1, k]$. If the datasets are more similar then their in-class distance will be lower. In Figure 11 we can see that P_3 will be less successful than when D_1 and D_2 have a closer in-class measure than if the datasets are very different.

Unfortunately this leads to a catch-22 scenario. Parties with very similar looking training datasets will not be motivated to participate in a federation as they are less likely to see significant increases in classification accuracy. Parties with very different looking training datasets, however, will be more vulnerable to insider membership inference.

We argue that the risk of *insider* membership inference attack is of particular concern as participants are likely to assume they are less vulnerable than in outsider scenarios due to the plausible deniability protections inherent in federated learning systems. The potential for insider attacks, however, calls for a robust trust policy for any federated learning system to consider such risk.

5 Mitigation Techniques

We categorize mitigation techniques to protect against membership inference into two categories: model hardening and API hardening.

Model Hardening. Model hardening mitigation techniques are implemented during the training phase of the target model F_t. We suggest four such techniques: (1) model choice, (2) fit control, (3) regularization, and (4) anonymization. In model choice, a service provider may introduce concerns of membership inference into their model selection process. For example, as was demonstrated in our experimentation, a Naïve Bayes target model will be much more resilient to membership inference attacks than a decision tree and therefore may be the preferred model type for a particular machine learning service. For fit control, service providers may leverage parameters such as the decision tree's complexity parameter for tree pruning to prevent overfitting and therefore decrease inference risk. Another technique is regularization where noise is added to a model's loss function. This technique is particularly relevant to deep learning models. Finally, the service provider may introduce anonymization techniques into D prior to training. That is, if D is made to be k-anonymous prior to the training of F_t then the impact of a single instance may be hid amongst k others prior to training.

API Hardening. API hardening techniques are implemented during the prediction phase through the machine learning service. For example, the service API may introduce noise into the prediction vector before returning it to the user. This will reduce the adversary's understanding of exactly where an instance lies with respect to the target model's decision boundary. Another option is to reduce the dimensionality of $\mathbf{p}$. This can be done either by limiting the return value to the top $k' < k$ values in $\mathbf{p}$ or even returning only the prediction label. Machine learning-as-a-service APIs may additionally be hardened through limiting queries and/or allocating a differential privacy budget to query responses [43]. This however requires service providers to track and control user queries which may not always be a suitable option.

Mitigation	Parameter	Model Accuracy	Attack Accuracy	Utility Δ
None		55%	83%	
Dimension Reduction [7]	$k' = 3$	55%	83%	▼
	$k' = 1$	55%	82%	▼
	k' =label	**55%**	**73%**	▼
Regularization [7]	L2 $\lambda = 1e-4$	56%	80%	▲ 1%
	L2 $\lambda = 5e-4$	57%	73%	▲ 2%
	L2 $\lambda = 1e-3$	56%	66%	▲ 2%
	L2 $\lambda = 5e-3$	**35%**	**52%**	▼ 20%
Adversarial Regularization [44]	0	51.9%	63%	▼ 3.1%
	2	**47.5%**	**51%**	▼ 7.5%

TABLE 9
Results of mitigation techniques using dataset of Texas hospital admissions which contains 100 classes. A neural network target model is used.

An issue that is pervasive in each mitigation technique is a loss of utility. This is demonstrated in Table 9. We note here that there is not significant reduction in attack accuracy until only the label is returned when using a dimension reduction technique. Consider a hospital processing images of mass scans for cancer classification. A service which says "This mass is cancerous." is significantly less useful than a service which says "There is a 56% chance that this mass is cancerous." Additionally, even with the strongest dimension reduction, the attack accuracy is still notably outperforming the baseline at 73%. Regularization on the other hand is able to successfully decrease attack accuracy to 52%. Unfortunately, to gain this level of protection the noise introduced to the model decreases model accuracy to 35%. This is a significant challenge in the mitigation of membership inference attacks.

