On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models

Abstract: With an increase in low-cost machine learning APIs, advanced machine learning models may be trained on private datasets and monetized by providing them as a service. However, privacy researchers have demonstrated that these models may leak information about records in the training dataset via membership inference attacks. In this paper, we take a closer look at another inference attack reported in literature, called attribute inference, whereby an attacker tries to infer missing attributes of a partially known record used in the training dataset by accessing the machine learning model as an API. We show that even if a classification model succumbs to membership inference attacks, it is unlikely to be susceptible to attribute inference attacks. We demonstrate that this is because membership inference attacks fail to distinguish a member from a nearby non-member. We call the ability of an attacker to distinguish the two (similar) vectors as strong membership inference. We show that membership inference attacks cannot infer membership in this strong setting, and hence unable to infer attributes. We therefore propose a relaxed notion of attribute inference, called approximate attribute inference, and show that it is possible to infer attributes close to the true attributes under this relaxed definition. We verify our results on three publicly available datasets, five membership, and three attribute inference attacks reported in literature.

1 Introduction

The introduction of low-cost machine learning APIs from Google, Microsoft, Amazon, IBM, etc., has enabled many companies to monetize advanced machine learning models trained on private datasets by exposing them as a service. This also caught the attention privacy researchers who have shown that these models may leak information about the records in the training dataset via membership inference (MI) attacks. In an MI attack, the adversary (for instance, a user of the service) with API access to the model, can use the model's responses (for instance, class labels and probability/confidence of each label) on input records of his choice to infer whether a target input was part of the training dataset or not. This can be a serious privacy breach when the underlying dataset is sensitive, e.g., medical data.

To date, membership inference attacks have been the primary focus of studies that contemplated on traits of the datasets and machine learning models that increase or decrease the attacks' likelihood and accuracy [17, 20, 22, 25, 28]. Our focus is on a related, and perhaps a more likely attack in practice, where the adversary with partial knowledge of a target's record seeks to complete its knowledge of the missing attributes by observing the model's responses. This attack is called *model inversion* [5, 6], or in general *attribute inference* (AI) [28]. Yeom et al. [28] provide a formal definition of an AI adversary, and argue that this adversary can infer the missing attribute values by using an MI adversary as a subroutine. More precisely, for a missing attribute with t possible values, the AI adversary constructs t different input (feature) vectors, gives them as input to the MI adversary, and outputs the attribute value which corresponds to the vector that the MI adversary deems to be in the training dataset.

Beyond providing a formal definition, Yeom et al. experimentally validate the success of an AI attack on regression models, and conclude that the more overfit the model, the higher the success of the AI attack [28, §6.3]. Seeking to replicate their results on classification models (rather than regression models), where the adversary is given a partial record and its true label, our results in this paper turn out to be different. We show that even if the target classification model is susceptible to MI attacks, AI attacks on the same model have negligible advantage. Furthermore, the results persist even for highly overfitted models. We explore the reasons behind this failure, and find that in order for AI attacks to be successful, the underlying MI attack, used as a subroutine, should be able to infer membership in a stronger sense. More precisely, the MI attack should be able to distinguish between a member of the training dataset and any non-members that are *close* to that member, according to a suitable distance metric. (We consider several such distance metrics based on the nature of the dataset.) We call this, strong membership inference

(SMI), parameterized by the distance from the training dataset.

We formulate the notion of SMI, and prove that a successful MI attack does not necessarily mean a successful SMI attack. Furthermore, we also formally show that a successful SMI attack is essential for an AI attack. This latter result shows that even a standalone AI attack (which does not use an MI attack as a subroutine) is bound to fail if SMI attacks are unsuccessful. We experimentally validate these results by evaluating several proposed MI attacks from the literature on several discrete and continuous datasets, and target machine learning models, and show that while these attacks are successful in inferring membership, they fall well short as an SMI attack, and consequently as an AI attack. On the positive side (from an attacker's point of view), we propose a more relaxed notion of attribute inference, called *approximate attribute inference* (AAI), where the adversary is only tasked with finding attributes *close* to the target attributes, according to a given distance metrics. Our results show that while AI attacks are not applicable, AAI attacks perform significantly better, and improve as the target model becomes more overfit.

In more detail, our main contributions are as follows.

- We provide a formal treatment of membership and attribute inference attacks, and propose new definitions of strong membership inference (SMI), and approximate attribute inference (AAI), building on the work from [28] on the definitions of MI and AI in Section 2. We formally prove that an SMI adversary is *strictly stronger* than an MI adversary (Theorem 1), and that SMI is necessary for AI (Theorem 2).
- We experimentally validate our theoretical findings through an extensive set of experiments involving five MI attacks, three black-box and two white-box, from [17, 20, 22, 28], eight datasets (constructed from 3 main binary and continuous datasets), and several target machine learning models (neural networks, support vector machines, logistic regression, and random forests) (cf. Section 3). Our results in Section 4 validate our formal separation and show that while these attacks are successful to infer membership, they are ineffective in inferring membership at distances close to the training dataset (SMI).
- In Section 5, we further construct 3 AI attacks using the MI attacks of [22, 28] and [20] as a subroutine, and show via experiments that these attacks are not effective in inferring attributes, even if we increase the overfitting levels of the target model. On the other hand, we show that our constructed AI attacks can approximately infer attributes (AAI), with the advantage increasing as the level of overfit of the target model increases.
- Our other key findings include the reasons behind the seemingly contradictory conclusions about AI attacks on regression models [28] and classification models (our focus) in Section 5.1. We also show that the success of an MI attack is dependent on the class label of the vector; if the corresponding class occupies an overwhelmingly large portion of the feature space, then training records belonging to this class are harder to distinguish from non-members (cf. Section 4.1.3). This gives one plausible reason why MI attacks have always performed poorly on target models for binary classification problems [20, 22].

2 A Formal treatment of membership and attribute inference attacks

In this section, we formally introduce the privacy notions of strong membership inference (SMI) and approximate attribute inference (AAI). In order to define them, we need rigorous definitions of a distance metric on the feature space, missing (features) attributes of a feature vector and its relation to distance, and how the probability distribution on the feature space behaves around feature vectors. We first define these concepts in the next section followed by privacy definitions in Section 2.2.

2.1 Notation and Definitions

Feature Space. Let $\mathbb{D}$ denote a subset of the real space $\mathbb{R}$. We assume the feature space to be $\mathbb{D}^m$, where each point $\mathbf{x} \in \mathbb{D}^m$ is called a feature vector consisting of m elements/features. We assume the output space to be $Y = \mathbb{R}^*$. Let $\mathcal{D}$ be a distribution over $\mathbb{D}^m$. The *training* dataset X is defined as a multiset of n elements drawn i.i.d. from $\mathbb{D}^m$ with distribution $\mathcal{D}$. Each $\mathbf{x} \in X$ is accompanied by its *true* label $\mathbf{y} \in Y$. We denote this mapping by c, which we call the *target concept* following standard terminology [12, 21]. Thus, for each $\mathbf{x} \in X$, $c(\mathbf{x})$ denotes is true label. The term label is used generically; it may be discrete, denoting different classes, or it may be continuous, denoting the confidence or probability score for the different classes. The support of distribution $\mathcal{D}$ is defined as $\text{supp}(\mathcal{D}) = \{\mathbf{x} \in \mathbb{D}^m \mid p_{\mathbf{x}} > 0\}$, where $p_{\mathbf{x}}$ is $\Pr_{\mathcal{D}}(\mathbf{x})$ if $\mathbb{D}^m$ is discrete and $f_{\mathcal{D}}(\mathbf{x})$ if $\mathbb{D}^m$ is continuous, f being the probability density function. The notation $a \leftarrow\$ A$ indicates sampling an element a from some set A uniformly at random. The notation $\mathbf{x} \leftarrow \mathcal{D}$ denotes sampling a feature vector according to the distribution $\mathcal{D}$. Similarly, the notation $X \leftarrow \mathcal{D}^n$ denotes sampling a multiset of n feature vectors (training set) drawn i.i.d. from $\mathcal{D}$.

Machine Learning Models. A machine learning model h_X trained on X, takes as input $\mathbf{x} \in \mathbb{D}^m$ and outputs a label $\mathbf{y} \in Y$. Let $L : Y \times Y \to \mathbb{R}$ denote a loss function. The training loss of h, denoted, $L_{\text{tr}}(h)$, determines how much h differs from c on all $\mathbf{x} \in X$. Similarly we define the test loss of h by $L_{\text{test}}(h)$, which is evaluated by computing $h(\mathbf{x})$ and $c(\mathbf{x})$ over the distribution $\mathcal{D}$. For instance, if Y is discrete, then L can be the 0-1 loss function, which evaluates to $L(h(\mathbf{x}), c(\mathbf{x})) = 0$, if $h(\mathbf{x}) = c(\mathbf{x})$, and 1 otherwise [28]. The generalization error of h is defined as

$$\text{err}(h) = L_{\text{tr}}(h) - L_{\text{test}}(h). \tag{1}$$

The exact form of the loss function L depends on the learning problem. More specifically, it depends on the nature of Y. If the learning problem is that of classification among k different classes, which is our focus, we have $|Y| = k$. The true label of a sample $\mathbf{x}$ is then a k-element vector $\mathbf{y} \in Y$ with 1 in the position corresponding to the true class, and 0 in all other places. A classifier h_X however, may output a vector $\mathbf{y}' \in Y$ such that each element $y_i \in [0, 1]$ and $\|\mathbf{y}'\|_1 = 1$.

Metrics. The notions of SMI and AAI, informally introduced in the introduction, are based on the ability to distinguish nearby vectors in the feature space. The notion of "nearness" is based on a distance metric on the feature space $\mathbb{D}^m$. The examples of metrics used in this paper are Hamming distance d_H for binary datasets, i.e., over the domain $\mathbb{D}^m = \{0, 1\}^m$, and Manhattan distance d_M for normalized continuous datasets, i.e., over $\mathbb{D}^m = [-1, 1]^m$. In general, our results generalize to any *conserving* metric (See Appendix F). The following defines the distance of a non-member vector from the training dataset.

Definition 1 (Distance and Neighbors)**.** Let d be a (conserving) metric on $\mathbb{D}^m$. Let r be a positive real number and let $\mathbf{x} \in \mathbb{D}^m$. The set of r-neighbors of $\mathbf{x}$ is the r-ball centered at $\mathbf{x}$ defined as

$$B_d(\mathbf{x}, r) = \{\mathbf{x}' \in \mathbb{D}^m \mid d(\mathbf{x}, \mathbf{x}') \leq r\}.$$

A member of $B_d(\mathbf{x}, r)$ is called an r-neighbor of $\mathbf{x}$. The distance of a vector $\mathbf{x} \in \mathbb{D}^m$ from a set $X \subseteq \mathbb{D}^m$ is defined as $\min_{\mathbf{x}' \in X} d(\mathbf{x}, \mathbf{x}')$. We call $\mathbf{x}'$ the nearest neighbour of $\mathbf{x}$ in X. □

For attribute inference, we define the notion of a vector with missing attributes as *portion*:

Definition 2 (Portions)**.** We introduce a special symbol $*$ called star, and define $\mathbb{D}^* = \mathbb{D} \cup \{*\}$. Let S be a subset of indexes from $[m]$, which we call the set of unknown features. We define the map $\phi_S : \mathbb{D}^m \to \mathbb{D}^{*m}$, which given as input a feature vector $\mathbf{x}$ outputs a vector $\mathbf{x}^*$, such that $x_i^* = *$ for each $i \in S$ and $x_i^* = x_i$ for all $i \notin S$. We call $\mathbf{x}^* = \phi_S(\mathbf{x})$ a *portion* of $\mathbf{x}$ under S, or simply a portion of $\mathbf{x}$ if reference to the set S is not relevant. The set of features that are *masked*, i.e., replaced by $*$, in $\phi_S(\mathbf{x})$ will be called the *unknown part* of $\mathbf{x}^*$. □

Definition 3 (Siblings)**.** Define the set:

$$\Phi_S(\mathbf{x}) = \{\mathbf{x}' \in \mathbb{D}^m \mid \phi_S(\mathbf{x}) = \phi_S(\mathbf{x}')\},$$

then $\Phi_S(\mathbf{x})$ is called the set of siblings of $\mathbf{x}$ under S, and any member of the set a sibling of $\mathbf{x}$ under S. Note that $\mathbf{x}$ is also a sibling of itself.

For attribute inference, the algorithm will be given a portion $\mathbf{x}^* = \phi_S(\mathbf{x})$, such that the feature corresponding to the set S will be missing (unknown). The set $\Phi_S(\mathbf{x})$ contains all vectors which could possibly have the portion $\mathbf{x}^*$, including the original vector $\mathbf{x}$. These are the possible *candidates* of the portion, and the algorithm would need to distinguish them from $\mathbf{x}$. In the extended version of this paper, we show that given a vector $\mathbf{x}$, all of its possible portions with i unknown features are within a ball whose radius can be determined through i. This result is useful to show the link between attribute inference and strong membership inference, as we shall see later.

In some of our inference definitions, we would need to sample vectors in the vicinity of some feature vector $\mathbf{x}$. Depending on the distribution $\mathcal{D}$, it may well be the case that the vectors around $\mathbf{x}$ have a negligible probability of being sampled as feature vectors. Thus, the adversary may simply be able to infer non-membership by checking which vector is not likely to be sampled under $\mathcal{D}$ [28]. To overcome this technical issue, we assume that the distribution $\mathcal{D}$ is such that it assigns more or less similar probabilities to all feature vectors within a small radius around $\mathbf{x}$. This is made precise by the following definitions.

Definition 4 (Induced Distribution)**.** Let Z be a set of feature vectors. Define $Z_{\mathcal{D}} = \text{supp}(\mathcal{D}) \cap Z$. We say that a vector $\mathbf{z}$ is sampled from Z according to the distribution induced by $\mathcal{D}$ if the resulting random variable has probability mass function $\frac{p_{\mathbf{z}}}{\sum_{\mathbf{z}' \in Z_{\mathcal{D}}} p_{\mathbf{z}'}}$ or the probability density function $\frac{p_{\mathbf{z}}}{\int_{Z_{\mathcal{D}}} f_{\mathcal{D}}(\mathbf{z}') d\mathbf{z}'}$ in the continuous case. □

Note that the probabilities are only defined if $Z_{\mathcal{D}}$ is non-empty. We shall always assume this to be the case.

Definition 5 (Indistinguishable Neighbour Assumption)**.** Let $r > 0$, and let d be a metric. Let $\mathbf{x} \leftarrow \mathcal{D}$. Then there exists at least one vector $\mathbf{x}' \in B_d(\mathbf{x}, r)$ such that for any algorithm (distinguisher) $\mathcal{A}$, the probability that $\mathcal{A}$ outputs 1 given $\mathbf{x}$ minus the probability that $\mathcal{A}$ outputs 1 given $\mathbf{x}'$ sampled from $B_d(\mathbf{x}, r)$ according to the distribution induced by $\mathcal{D}$, is at most $\epsilon(r)$. We call $\epsilon(r)$, the r-neighbour distinguishability advantage. □

The above assumption states around any vector $\mathbf{x}$, there is at least one vector sampled according to the distribution induced by $\mathcal{D}$, such that any algorithm cannot distinguish which one is sampled through the original distribution versus the induced distribution. We argue that this is a plausible assumption. For instance, consider the Purchase (shopping transactions) dataset [2], which records the items bought by customers; 1 if the corresponding item is purchased by the customer and 0, otherwise. Given any vector $\mathbf{x}$, a nearby vector where a few item purchases have been flipped can barely be considered an anomaly. Further note that the ability to distinguish increases, the further we move from the original vector, since now there are other vectors likely to be sampled through the induced distribution which are starkly different from $\mathbf{x}$, i.e., at greater distance from $\mathbf{x}$. Hence, the advantage $\epsilon(r)$ is defined as a function of r.

Decision Regions. Our final definition in this section is that of decision regions, i.e., regions in the feature space assigned to a given class. We shall show later that performance of membership inference is linked to the volume of decision regions. Let $k \geq 2$ be the number of classes.

Definition 6. Given a classifier h_X, for each class $j \in [k]$, we define its *decision region* (DR) as

$$\mathcal{R}_j = \{\mathbf{x} \in \mathbb{D}^m : h_X(\mathbf{x}) = j\} \tag{2}$$

This is analogous to the definition of acceptance region in [30]. Similar to [30], we sample a large number of feature vectors from $\mathbb{D}^m$ uniformly at random, and use the fraction of vectors labelled j by h_X to estimate the *fractional volume* of the decision region $\mathcal{R}_j$. Overloading notation, we shall use decision region to mean both the region and its fractional volume. A class is said to *dominate* another class if the DR of the former is larger than the DR of the latter. The class with the largest DR shall be called the *most dominant* class.

2.2 Formalization: Membership and Attribute Inference

Membership Inference. Our first definition is that of membership inference which is derived from the definition in [28].

Experiment 1 (Membership Inference (MI) [28])**.** Let $\mathcal{A}$ be the adversary, let $X \leftarrow \mathcal{D}^n$ be the input dataset.

1. Construct model h_X.
2. Sample $b \leftarrow\$ \{0, 1\}$.
3. If $b = 0$, sample $\mathbf{x} \leftarrow \mathcal{D}$.
4. Else if $b = 1$, sample $\mathbf{x} \leftarrow\$ X$.
5. $\mathcal{A}$ receives $\mathbf{x}$, $c(\mathbf{x})$ and oracle access to h_X.
6. $\mathcal{A}$ announces $b' \in \{0, 1\}$. If $b' = b$, output 1, else output 0.

Using the True Label. Note that in addition to the vector $\mathbf{x}$, its true label $c(\mathbf{x})$ is also given to the adversary. This then allows the adversary to compute the loss function $L(h_X(\mathbf{x}), c(\mathbf{x}))$ from the output of the model h_X. This is considered for instance in [28], the shadow model technique in [22] and the shadow model variants of membership inference attacks in [20]. However, note that the true label is not necessarily required as is demonstrated in one of the attacks in [20] which only uses the knowledge of the input sample and the prediction returned by h_X. In this case, the adversary simply ignores the true label $c(\mathbf{x})$. The same is true in all the other experiments (definitions) to follow.

Let $\text{Exp}_{\text{MI}}(\mathcal{A}, h, n, \mathcal{D})$ denote the output of the above experiment.

Definition 7 (Membership Inference Advantage)**.** The membership inference advantage of $\mathcal{A}$ on the classifier h, i.e., $\text{Adv}_{\text{MI}}(\mathcal{A}, h, n, \mathcal{D})$, is defined as

$$\begin{aligned} &\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0] \\ &= \Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1] \end{aligned}$$

It is the thesis of this paper that an MI adversary with a significant advantage in distinguishing between members and non-members is due to the fact that non-members are at a significant distance away from member vectors. If on the other hand a non-member vector is close to a member vector, then the adversary may not be able to distinguish between the two. We therefore present another definition of membership inference, called strong membership inference (SMI) defined next. The definition challenges the adversary to distinguish between two neighbouring feature vectors. The closeness of the two vectors is controlled by the parameter r in the definition. We show later why such a strong inference attacker is a better starting point for constructing an attribute inference attacker in the spirit of [28].

Experiment 2 (r-Strong Membership Inference (SMI))**.** Let $\mathcal{A}$ be the adversary, let $X \leftarrow \mathcal{D}^n$ be the input dataset, let d be a (conserving) metric, and let $r > 0$ be a real number.

1. Construct model h_X.
2. Sample $b \leftarrow\$ \{0, 1\}$.
3. Sample $\mathbf{x}_0 \leftarrow\$ X$.
4. If $b = 0$, sample $\mathbf{x}$ from $B_d(\mathbf{x}_0, r)$ according to the distribution induced by $\mathcal{D}$ (cf. Definition 4).
5. Else if $b = 1$, $\mathbf{x} = \mathbf{x}_0$.
6. $\mathcal{A}$ receives $\mathbf{x}$, $c(\mathbf{x})$ and oracle access to h_X.
7. $\mathcal{A}$ announces $b' \in \{0, 1\}$. If $b' = b$, output 1, else output 0.

Definition 8 (Strong Membership Inference Advantage)**.** The SMI advantage of $\mathcal{A}$ on the classifier h, i.e., $\text{Adv}_{\text{SMI}}(\mathcal{A}, h, r, n, \mathcal{D})$, is defined as

$$\begin{aligned} &\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0] \\ &= \Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1] \end{aligned}$$

Relationship between MI and SMI. SMI is the same as MI if r is large enough to encompass all feature vectors in the support of $\mathcal{D}$. Otherwise, the next theorem shows that the two definitions are not equivalent.

Theorem 1. *There exists a domain $\mathbb{D}^m$, a distribution $\mathcal{D}$ on the domain, an $r > 0$, a dataset $X \leftarrow \mathcal{D}^n$, a classifier h, and an algorithm $\mathcal{A}$ such that an MI adversary gains non-negligible advantage using $\mathcal{A}$ whereas an SMI adversary has 0 advantage using the same algorithm.*

Proof. See Appendix G. □

The proof of the above result essentially constructs a dataset such that the output of the classifier is constant around any vector $\mathbf{x}$ in the dataset. In real-world dataset, this implies that we assume the output of the classifier to be nearly constant around any feature vector $\mathbf{x}$, thus making it hard for an SMI attack to distinguish non-members in the vicinity of members. We shall later show that this assumption holds for real-world datasets and classifiers.

Attribute Inference. We first start with the definition of attribute inference derived from [28].