Recent work in [44] has proposed a regularization technique specifically targeting membership inference attacks. While this approach is able to decrease attack accuracy to 51% with a comparatively moderate decrease in test accuracy to 47.5%, the training accuracy drops from 81.6% to 55%. It is therefore unclear to what degree regularization

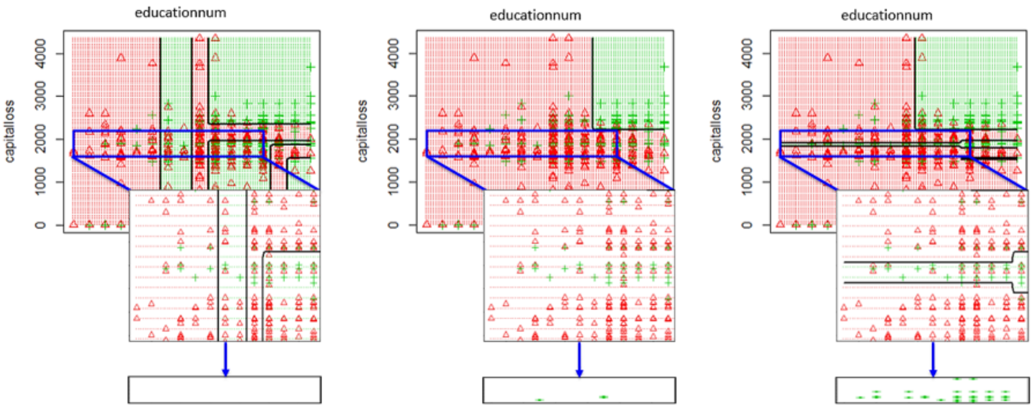

Fig. 10. Decision boundaries for 3 different participants in a federated system using the Adult dataset. An area where the decision boundaries are significantly different is highlighted as well as the training instances provided to each participant relevant to the highlighted area.

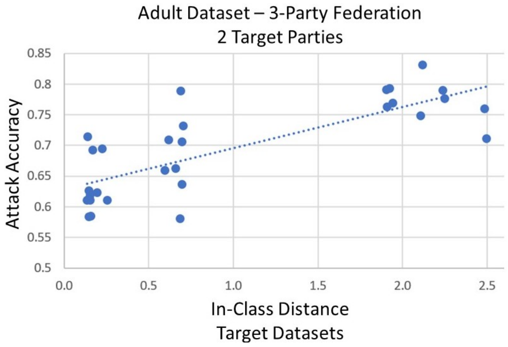

Fig. 11. In-class distance between different parties compared with insider attack accuracy for the Adult dataset with a 3-party federated learning system.

mitigation is simply a decrease in learning. If the model learns less from the training data in the first place it will naturally have less to reveal when under attack.

Differential Privacy. The definition of membership inference leads to a natural assumption that a solution lies with differential privacy. There are, however, remaining questions on the exact relationship between differential privacy and membership inference attacks as the former is a theoretical framework whereas the latter is an attack with empirical results. Further investigation into this relationship is crucial in understanding what attackers are truly learning in a membership inference attack and to what degree differential privacy curtails attackers' ability to learn this information.

For example, Figure 7 shows that even if an instance which is relatively close to the target instance is provided, the membership inference attack can still be successful. Therefore, is the attacker learning that $\mathbf{x}$ is **in** D, is the attacker learning that something *similar* to $\mathbf{x}$ is in D, or is the attacker simply learning that $\mathbf{x}$ is consistent with a decision boundary informed by D?

Additionally, if differential privacy is the answer to mitigating membership inference vulnerability, what level of privacy (value of ϵ) is sufficient to provide protection? For example, the authors in [45] launch membership inference attacks against deep learning models trained with differentially private mechanisms. While a privacy parameter of $\epsilon = 1$ is able to protect against membership inference, dropping attack accuracy to 50.8% with the CIFAR-10 dataset, the training accuracy also drops from 94.4% in the non-private case to 24.7%. Even with a modest privacy parameter of $\epsilon = 8$ the authors report an attack accuracy 58.3% with a training accuracy of just 68.6%. The authors in [46] also remark that defense mechanisms based on differential privacy are not always effective, particularly when an attacker is able to mimic the behavior of the perturbation.

Further research is necessary to understand which differential privacy mechanisms under what settings are able to provide viable protection to membership inference without forfeiting model utility.