Experiment 3 (Attribute Inference (AI) [28])**.** Let $\mathcal{A}$ be the adversary, let $X \leftarrow \mathcal{D}^n$ be the input dataset, and let S be a subset of $[m]$ with cardinality m' such that $1 \leq m' < m$.

1. Construct model h_X.
2. Sample $b \leftarrow\$ \{0, 1\}$.
3. If $b = 0$, sample $\mathbf{x} \leftarrow \mathcal{D}$.
4. Else if $b = 1$, sample $\mathbf{x} \leftarrow\$ X$.
5. Let $\mathbf{x}^* = \phi_S(\mathbf{x})$ be a portion of $\mathbf{x}$.
6. $\mathcal{A}$ receives $\mathbf{x}^*$, $c(\mathbf{x})$ and oracle access to h_X.
7. $\mathcal{A}$ announces $\mathbf{x}' \in \mathbb{D}^m$. If $\mathbf{x}' = \mathbf{x}$ output 1, else output 0.

Definition 9 (Attribute Inference Advantage)**.** The AI advantage of $\mathcal{A}$ on the classifier h, i.e., $\text{Adv}_{\text{AI}}(\mathcal{A}, h_X, m', n, \mathcal{D})$, is defined as

$$\begin{aligned} &\Pr[\text{Exp}_{\text{AI}}(\mathcal{A}, h_X, m', n, \mathcal{D}) = 1 \mid b = 1] \\ &- \Pr[\text{Exp}_{\text{AI}}(\mathcal{A}, h_X, m', n, \mathcal{D}) = 1 \mid b = 0]. \end{aligned}$$

The above definition mirrors the one from [28]. However, the attribute inference covered in [28] is more general; it considers background knowledge about $\mathbf{x}$, and not necessarily a portion. The version that we consider is called the model inversion attack [6, 28].

Inferring through the Distribution vs the Model. Note that these definitions purposely define advantage as the difference between inferring through the distribution alone versus inferring via access to the model. For instance, one way to infer the missing features is to exploit statistical correlations between the observed features and the label. But notice that this can be done directly through knowledge of the distribution, irrespective of access to the model. The AI advantage will therefore be negligible for such a strategy. Hence, the definitions only define an AI attack as advantageous if it can infer more through the model as opposed to through statistical trends of the feature vectors. The same applies to approximate attribute inference to be defined shortly.

Relationship between AI and SMI. It is easy to see how an AI adversary can use an SMI adversary to infer attributes. Given a portion $\mathbf{x}^* = \phi_S(\mathbf{x})$, the AI adversary uses the size of S, i.e., m', to choose an r according to Corollary 1, in Appendix F, and then runs the SMI adversary with input r and each possible *sibling* of the vector $\mathbf{x}$ (Even though the set S is not explicitly given to the AI adversary, it is implicit from the portion). Whenever, the SMI adversary outputs 1, i.e., predicts the corresponding vector to be a member, our AI adversary outputs that vector as its guess for $\mathbf{x}$.

In the other direction, the following theorem shows that AI implies SMI, or in other words $\neg$SMI $\Rightarrow$ $\neg$AI. Therefore, if an SMI adversary has negligible advantage, then we cannot hope to find an AI adversary with significant advantage.

Theorem 2. *Let $\mathcal{A}$ be an AI adversary with advantage δ. Then there exists an SMI adversary $\mathcal{B}$ with advantage $\delta + \epsilon(r)$, where $\epsilon(r)$ is the r-neighbor distinguishability advantage.*

Proof. See Appendix G. □

The above observation is mirrored by our experiments where we show that constructing an attacker that can *exactly* predict the missing values of a portion of a member vector with high probability is highly unlikely. Thus, we propose below the definition of an approximate AIA, that requires the attacker to predict the missing values only "approximately close" to a member vector.

Experiment 4 (Approximate Attribute Inference (AAI)). Let $\mathcal{A}$ be the adversary, let $X \leftarrow \mathcal{D}^n$ be the input dataset, let S be a subset of $[m]$ with cardinality m' such that $1 \leq m' < m$, and let $\alpha \geq 0$ be a distance parameter.

1. Construct model h_X.
2. Sample $b \leftarrow\!\$ \{0, 1\}$.
3. If $b = 0$, sample $\mathbf{x} \leftarrow \mathcal{D}$.
4. Else if $b = 1$, sample $\mathbf{x} \leftarrow\!\$ X$.
5. Let $\mathbf{x}^* = \phi_S(\mathbf{x})$ be a portion of $\mathbf{x}$.
6. $\mathcal{A}$ receives $\mathbf{x}^*$ and oracle access to h_X.
7. $\mathcal{A}$ announces $\mathbf{x}' \in \mathbb{D}^m$. If $d(\mathbf{x}', \mathbf{x}) \leq \alpha$ output 1, else 0.

Definition 10 (Approx. Attribute Inference Advantage). The AAI advantage of $\mathcal{A}$ on the classifier h, i.e., $\text{Adv}_{\text{AI}}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D})$, is defined as

$$\begin{aligned} &\Pr[\text{Exp}_{\text{AI}}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D}) = 1 \mid b = 1] \\ &- \Pr[\text{Exp}_{\text{AI}}(\mathcal{A}, h_X, m', n, \alpha, \mathcal{D}) = 1 \mid b = 0]. \end{aligned}$$

Note that with $\alpha = 0$, Experiment 3 becomes a special case of Experiment 4. It is easy to see that AI $\Rightarrow$ AAI, but the converse is not necessarily true.

Computing Advantages in Practice. As most prior work on membership inference uses the Area Under the Curve (AUC) of a Receiver Operating Characteristics (ROC) curve as a measure of aggregated classification performance of the MI attacker (viewed as a binary classifier), we use the same metric in our experiments in Section 3. In Appendix H, we show how our advantage definitions 7 and 8 are related to the AUC statistic. For the evaluation of AI and AAI attacks we employ the advantage metrics defined in Definitions 9 and 10.

3 Experimental Methodology

In this section, we describe the datasets, instances of MI and AI attacks used, and how we carry out membership and attribute inference attacks in our experiments in Sections 4 and 5. We first evaluate the performance of several MI attacks in terms of MI advantage (Def. 7) with increasing distance of the challenge vectors from the training set (Section 4). We then evaluate the performance of AI attacks in terms of AI advantage (Def. 9) which use MI attacks as a subroutine (Section 5.1). Finally, we study the performance of the same AI attacks in the sense of approximate attribute inference (Def. 10). These experiments demonstrate the shortcomings of MI and AI definitions and the need for our newly proposed definitions, i.e., SMI and AAI.

3.1 Data and Machine Learning Models

We evaluate MI and AI attacks on three different datasets: (a) *Location:* a social network locations check-in dataset obtained from Foursquare [26], (b) *Purchase:* a shopping transactions dataset [2], and (c) *CIFAR* an image dataset [13]. These datasets have previously been used to demonstrate MI [10, 20, 22] and AI attacks [10]. The first two datasets are binary, with 467 binary features in Location and 699 in Purchase, whereas the CIFAR dataset was processed, using principal component analysis (PCA), to yield 50 continuous features normalized between -1 and 1 [10]. We applied k-means clustering to obtain class labels in both the Location and Purchase datasets. The number of classes in the Location dataset is 30 and for the Purchase dataset, we create 5 variants differing in the number of classes (2, 10, 20, 50, 100), as is done in [20]. Finally, the CIFAR dataset contains 100 class labels for the images, with an additional set of 20 labels which are a superset of the 100 classes, e.g. the label “flowers” is the superset of orchids, poppies, roses, sunflowers, and tulips. We call the two datasets CIFAR-100 and CIFAR-20.

We predominantly explore the neural network as our target model. However, later in Section 4.2, we show that our observations generalize to Logistic Regression, Support Vector Machine, and Random Forest classifiers. The exact configurations of these models for each experiment are detailed in Appendix A.

3.2 MI and AI Adversaries

We use five MI attacks from literature as examples of an MI adversary (Def. 7), and three AI attacks as examples of an AI adversary (Def. 9).

3.2.1 MI Attacks

Our MI attacks include three black-box attacks: the shadow model based attack from Shokri et al. [22], the attack from Yeom et al. based on prediction loss [28], and the attack from Salem et al. based on maximum prediction confidence [20], and two variants (local and global) of a white-box attack from Nasr et al. [17]. Recall that in an MI attack, the attacker is given a member or a non-member vector with optionally its true label, and is asked to infer membership.

Shadow MI [22]. This attack trains a machine learning model, called an attack model, to discern membership of a given vector from the prediction output vector (confidence of every class label). This attack model leverages outputs from shadow models which are trained with a disjoint dataset to mirror the behaviour of the target model.

Loss MI [28]. This attack eliminates the high computational cost of training shadow and attack models by evaluating the prediction loss of a vector on the target model directly. This attack, in practice, may use the target model training loss as a loss threshold to determine membership.

Conf MI [20]. Conf MI, short for Confidence, is even simpler than Loss MI; instead of computing the prediction loss, the attack simple uses the confidence value of the most likely label. With less information available to the attack, it performs worse than both Loss MI and Shadow MI (as we shall see in Section 4). However, it is arguably a more practical attack, requiring less information.

Local White Box (WB) and Global White Box (WB) MI [17]. The three previous attacks are all black-box attacks with little to no information about the target model, and only API access to the model. An alternative form of MI attack is a white-box membership inference attack, which in a federated setting, may offer additional information for an adversary to launch an MI attack. Despite the federated setting, we suspect any observations we perform on the black-box setting should be reflected in a white-box setting. Nasr et al. attack [17] is a standalone attack targeting federated machine learning models in a white-box setting. The white-box setting lends additional hidden layer information and intermediate model states from the training process to better inform the attack model. This information includes the final layer gradients, outputs and the true label, obtained from intermediate and final states of the target model.

The federated setting consists of multiple parties, each training models independently and contributing parameters to a central server. The server aggregates these parameters before sending the results back to each party to replace their individual model. Two different attacks are tested: the *Global WB MI* attack, where the attacker has server level information and attacks each of the parties individually (in the case of a Malicious MLaaS provider); and the *Local WB MI* attack whereby the attacker is an external or contributing party attacking the server or MLaaS provider.

3.2.2 Attribute Inference (AI) Attacks

We use three AI attacks as examples of an AI adversary. All three attacks use an MI attack as a subroutine as mentioned in Section 2. We, therefore, use the same names for them as the underlying MI attacks. Briefly, our general procedure to evaluate an AI attack is as follows. Given a portion $\mathbf{x}^* = \phi_S(\mathbf{x})$ for a set S of unknown features (cf. Def. 2), we first construct all siblings of $\mathbf{x}$ (cf. Def. 3), by trying all possible permutations of the missing attribute(s), i.e., features. We then give each sibling as input to the MI attack. From the set of siblings, the vector with the highest membership confidence from the underlying MI attack is deemed the original vector $\mathbf{x}$, and thus its attributes identified as the missing attributes.

Shadow AI. The basis of this attack is to use the attack model from Shadow MI [22] for AI. While the MI version of the attack only uses the final decision (member or non-member), in the AI attack, we use the prediction confidence from the attack model to gauge which vector is most likely the original vector, and thus infer attributes.

Loss AI [28]. This attack follows the original proposal from Yeom et al. to use the training loss as the deciding factor for attribute inference. Given all siblings, the vector that achieves the prediction loss (from the target model) closest to the training loss, is flagged as the original vector.

Conf AI [29]. Recall that Conf MI [20] uses the single largest prediction confidence of the vector to deduce its membership. We repeat the same process, and flag the highest confidence vector (prediction confidence from the target model) from all siblings as the original vector.

Note. Although both Local WB and Global WB MI attacks can also be used to perform AI, we opted against, as they are computationally more demanding than other attacks. Fortunately, as we shall show, Local WB and Global WB MI attacks show similar trends as the other 3 MI attacks we use as subroutines for AI.

3.3 Attack Methodology

Prior to inference, we must first train a target model on a given dataset. To do so we split the dataset into training and testing sets. We describe the exact training/testing data split, the architecture of the neural network, and other hyper-parameters in Appendix A. These models have been tuned to replicate models observed in prior works. The training set is used to train the target model, and the prediction accuracy of the target model is evaluated on the testing set. We tune our target models to produce prediction accuracies comparable to [22] (exact attack accuracy values are reported in Table 4 in Appendix A). From the training and testing sets we then sample 1000 vectors each to serve as our member and non-member sets. With the target model prepared, we take the following steps to launch MI and AI attacks.

MI. For MI, we obtain AUCs by evaluating the member and non-member subsets with either the MI attack model (for Shadow, Local WB and Global WB MI), or the target model (for Loss and Conf MI) for a membership confidence score.

AI. For AI, we take our set of member and non-members, and then use the top most informative features according to the Minimal Redundancy Maximal Relevance (mRMR) criterion [19]. The set of most informative features forms the set S of unknown features. For each vector, we then create its portion based on S, and generate all siblings of the vector, only one of which is the original vector with the target attribute values. With this set of siblings, for each member and non-member vector, we perform an MI attack. This produces a measure of membership confidence (either as attack model probability, prediction loss, or prediction confidence, c.f. Section 3.2.2). From this measure, the sibling with the highest membership confidence is regarded as the correct vector, and consequently containing the correct missing attributes. For AI, we regard the attack as a success when the recovered sibling is exactly equal to the original vector (Exp. 3). For AAI, we regard the attack a success when the recovered sibling is within a given α distance away from the correct attributes (Exp. 4).

4 Membership Inference

We first show results from MI attacks highlighting the need for our definition of strong membership inference (SMI) (Exp. 2). Two key findings are:

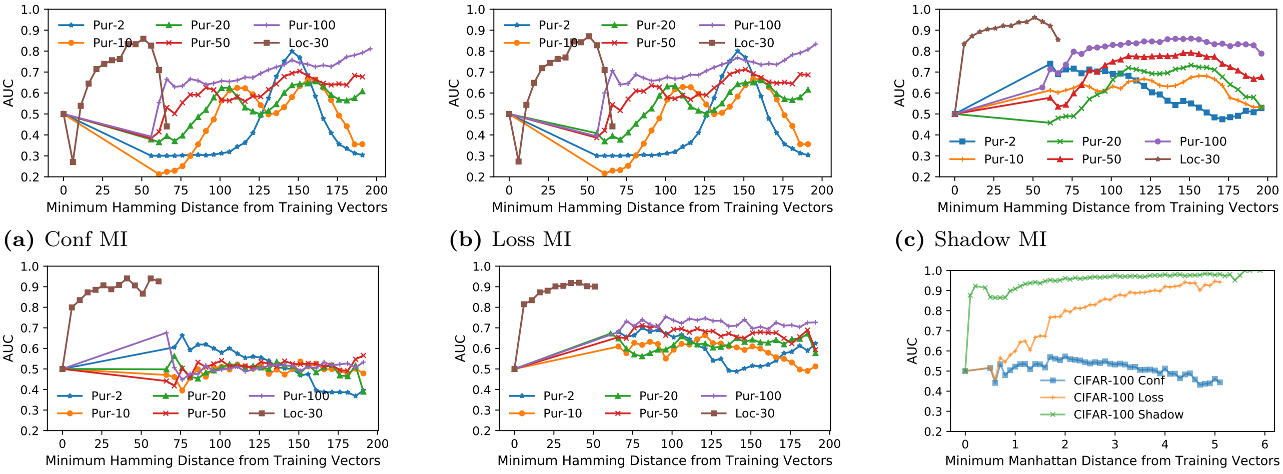
- MI attacks perform better if the non-members are at a greater distance from the training dataset. This observation is crucial for attribute inference, as we shall see in the next section.
- MI attack performance is not uniform across all classes in the dataset. In fact, it is inversely related to the dominance of the class, i.e., the decision region of the class (Def. 2).

4.1 MI Attacks on Neural Networks

We first inspect the performance of the five MI attacks (See Section 3.2.1) on members and non-member vectors from the original dataset as a function of their distance from the training dataset (Def. 1). We observe that the vectors in the original dataset are quite far away from each other, consequently lacking MI performance information at small distances. Thus we follow this analysis with MI performance on synthetically generated vectors, to illustrate a complete picture of MI performance as a function of distance from the training dataset (Section 4.1.2). We also explore the relationship between MI attack performance and the decision region of a class (Section 4.1.3).

4.1.1 MI Performance on the Original Dataset as a Function of Distance

After training the target model, we compute the distance of each non-member vector from the training set.



(a) Conf MI **(b)** Loss MI **(c)** Shadow MI **(d)** Local WB MI **(e)** Global WB MI **(f)** CIFAR-100

Fig. 1. Increasing AUC of various MI attacks with increasing Hamming distance of original non-members from the training dataset on target models. Subplot (f) compares the difference in attack AUC between MI attacks on CIFAR-100 (CIFAR-20 can be found in Appendix C).

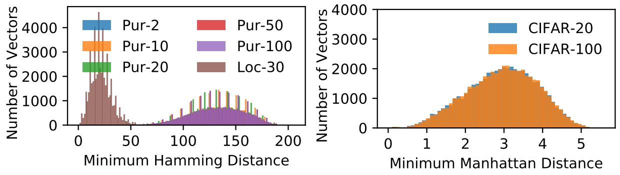
Recall from Section 2, we use Hamming distance d_H for Location and Purchase datasets (which are binary), and Manhattan distance d_M for the continuous (normalized) CIFAR datasets. The vectors are then grouped according to their distance from the training dataset (the distance is 0 for members). We then calculate AUC for each distance by taking the membership score of each vector in this distance group as the negative class, and all member vectors as the positive class. This test is repeated 50 times (10 for the WB MI attacks due to computational resource limitations), and the AUC is computed on the aggregation of all confidence values (Fig. 1).

Results. From Figs. 1a to 1e, we observe that for the Location dataset the AUC improves as the distance of non-members from the training dataset increases in all five MI attacks, with the AUC being closer to random guess (0.5) for non-members closest to the training dataset. From the same figures, we can see that this trend is less obvious for the Purchase datasets. This is mainly because non-members in the Purchase datasets are at a greater distance from the training dataset. The same observation can be made for CIFAR-100 in Fig. 1f (results for CIFAR-20 are in Appendix C). This gives a first indication that SMI (Exp. 2) is less successful than MI (Exp. 1).

An issue with the results in Figure 1 is that there is a lack of vectors close to and farthest away from the training datasets. This is evident from the distribution of distances displayed in Fig. 2. Observe that there is little data available when we attempt to inspect AUC for distances close to the original dataset. As the non-members in the original Purchase datasets do not provide a full picture of how the MI performance behaves across all distances, and hence MI performance, in the next section, we generate synthetic vectors allowing us to control the distance (Hamming or Manhattan) from the training dataset providing a more complete picture.

A few other observations are worth highlighting:

- Consistent with what has been previously reported on MI attacks, the attack accuracy improves on target models with a greater number of classes [20, 22]. Higher number of classes is also linked to a higher degree of overfitness (Table 4).
- The AUC performance of the Loss and Conf MI attacks is almost identical. Recall that Conf MI uses the maximum confidence value of the prediction, while Loss MI uses the prediction loss. Note that the prediction loss for a classification model is simply the loss between the confidence of the true label and 1. Given that a (good) target model is likely to predict the correct label of the vector, it follows that, most of the times, the maximum prediction confidence (as used in Conf MI) will be equal to the confidence used to compute the loss in Loss MI.
- Some of the AUCs exhibit peaks; an increase as the distance from the training dataset increases followed by a decrease. This is due to the decision regions (DR) learnt by the classifiers. We shall elaborate on this in Sections 4.1.2 and 4.1.3.
- Another peculiar observation is that some of the AUCs drop below 0.5, meaning that the strategy employed by the corresponding MI attack predicts



(a) Hamming distance **(b)** Manhattan distance

Fig. 2. Histogram of distances of non-members from members in our training datasets. This data distribution is consistent across all attacks.

flips and applies more to non-members than to members. The potential reason behind this is the same as the observation above which we shall explain in Section 4.1.3.