6 Related Work

Membership inference is a young area, but there still exist a few works since [7] investigating the risk of membership inference attacks. Most of the existing proposals focus on deep learning models and are influenced by adversarial deep learning research such as [47], [48], [49]. For example, [8] identifies vulnerable instances for membership inference attacks exclusively relating to deep learning models while [10] seeks to define a measure of deep learning model vulnerability, with respect to the model's encoding of a random secret within the training data, orthogonal to membership inference. [9] studies membership inference in generative adversarial networks (GANs), and shows that the level of generalization required to mitigate against membership inference in GANs will lead to worse results in accuracy and utility. This serves as independent evidence for the mitigation strategies and research direction we promote, which include methods for anonymizing the training datasets while preserving the model training quality.

[42] proposed a measure of risk at the data instance level, and evaluated the measure on the Adult and Purchases-10 datasets with attack model, target model, and shadow model of the same type. The identified instance-level risks exemplify our analysis that membership inference attacks are data-driven.

Alternative study on membership inference relates to the impact of overfitting based on the belief that the cause of membership inference is model overfitting. [50] investigates this belief and concludes that overfitting is not necessary for a model to show vulnerability to membership inference. Their investigation is limited to the role of overfitting and assumes a powerful adversary with prior knowledge of the average training loss for F_t. This work does motivate that membership inference vulnerability is more complex than just the overfitting in the training data.

Application-specific membership inference, such as [46], studied membership inference vulnerability specific to location data under a powerful adversary with deep prior knowledge. Though this work aims at attacking aggregate data rather than a trained target model and its training data, it does demonstrate the risk of membership inference attacks in a privacy-conscious domain.

Our work is mainly inspired by [7], the first exploratory work in membership inference, which shows membership inference risks for a deep learning trained target models. In this paper we extend the work done in [7] to a more general setting towards demystifying the adverse effect of membership inference across different types of models with both general and empirical characterization of why membership inference attacks are more effective in some scenarios than in others.

7 Conclusion

We have presented the first generalized framework for the development of a membership inference attack model. This general formulation enables an in-depth characterization of membership infernece attacks against different types of machine learning models. Through extensive experimentation and empirical evidence, we show *when* and *why* machine learning models may be vulnerable to membership inference attacks. By exploring a variety of machine learning model types and their correlations with respect to the three phases of the attack generation process, we present five interesting characteristics of membership inference attacks: (1) they are data-driven attacks, (2) attack models are transferable, (3) target model type is a strong indicator of model vulnerability, (4) attack data generation techniques need not explicitly mirror the target model, and (5) membership inference attacks can persist as insider attacks in federated systems. We also include a discussion on countermeasures and mitigation methods against membership inference attacks.

Our research on membership inference attacks and membership privacy continues along several dimensions. First, we are engaged in the development of countermeasures and defense methods. Second, we are currently studying the scale and diversity of membership inference attacks in federated and collaborative learning systems. Third, we are investigating the complex relationships between membership inference attacks, membership privacy, and differential privacy.

Acknowledgments

This research is partially support by the National Science Foundation (NSF) under Grants SaTC 1564097, NSF 1547102, and a Georgia Tech IISP grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF and IISP

References

[1] *Amazon Machine Learning: Developer Guide*. Amazon Web Services, 2018.

[2] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, *Microsoft Azure: Planning, Deploying, and Managing Your Data Center in the Cloud*, 1st ed. Berkely, CA, USA: Apress, 2015.

[3] *IBM Data Science and Machine Learning*. IBM, 2018.

[4] *Cloud Machine Learning Engine Documentation*. Google, 2018.

[5] P. Lake and P. Crowther, "Data, an organisational asset," in *Concise Guide to Databases*. Springer, 2013, pp. 3–19.

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

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

[8] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *arXiv preprint arXiv:1802.04889*, 2018.

[9] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: evaluating privacy leakage of generative models using generative adversarial networks," *arXiv preprint arXiv:1705.07663*, 2017.

[10] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *arXiv preprint arXiv:1802.08232*, 2018.

[11] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: the sulq framework," in *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2005, pp. 128–138.

[12] J. Vaidya, B. Shafiq, W. Fan, D. Mehmood, and D. Lorenzi, "A random decision tree framework for privacy-preserving data mining," *IEEE transactions on dependable and secure computing*, vol. 11, no. 5, pp. 399–411, 2014.

[13] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[14] D. J. Wu, T. Feng, M. Naehrig, and K. Lauter, "Privately evaluating decision trees and random forests," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 335–355, 2016.