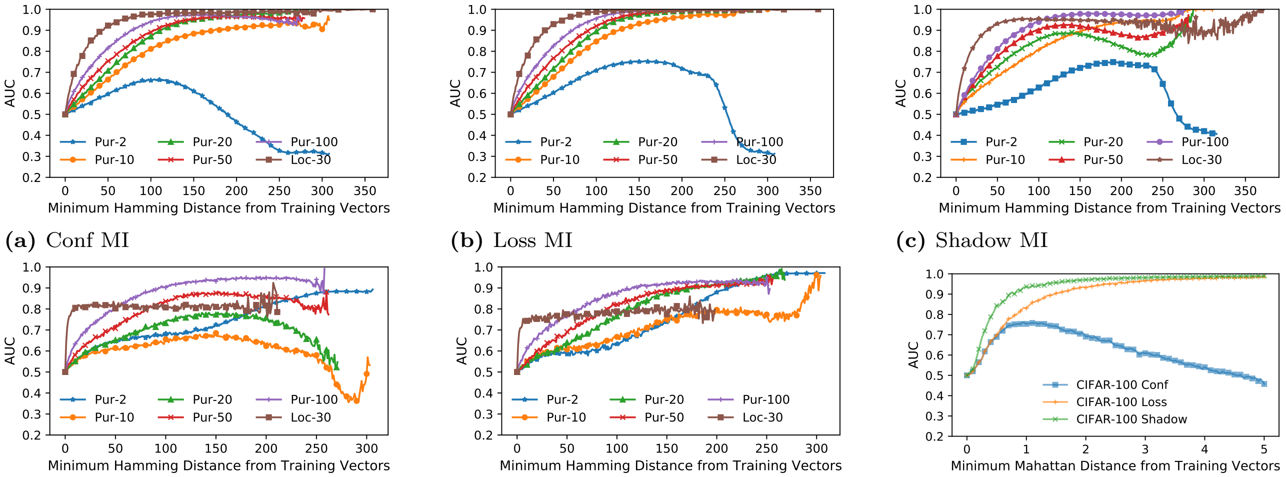
Observation 1. *In the MI attacks reported in literature, the distance of non-members from the training dataset is large. In general, an MI attack is more likely to accurately predict a non-member, the greater its distance from the training dataset.*

4.1.2 MI Performance on Synthetic Non-Members as a Function of Distance

To generate synthetic vectors for the binary datasets (Location and Purchase), we (a) randomly select a member of the training set, (b) randomly select features to invert, (c) and vary the number of features and generate 5 non-members for each distance group, ranging from Hamming distance 1 to, 467 for Location, and 599 for Purchase. For CIFAR datasets, we define Manhattan distance groups at increments of 0.05 from the training dataset, starting from 0.05 to 5. We then produce non-members by randomly selecting features and adding additive perturbations to the feature values of the original vector. The process is repeated 5 times for each Manhattan distance group. The entire process is repeated for all selected 1000 member vectors for each dataset. The distance to the training dataset is recomputed for all non-members, to cater for the event that the nearest neighbor of a non-member in the training dataset has changed. The vectors thus generated are non-members, with the same label as the original member, unless, by chance, any of them collides with a member, in which case we discard it. We also ensure that the nearest neighbor in the dataset of the newly generated vector is of the same label as the base member vector, if not, this generated vector is discarded.

Results. The AUCs of the five MI attacks are displayed in Fig. 3. For all five attacks, we observe that the AUC is close to 0.5 for vectors close to the training dataset, and starts improving as the distance from training dataset increases. It is also evident that the higher the number of classes, the steeper the improvement in AUC as the Hamming distance increases for the Location and Purchase datasets. This is more obvious through the magnified Fig. 6, where we show AUC of the Conf MI attack on the Location, Purchase and CIFAR datasets at smaller distances from the datasets. The AUC is below 0.6 for Hamming distances of less than 5 and Manhattan distance of less than 0.2. This implies that the MI attack is not successful enough in the stronger sense, i.e., in the sense of SMI (Def. 8). This has implications for attribute inference, as we shall see in Section 5.

On datasets with higher number of classes, the AUCs of Loss MI (Fig. 3b), Local WB (Fig. 3d) and Global WB (Fig. 3e) MI, show little change after a certain distance, even if the distance of non-members from the training dataset increases. On the other hand, on the Purchase datasets, for smaller number of classes (2, 10 and 20), Conf (Fig. 3a), Loss (Fig. 3b) and Shadow (Fig. 3c) MI attacks observe an increase in AUC followed by a decrease. For the 10 and 20 class variants, we see a second incline in the AUC performance of Shadow MI around a Hamming distance of 250. The reason for this is that at certain distances a non-member vector $\mathbf{x}'$ with a class label j, might be in the decision region of another class, even when the nearest neighbor of $\mathbf{x}'$ in the dataset has the class label j. We elaborate this in the next section. Interestingly, in Fig. 3f, the AUC curves of Conf and Loss MI diverge as the Manhattan distance from the training dataset grows greater than 0.7-0.8. This is because at larger Manhattan distance, the target model starts giving incorrect label predictions. The Loss MI attack detects this (as it computes loss with the predicted confidence). On the other hand, Conf MI only uses the highest confidence. It is therefore unable to detect this, showing worse performance. Finally, we note that a few of the AUC lines are ragged, especially at distances furthest away from the datasets. This is exhibited by attack model based MI attacks (Shadow, Local and Global WB). This is because the underlying attack models have less exposure to vectors at large distances as a result of the data distribution (c.f. Fig. 2a, corresponding to distances where the AUC lines becomes ragged). The AUC curves of Loss and Conf MI are smooth throughout.



(a) Conf MI **(b)** Loss MI **(c)** Shadow MI

(d) Local WB MI **(e)** Global WB MI **(f)** CIFAR-100

Fig. 3. Increasing AUC of various MI attacks with increasing Hamming distance of synthetic non-members from the training dataset on target models. (f) compares the difference in attack AUC between MI attacks on CIFAR-100 (CIFAR-20 can be found in Appendix C).

Observation 2. *The existing success of MI is a consequence of most non-member vectors being very different to members in terms of distance. For non-member vectors very close to members, the MI attacks perform similar to a random guess (0.5 AUC), and hence fail in the sense of SMI. Thus, the incumbent definition of MI does not capture the behavior of an MI adversary for non-members at distances close to the training data, i.e., SMI, which is essential for launching attribute inference attacks (Theorem 2).*

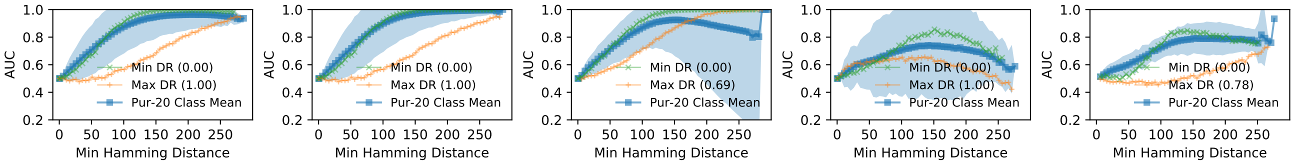
4.1.3 MI performance on Synthetic Non-Members as a Function of Class Label and Distance

The results thus far have been averaged over members and non-members from all classes. However, as we shall show, the performance of the MI attacks is not consistent over all classes. In fact, the more dominant a class, i.e., the larger the decision region (DR) of the class (Def. 6), the less likely it is to be susceptible to membership inference. We empirically measure the decision region of a given class by sampling one million vectors from the feature space by sampling each feature uniformly at random within feature bounds (see feature bounds in Section 3.1). A similar approach has been adopted in [30] for binary classification.

For per-class analysis, we train the target model and generate the synthetic vectors as before, except that now not only do we group synthetic vectors by the distance from the training dataset, but also according to the class label of the nearest training dataset vector. Due to space restrictions, we only show results for the Purchase-20 dataset. Results from the other datasets are in agreement with the conclusions drawn here, and are presented in Appendix D. In the figures, we highlight the AUC performance of the most dominant (largest DR) and least dominant (smallest DR) classes.

Results. Each plot in Fig. 4 has 4 salient features. A blue line representing the mean AUC of all classes, an accompanying blue shaded area representing 2 standard deviations of AUC between classes, a green and blue line representing the class with the smallest DR, and the largest DR, respectively. From Fig. 4, we observe that across all MI attacks, the AUC of the most dominant class is well below the average. In particular, at distances close to the dataset.

This can be explained as follows. Near the dataset, a non-member vector with class label j (which is also the label of its nearest neighbor in the dataset) is likely to lie in the decision region $\mathcal{R}_j$ of class j. As we move away from the dataset, by varying the distance, the corresponding non-member vectors shift further away from the spot in the decision region occupied by their nearest neighbors in the dataset. At certain distance, depending on the target or attack model, the decision region changes to a decision region occupied by a different class, even though the nearest neighbor still has the class label j. These non-members are then likely to be misclassified as member vectors of another class, since they lie deep in the decision region of another class. This phenomenon is particularly true if one class overwhelmingly dominates other classes, thus occupying the bulk of the decision region. In this case, the attack will not be



(a) Conf MI **(b)** Loss MI **(c)** Shadow MI **(d)** Local WB MI **(e)** Global WB MI

Fig. 4. Increasing AUC of various MI adversaries with increasing Hamming distance of synthetic non-members from the training dataset on target models, with a separation of class labels depending on the size of the Decision Region (DR), for the Purchase-20 dataset.

able to distinguish between members and non-members from the dominating class.

This is most evident from the results on the 2-Purchase dataset (Fig. 8a-e in Appendix D), in which one of the two classes overwhelmingly dominates the other class (a DR of almost 1). The AUC performance of the dominant class is poor, whereas it is high for the other class, bringing the average AUC close to 0.5. This partly explains why the reported performance of MI attacks on 2-Purchase has always been comparatively poorer in the literature [20, 22]. The per-class analysis on the remaining binary datasets is in Appendix D.

Observation 3. *If a class overwhelmingly dominates other classes, i.e., occupies a significant portion of the decision region in the feature space, then it is least susceptible to MI and SMI. An MI or SMI attack is unable to efficiently distinguish between members and non-members from this class.*

Tuning Attack Models for SMI. It may be argued that these MI attacks are not specifically trained to distinguish between members and nearby (synthetic) non-members, which may explain their poor performance in terms of SMI. We performed additional experiments where we tuned the training process of these attack models to further include nearby synthetic non-members. We observe even with tuning, the attack model is unable to achieve SMI. Details appear in Appendix E.

4.2 Generalization to Other Machine Learning Models

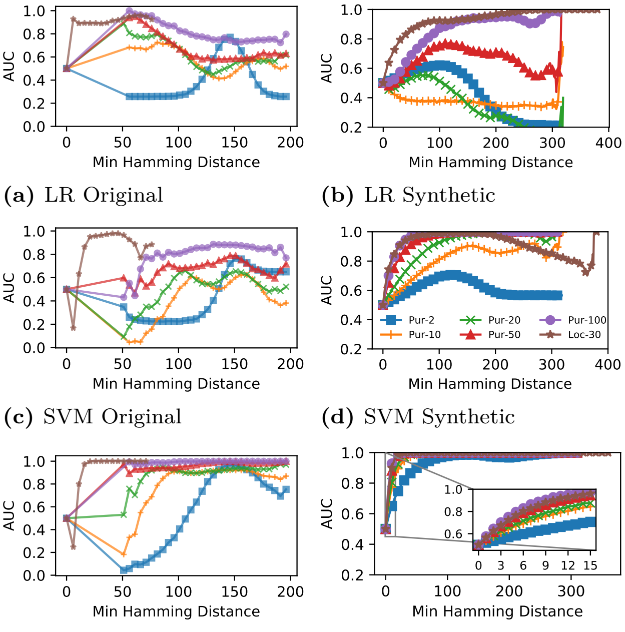
In this section, we demonstrate that the previous observations are not just limited to neural networks, and generalize to other machine learning models as well. More specifically, we use Logistic Regression (LR), Support Vector Machines (SVM) and Random Forests (RF) classifiers as the target classification models. Since our observations are consistent across all MI attacks, we only evaluate the Conf MI attack as it requires the least amount of information about the target model, making it the most portable attack between different machine learning target models.

Results. Figs. 5a, 5c, 5e display the AUCs on the original non-members from the datasets. We see that, in general, they exhibit the same as the neural network: the AUC improves as the distance of non-members from the dataset increases, with the AUC performance closer to 0.5 near the dataset. This trend in the AUCs is more prominent on the synthetic non-members shown in Figs 5b, 5d, 5f. An interesting observation is that the AUC of the RF model is very high even for non-member vectors close to the dataset, across all datasets. The main reason for this is that the RF model in general is more overfitted than the other models (see Table 4 of Appendix A). This may seem to suggest that it is possible to launch a successful SMI attack on an RF-based target model. However, if we zoom into distances close to the training dataset, i.e., inset Fig. 5f, we see that the AUC is close to 0.5 for Hamming distance ≤ 2. Thus, it is still difficult to launch an SMI attack for small distances.

Observation 4. *The observation that an MI attack is unable to distinguish between members and nearby non-members (strong membership inference) is consistent across different machine learning target models.*

5 Attribute Inference

In this section, we first present the results of our experiments using the three attribute inference (AI) attacks described in Section 3.2.2. We show that all three AI attacks have negligible advantage in inferring the missing attributes of a target vector. On the other hand, for the same three attacks, we show that approximate attribute inference attack (AAI) advantage (Def. 10) is significant, thereby suggesting that these attacks can approximately guess the missing attributes with a probability better than a random guess. We only focus on



(a) LR Original **(b)** LR Synthetic

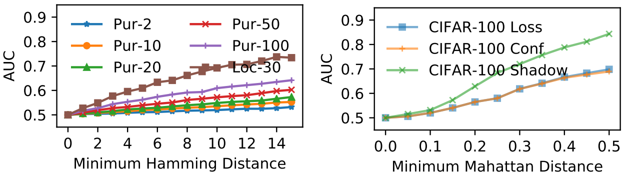
(c) SVM Original **(d)** SVM Synthetic

(e) RF Original **(f)** RF Synthetic

Fig. 5. Increasing AUC of MI with increasing Hamming distance of original and synthetic non-members from the training dataset on target models with various ML algorithms.

Table 1. Attribute Inference (Exp. 3) Advantage, where the adversary seeks to infer the exact attributes. The results below are normalized when dealing with ties.

AI	Loc-30	Pur-2	Pur-10	Pur-20	Pur-50	Pur-100	CIF-20	CIF-100
Conf	7.78E-4	1.38E-5	-3.69E-4	2.16E-4	2.00E-3	1.65E-3	-3.32E-7	4.14E-7
Loss	7.76E-4	-9.79E-5	5.57E-3	6.69E-3	4.59E-3	5.09E-3	3.33E-4	7.80E-4
Shadow	8.00E-4	-2.00E-4	2.17E-3	2.63E-3	4.10E-3	4.20E-3	2.26E-4	7.99E-4



(a) Loss MI - Location and Purchase, 15 hamming distance.

(b) CIFAR-100, zoomed to 0.5 Manhattan distance.

Fig. 6. Closer inspection of Hamming and Manhattan distance for select datasets and MI attacks previously seen in Fig. 3. Note at small distances from the training vectors, the AUC is close to 0.5, suggesting a poor AI attack.

neural networks as the target model, since we have already shown that the results generalize to other machine learning models. We also study the effect of overfitting on the success advantage of both AI and AIA attacks in the last subsection.

5.1 Attribute Inference Attacks

To perform AI experiments (Exp. 3), we train the model exactly as described in Section 3.3. We then (a) randomly select a member of the training set, or a non-member (from the testing set), (b) we mask a select number of most informative feature values as determined by mRMR [19] on the entire dataset to create the set S of unknown features (15 binary features for Location and Purchase; 5 continuous features for CIFAR datasets), (c) and generate all possible siblings of the vector under S (2 values per feature for Location and Purchase, and up to 10 values per feature for CIFAR). We then evaluate AI attacks by giving each of the generated siblings to the underlying MI attack, and flagging those siblings that the corresponding MI attack identifies as a member vector. We determine the AI attack to be successful, if the original member vector is in this set of *flagged siblings*. If there are more than one flagged sibling (excluding the original vector), we treat it as a tie and regard the attack as only partially successful. We add a fraction (determined by the number of ties) to its success count. For instance, 1/100 if there is a tie between 100 candidates. We then compute the AI advantage as the difference in the success counts between members, and non-members divided by the total counts of the tested members and non-members, respectively.

Results. Across all attacks, we observe negligible AI advantages irrespective of the dataset and the attack (see Table 1). Moreover, the advantages are also very low for more overfitted target models (Location-30, Purchase-50, Purchase-100). This suggests that an AI attack is difficult to launch, even though the same target model and datasets are susceptible to MI attacks. Our conclusion runs counter to the results from Yeom et al. on the success of attribute inference [28], who demonstrate that on regression problems, a Loss AI attack can successfully infer attributes (using Loss MI attack as a subroutine), and the more overfit the target model, the more successful the attack. But this is easily reconciled by noting that our results apply to the classification problem, where the true label given to the attacker is discrete (class label). This is in contrast to the regression problem, where the true label (response) is a continuous value. The latter provides more information to the attack algorithm, which can be employed to launch a loss-based attack, i.e., Loss AI. The link to overfitting merits further exploration, and we defer this to Section 5.3.

Table 2. Approximate AI Advantage (Def. 10), where the adversary seeks to infer approximate attributes ($\alpha = 7.5$ for Location and Purchase, $\alpha = 3.33$ for CIFAR). The results below are normalized when dealing with ties.

AAI	Loc-30	Pur-2	Pur-10	Pur-20	Pur-50	Pur-100	CIF-20	CIF-100
Conf	0.1609	0.0366	0.0516	0.0502	0.0958	0.1307	-0.0004	0.0016
Loss	0.1030	0.0125	0.0516	0.0541	0.0789	0.1012	0.0300	0.0325
Shadow	0.0554	0.0054	0.0067	0.0149	0.0766	0.0964	0.0339	0.0445

A closer look at the Location dataset sheds more light on the reasons behind the failure of the AI attack. Previously, in Section 4.1.2, we observed that the performance of the Loss MI attack on the Location dataset reaches AUC greater than ≥ 0.7, significantly higher than other datasets. In Fig. 6a we focus on the Loss MI attack on non-members at Hamming distances 1 to 15 from the dataset. We can see that the AUC reaches 0.7 at Hamming distance 10 but remains close to 0.5 between distance 1 to 3. Thus, while the Loss MI attack should easily be able to discard siblings of the original vector at Hamming distances greater than 10, it fails at closer distances and thereby resulting in an overall negligible advantage for the corresponding AI attack. The same reasoning applies to the CIFAR-100 dataset (Fig. 6b), although under Manhattan distance.

Observation 5. *It is difficult to infer (exact) attributes of a target vector in the training dataset from a machine learning model trained for a classification task, even if it is susceptible to membership inference.*

5.2 Approximate Attribute Inference Attacks

Since an MI attack starts performing better as the distance of non-member vectors from the dataset increases, this suggests that the relaxed notion of approximate attribute inference (AAI) defined in Exp. 4 may be realizable in practice. Recall that an AAI adversary is given a portion $\mathbf{x}^*$ of a vector $\mathbf{x}$, and is asked to return a vector $\mathbf{x}'$ such that $d(\mathbf{x}, \mathbf{x}') \leq \alpha$, where the parameter α determines closeness to the exact attributes. In this section, we evaluate AAI attacks. These are essentially AI attacks, but the success is determined by the parameter α. We shall set α equivalent to the expected distance of a randomly guessed vector $\mathbf{x}'$ from $\mathbf{x}$. This means, for the Location and Purchase datasets, where we have 15 unknown features, we set $\alpha = 7.5$, and for the CIFAR dataset, with 5 unknown continuous features (normalized between -1 and 1), we set $\alpha = 3.33$, which is the average distance of a random guess from the original values (See Appendix H).

Table 3. Approximate AI (Exp. 4) Advantage, where the Shadow adversary seeks to infer approximate attributes ($\alpha = 7.5$) from various states of generalized Purchase-100 Models

Dataset Size	20K	40K	60K	80K	100K	150K	200K
Overfitting	0.368	0.301	0.271	0.251	0.237	0.211	0.193
Shadow AI	0.0024	0.0046	0.0021	0.0052	0.0040	0.0049	0.0033
Shadow AAI	0.118	0.098	0.096	0.078	0.066	0.046	0.026

Results. Table 2 shows the AAI advantage (Def. 10) of the three AI attacks on all datasets. Overall, the AAI advantage is considerably higher than the AI advantage (from Table 1), reaching up to 0.1609 for the Loss AI attack on the Location dataset. However, the advantage obtained is still lower than the theoretical maximum of 1. Furthermore, the advantage is higher for more overfitted datasets, i.e., Location, Purchase-50, Purchase-100, and CIFAR-100. This indicates that increasingly the level of overfitting may improve the attack accuracy, which we shall explore in the next section. Interestingly, Shadow AI either performs worse or comparable to Conf AI and Loss AI, even though the latter attacks have less information available to them.

Observation 6. *It is possible to infer attributes approximately close to their true values with a success rate significantly greater than random guess when the target model is susceptible to membership inference.*

5.3 AI, AAI and Relation to Overfitting

In both AI and AAI attacks, we observed greater advantage on more overfitted target models. To explore this further, we focus on the Purchase-100 dataset and the Shadow AI attack. We define the overfitting level of a model as the generalization error (GE) as defined in Eq. 1. To alter GE, and hence the degree of overfitting, we vary the amount of training data, while maintaining proportional splits between training and testing sets. As we increase the training data size from 20,000 (20K) to 200,000 (200K), the generalization error decreases from 0.368 down to 0.193 as shown in Table 3.

Results. From the "Shadow AI" row of Table 3, we can see that increasing the overfitting level has little to no impact on the AI advantage (the Shadow AI result in Table 1 corresponds to a dataset size of 40K). Returning to the comparison with the findings of Yeom et al. on the effectiveness of AI on regression tasks in Section 5.1, our results indicate that for a classification problem, AI remains ineffective even if we increase the degree of overfit. On the other hand, there is a positive correlation between overfitting level and the AAI advantage, evident from the row labeled "Shadow AAI" in Table 3. As the overfitting level increases from 0.193 up to 0.368, the AAI advantage improves from 0.026 to 0.118.