[15] M. De Cock, R. Dowsley, C. Horst, R. Katti, A. Nascimento, W.-S. Poon, and S. Truex, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[16] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[17] A. Sankaran, K. Guinn, and D. Nguyen, "Volume shadow copy service," *Power Solutions, March*, 2004.

[18] X. Zhu, "Semi-supervised learning literature survey," 2005.

[19] D. Sculley, "Online active learning methods for fast label-efficient spam filtering." in *CEAS*, vol. 7, 2007, p. 143.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[22] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[23] B. Parmanto, P. W. Munro, and H. R. Doyle, "Improving committee diagnosis with resampling techniques," *Advances in neural information processing systems*, pp. 882–888, 1996.

[24] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[26] Python Core Team, *Python: A dynamic, open source programming language*, Python Software Foundation, 2017. [Online]. Available: https://www.python.org/

[27] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[28] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[29] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[31] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint*, 2016.

[32] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[33] A. Rozsa, M. Günther, and T. E. Boult, "Are accuracy and robustness correlated," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 227–232.

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

[35] B. Kisačanin, "Deep learning for autonomous vehicles," in *Multiple-Valued Logic (ISMVL), 2017 IEEE 47th International Symposium on*. IEEE, 2017, pp. 142–142.

[36] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8599–8603.

[37] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[39] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[40] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 39–57.

[41] C. H. Lee and H.-J. Yoon, "Medical big data: promise and challenges," *Kidney research and clinical practice*, vol. 36, no. 1, p. 3, 2017.

[42] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *arXiv preprint arXiv:1712.09136*, 2017.

[43] A. Haeberlen, B. C. Pierce, and A. Narayan, "Differential privacy under fire." in *USENIX Security Symposium*, 2011.

[44] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 634–646.

[45] M. A. Rahman, T. Rahman, R. Laganiere, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model." *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

[46] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," *arXiv preprint arXiv:1708.06145*, 2017.

[47] M. Goodfellow and A. L. Jones, "Laceyella," *Bergey's Manual of Systematics of Archaea and Bacteria*, pp. 1–4, 2015.

[48] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[49] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv preprint arXiv:1712.07107*, 2017.

[50] S. Yeom, M. Fredrikson, and S. Jha, "The unintended consequences of overfitting: Training data inference attacks," *arXiv preprint arXiv:1709.01604*, 2017.

Stacey Truex is a PhD student in the School of Computer Science at Georgia Institute of Technology where she is advised by Prof. Ling Liu. She received her MS degree from the University of Washington in Computer Science and Systems and her BS in Computer Science and BA in Mathematics from Wake Forest University. Her research interests include data privacy, security, federated learning systems, and big data analytics.

Ling Liu is a professor in the School of Computer Science, Georgia Institute of Technology. She directs the research programs in the Distributed Data Intensive Systems Lab (DiSL). She is an elected IEEE fellow, a recipient of the IEEE Computer Society Technical Achievement Award in 2012, and a recipient of the best paper award from a dozen top venues, including ICDCS 2003, WWW 2004, 2005 Pat Goldberg Memorial Best Paper Award, IEEE Cloud 2012, IEEE ICWS 2013, ACM/IEEE CCGrid 2015, and IEEE Symposium on BigData 2016. In addition to serving as the general chair and PC chair of numerous IEEE and ACM conferences in data engineering, she has served on the editorial board of over a dozen international journals. Her current research is primarily sponsored by NSF, IBM, and Intel.

Mehmet Emre Gursoy received his MS degree from University of California Los Angeles (UCLA) and his BS degree from Sabanci University, Turkey, both in Computer Science. He is currently pursuing his PhD in the School of Computer Science, Georgia Institute of Technology, where he is advised by Prof. Ling Liu. His research interests include privacy, security, machine learning, mobile computing, and big data analytics.

Lei Yu is a PhD student in Computer Science in Georgia Institute of Technology. He received the BS and MS degree in Computer Science from Harbin Institute of Technology, China in 2004 and 2006, respectively. His research interests include data privacy, deep learning and big data analytics.

Wenqi Wei is currently pursuing his PhD in the School of Computer Science, Georgia Institute of Technology, where he is advised by Prof. Ling Liu. He received his B.E. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include data privacy, security, machine learning, and big data analytics.