Observation 7. *The more overfitted a target classification model, the more susceptible it is to approximate attribute inference. On the other hand, attribute inference remains hard even with increased overfitting levels.*

6 Related Work

The three black-box MI attacks evaluated in this paper were proposed by Shokri et al. [22], Salem et al. [20] and Yeom et al. [28]. All three works have used a split of a real dataset into training and testing sets, and demonstrated the effectiveness of MI using the testing sets. We have shown that most vectors in the testing set, i.e., non-members, are expected to be far from the training set, which explains why the relationship of MI performance to distance from members was not identified in these works. We have also shown that our results apply in the white-box setting, by evaluating the MI attacks from Nasr et al. [17], who proposed passive and active white box attacks targeting both standalone and federated models. Of course, the research on MI is not limited to these works. For instance, in [9] black and white box MI attacks are evaluated on generative adversarial networks; in [11] a new MI attack is proposed based on the loss-based MI attack from Yeom et al. evaluated in our paper, and in [25] the authors show that even if MI attacks are ineffective as a whole on a dataset, they have disparate effectiveness on different sub-groups in the dataset. We have already demonstrated that our observations generalize to other MI attacks and models, since the underlying principle remains the same, i.e., ML models are less susceptible to strong membership inference in the classification setting.

The central theme of our paper is on the feasibility of attribute inference, also known as model inversion [5, 6, 27, 31]. A criticism of these works on model inversion is that they essentially exploit the correlation between the attributes and the true label, to infer the missing attributes [22]. Finding such correlations is the very purpose of the learning task, and therefore, the missing attributes would be learned regardless of whether the challenge vector is a member or a non-member [22]. The model inversion or attribute inference definition from Yeom et al. [28] avoids this issue by defining the AI advantage as the difference between inferring attributes with the model and without the model (i.e., through the distribution). Indeed, our definitions of AI and AAI use the same approach, based on their work. Yeom et al. [28] are also the first to formally relate MI attacks to AI attacks. They also formalise the role of overfitting to the effectiveness of MI and AI attacks, a link which was previously experimentally identified and demonstrated in [20, 22]. As mentioned previously, they demonstrate that AI attacks are feasible on regression problems, with the accuracy of the attacks improving with the level of overfit. We have shown that for classification problems, only approximate attribute inference seems to be feasible.

Other attacks on machine learning models, such as *model extraction* [24], apply to the entire model itself and not necessarily to individuals in the training dataset. In a model extraction attack, unknown parameters of the model are retrieved to construct similarly behaving models (hence stealing the model in a proprietary sense). On the defense side, it has been demonstrated that MI and AI attacks can be mitigated by the use of *differential privacy* [1, 3, 10], although, this comes at a potential loss in utility [4, 10, 31]. Our findings on the infeasibility of AI attacks indicate that we may only need protection against (the weaker) approximate attribute inference, for which tailored differentially private learning algorithms can be constructed offering better utility. This is particularly useful for applications where membership inference is less of a concern, or may even be desirable. A case in point being machine learning auditors, based on membership inference attacks, to prevent unauthorised use of personal data [15, 23]. Additionally only evaluating defenses against AI may mask potential privacy leakage though AIA, an arguably simpler attack and thus a more difficult task to defend.

7 Conclusion

Our results show that it is infeasible for an attacker to correctly infer missing attributes of a target individual whose data is used to train a machine learning model for a classification problem owing to the inability of membership inference attacks to distinguish between members and nearby non-members. For applications, where the privacy concern is attribute inference, and not membership inference, defense mechanisms tailored to protect against approximate attribute inference can be constructed. As a future direction, it will be interesting to explore whether the approximate attribute inference attacks mentioned in this paper can be improved to infer missing attributes as close as possible to the original attributes.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Acquire valued shoppers challenge - kaggle. https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data, 2014. Accessed: 2019-06-30.

[3] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[4] Farhad Farokhi and Mohamed Ali Kaafar. Modelling and quantifying membership information leakage in machine learning. *arXiv preprint arXiv:2001.10648*, 2020.

[5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[6] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

[7] Philippe Gaborit and Gilles Zemor. Asymptotic improvement of the gilbert–varshamov bound for linear codes. *IEEE Transactions on Information Theory*, 54(9):3865–3872, 2008.

[8] Venkatesan Guruswami. Gilbert-varshamov bound. Lecture Notes, Introduction to Coding Theory, 2010.

[9] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

[10] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX*, 2019.

[11] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[12] Michael J Kearns and Umesh V Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[13] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[14] Bogdan Kulynych and Mohammad Yaghini. mia: A library for running membership inference attacks against ML models, September 2018.

[15] Yuantian Miao, Ben Zi Hao Zhao, Minhui Xue, Chao Chen, Lei Pan, Jun Zhang, Dali Kaafar, and Yang Xiang. The audio auditor: Participant-level membership inference in voice-based iot. *arXiv preprint arXiv:1905.07082*, 2019.

[16] D. S. Mitrinović, J. E. Pečarić, and A. M. Fink. *Bernoulli's Inequality*, pages 65–81. Springer Netherlands, 1993.

[17] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. *arXiv preprint arXiv:1812.00910*, 2018.

[18] Mícheál O'Searcoid. *Metric spaces*. Springer Science & Business Media, 2006.

[19] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2005.

[20] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *NDSS*, 2019.

[21] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[22] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[23] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[24] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, pages 601–618, 2016.

[25] Mohammad Yaghini, Bogdan Kulynych, and Carmela Troncoso. Disparate vulnerability: on the unfairness of privacy attacks against machine learning. *arXiv preprint arXiv:1906.00389*, 2019.

[26] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):30, 2016.

[27] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 225–240, 2019.

[28] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[29] Benjamin Zi Hao Zhao, Hassan Jameel Asghar, Raghav Bhaskar, and Mohamed Ali Kaafar. On inferring training data attributes in machine learning models. *arXiv preprint arXiv:1908.10558*, 2019.

[30] Benjamin Zi Hao Zhao, Hassan Jameel Asghar, and Mohamed Ali Kaafar. On the resilience of biometric authentication systems against random inputs. In *Network and Distributed System Security Symposium (NDSS)*, 2020.

[31] Han Zhao, Jianfeng Chi, Yuan Tian, and Geoffrey J Gordon. Adversarial privacy preservation under attribute inference attack. *arXiv preprint arXiv:1906.07902*, 2019.

A Model Parameters

In this Appendix we describe the training split and the model hyper-parameters used to train the target model for each dataset.

Table 4. Summary of training and testing accuracies, with MI AUC for all machine learning classifiers.

Dataset	Model	Train Acc	Test Acc	MI AUC	Model - MI	Train Acc	Test Acc	MI AUC
Loc-30	LR - Conf	1.000	0.582	0.897	NN - Conf	1.000	0.794	0.705
	SVM - Conf	1.000	0.731	0.916	NN - Loss	1.000	0.794	0.710
	RF - Conf	1.000	0.566	0.975	NN - Shadow	1.000	0.666	0.909
	NN - Local	0.998	0.430	0.891	NN - Global	0.998	0.430	0.886
Pur-100	LR - Conf	1.000	0.484	0.765	NN - Conf	0.999	0.765	0.708
	SVM - Conf	1.000	0.799	0.855	NN - Loss	0.999	0.765	0.720
	RF - Conf	1.000	0.606	0.998	NN - Shadow	1.000	0.700	0.842
	NN - Local	0.538	0.487	0.508	NN - Global	0.538	0.487	0.719
Pur-50	LR - Conf	0.995	0.601	0.614	NN - Conf	0.998	0.832	0.629
	SVM - Conf	1.000	0.857	0.716	NN - Loss	0.998	0.832	0.638
	RF - Conf	1.000	0.724	0.980	NN - Shadow	1.000	0.778	0.763
	NN - Local	0.692	0.657	0.520	NN - Global	0.692	0.657	0.668
Pur-20	LR - Conf	0.973	0.785	0.552	NN - Conf	0.999	0.889	0.577
	SVM - Conf	1.000	0.906	0.584	NN - Loss	0.999	0.889	0.582
	RF - Conf	1.000	0.813	0.917	NN - Shadow	1.000	0.841	0.690
	NN - Local	0.803	0.781	0.505	NN - Global	0.803	0.781	0.626
Pur-10	LR - Conf	0.973	0.878	0.521	NN - Conf	0.999	0.911	0.558
	SVM - Conf	1.000	0.932	0.530	NN - Loss	0.999	0.911	0.561
	RF - Conf	1.000	0.840	0.902	NN - Shadow	1.000	0.868	0.644
	NN - Local	0.836	0.818	0.503	NN - Global	0.836	0.818	0.608
Pur-2	LR - Conf	1.000	0.986	0.499	NN - Conf	0.998	0.959	0.521
	SVM - Conf	1.000	0.987	0.502	NN - Loss	0.998	0.959	0.522
	RF - Conf	1.000	0.921	0.781	NN - Shadow	0.999	0.944	0.580
	NN - Local	0.914	0.906	0.505	NN - Global	0.914	0.906	0.567
CIFAR-20	NN - Conf	0.920	0.322	0.544	NN - Loss	0.920	0.322	0.799
	NN - Shadow	0.999	0.281	0.925	-	-	-	-
CIFAR-100	NN - Conf	0.831	0.214	0.524	NN - Loss	0.831	0.214	0.844
	NN - Shadow	0.999	0.170	0.967	-	-	-	-

A.1 Target Models

We will first describe the neural network based target models used in the bulk of our experiments, followed by the classifier configurations of the classifiers in Section 4.2. The training and testing accuracies can be found in Table 4.

A.1.1 Location

The model was trained in keras as a fully connected neural network with 1 hidden layer of 128 nodes with the "tanh" activation function. This architecture replicates the training and testing accuracy for the target model as previously reported in [22].

A.1.2 Purchase

The target model was trained in keras as a fully connected neural network with 1 hidden layer of [128] nodes with a "tanh" activation function. This architecture replicates the training and testing accuracy for the target model as previously reported in [22].

A.1.3 CIFAR

The target model is a multilayer perceptron, consisting of two hidden layers of 256 units, with relu activation layer and a softmax output layer. This is the same architecture used in [10].

A.2 Other Machine Learning Classifier Configuration

Logistic Regression (LR): The parameter C was set at 100 for all datasets, with all other parameters remain at the default values. **Support Vector Machine (SVM)**: We select a linear kernel for all the datasets. We keep parameters at default values. **Random Forest (RF)**: The number of estimators was chosen to be 100 with no depth specified, the remaining parameters were kept as defaults.

The training and testing accuracies for each algorithm, and for each datasets are noted in Table 4.

A.3 MI Attack Configurations

Due to the different data requirements for each attack, the way the data is partitioned differs, we note these differences in this section. The average MI AUC can be found in Table 4. For the Conf and Loss attacks, we do not require additional data to train an attack model, as such:

A.3.1 Conf and Loss attacks

Location We take the full dataset and divide it into 2 parts. 20% is used for training the target model and remainder 80% is kept for testing purposes. **Purchase** We sample 20,000 records from the dataset and divide it into 2 parts. The first 80% is used for training the target model and remaining 20% is kept for testing purposes. **CIFAR** 50,000 records are sampled from the dataset to constitute our experimental dataset, from this 20% is reserves as the training data, and the remaining 80% is use for testing.

A.3.2 Shadow MI

Location: We take the full dataset and divide it into 3 parts. The first 20% is used for training the target model, 64% for training the shadow models and the remaining 16% is retained for testing. Our Shadow MI attack is from the open-source library [14]. The training and testing accuracies are found in Table 4. Our models are as follows:

1. **Shadow Models:** We select 60 attack models for Location dataset, consistent with [22]. The architecture of these shadow models and the size of their training dataset are equivalent to the target model.
2. **Attack Model:** The attack model is multilayer perceptron with a 64-unit hidden layer and a sigmoid output layer. This architecture replicates the precision and recall as previously reported in [22]. For the Location-30 dataset our MI attack obtains a precision of 0.93 and recall of 0.82

Purchase: We sample 40000 records from the dataset and divide it into 3 parts. The first 25% is used for training the target model, 67.5% for training the shadow models and the last 7.5% is kept for testing. The setup for running this attack on the Purchase datasets are as follows:

1. **Shadow Models** We chose the number of shadow models as 20 for Purchase dataset. The architecture of these shadow models and the size of their training dataset are the same as the target model.
2. **Attack Model** The attack model is multilayer perceptron with a 64-unit hidden layer and a sigmoid output layer. This architecture replicates the precision and recall observed in [22]. We obtain precision of 0.66, 0.78, 0.81, 0.85, 0.89 and recalls of 0.54, 0.57, 0.6, 0.67, 0.76 for Purchase-2, 10, 20, 50, 100, respectively.

CIFAR We sample complete dataset(around 50000 records) from the dataset and divide it into 3 parts. The first 20% is used for training the target model,next 72% for training the shadow model and the rest 8% is kept for testing purposes. The setup for running this attack on this dataset is as follows:

1. **Shadow Models** We chose the number of attack models as 5 for CIFAR dataset which is the same as [10]. The architecture of this shadow model and the size of the training dataset is the same as the target model.
2. **Attack Model** It is a multilayer perceptron(two 64 unit hidden layer with “tanh” activation layer and a sigmoid output layer).This architecture matches the precision and recall of the attack model as previously reported in the [22]. We achieve 0.98 precision and 0.9 recall for CIFAR-100.

B Nasr et al. (Local and Global) White Box Inference Attacks

As a result of the federated setting, the target models for our datasets differ. The target models and attack model architecture, as well as the training and testing setup, originally described by [17] are utilised in this study.

B.1 Target Model

Our target model for both datasets consisted of five layers (1024, 512, 256, 128, 100) with “tanh” activation, replicated from [17]. Each party as well as the server is trained on this model across 100 epochs with an Adam optimizer with learning rate of 0.0001 and cross entropy loss.

B.2 Attack Model

The attack model takes in a number of different inputs from the target model, which are trained on ’submodules’ before being combined in a final network. These inputs described below, with c being equal to the number of classes of the dataset:

- The gradient of the loss of the final layer - One convolutional layer (1000) with kernel size (1,c) and three hidden layers (1024,512, 128)
- The one hot encoded true label - Two hidden layers (128, 64)
- The predicted labels - Two hidden layers (100, 64)
- The output for the correct label – Two hidden layers (c, 64)

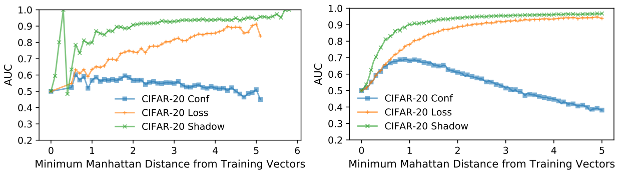
The combined input is trained using three hidden layers (256, 126, 64, 1). "ReLu" activation is used throughout the attack model, with an Adam optimiser with learning rate of 0.00001 and mean square error loss.

B.3 Datasets

During target model training the Location and Purchase datasets were both split with 20% (30,000 for Purchase, 1,158 for Location) used for the initial target model training, and 80% (150,000 for Purchase, 5,790 for Location) for testing, as described for the purchase dataset in [17]). The data was further split equally amongst the three parties so that each party had a training and testing set of the same size. The attack model was subsequently trained with half of the original training data and the same amount of the original testing data (representing members and nonmembers, respectively). Each batch was designed to have 50% of members and nonmembers. The remaining samples were used for testing.

C CIFAR-20 Plots

In Section 4, we presented results for CIFAR-100, here we provide accompanying plots in Figure 7a and 7b for CIFAR-20, which demonstrates the same trends as those observed in CIFAR-100. We do note that the AUC curves for CIFAR-20 are slightly lower than the respective CIFAR-100 curves. An expected result due to the reduction in the number of class labels.



(a) Original vectors **(b)** Generated vectors

Fig. 7. AUC of MI attacks on original and synthetic non-member vectors of the CIFAR-20 dataset as a function of Manhattan distance.

D Per-Label Plots

As previously discussed in Section 4.1.3, we had only highlighted the Purchase-20 dataset. We provide in this appendix the per-label plots of our remaining binary datasets. They can be found in Fig. 8.

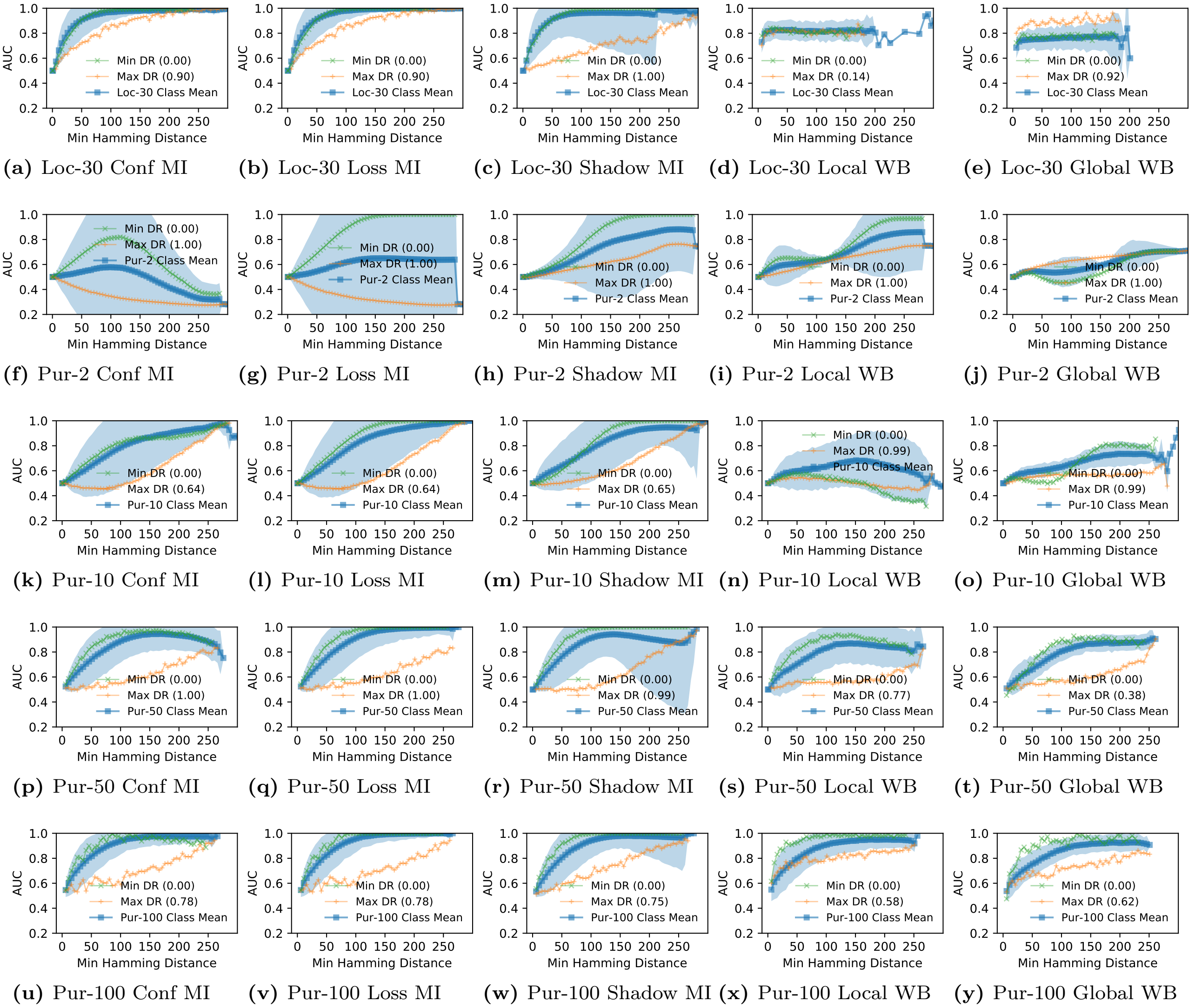
E Tuning Attack Models for SMI

It may be argued that these MI attacks are not specifically trained to distinguish between members and nearby (synthetic) non-members, which may explain their poor performance in terms of SMI. To investigate if we can improve their performance as SMI, we tune the training process of these attack models to further include nearby synthetic non-members. This augmented training process is only applicable to the MI attacks that employ an attack model, i.e., Shadow, Local WB, and Global WB. The other two MI attacks, i.e., Conf and Loss MI, directly inspect the outputs of the target model for their MI decision, and hence fine tuning the decision based on member and nearby synthetic non-member vectors is not applicable to these models.

To perform this experiment we take the same experimental steps as that of Section 4.1.2, and select the Shadow MI attack, and augment the tuning step with synthetic non-members generated from both members and non-members of the attack model training set. For each training vector (member or non-member), we generate two synthetic vectors at all Hamming distances up to 10. These synthetic non-members are then used to update the attack model.

From Fig. 9, it can be observed that the AUC of the attack at distances close to the dataset still remains close to 0.5, while at larger distances, the AUC approaches 0, indicating that the attack can distinguish between members and non-members as we move away from the dataset, although with membership label reversed, i.e., more members are now classified as nonmembers and vice versa. Upon closer inspection, the attack model had no advantage in inferring membership of member vectors (near 0.5 AUC across all datasets). On the other hand, the attack model erred more towards mislabeling non-members (both original and synthetic) as members. We hypothesize this output label 'flipping' of the trend is due to the numerous additional close non-members provided to the attack model, which "confuses" the model in distinguishing members from non-members, producing an AUC below 0.5. Regardless, for all datasets tuning the attack model for SMI does not show any improvement in detecting non-members close to the dataset compared to the original attack model. We also carried out an additional repetition of the experiment with one synthetic vector generated per member and non-member, at each Hamming distance up to 50. This demonstrated worse AUC performance over all distances.

We conclude that despite the retraining the attack model with additional nearby non-members, the attack failed to achieve SMI. In fact, MI performance generally decreased, due to the similarity of members and the synthetic nearby non-members.



(a) Loc-30 Conf MI **(b)** Loc-30 Loss MI **(c)** Loc-30 Shadow MI **(d)** Loc-30 Local WB **(e)** Loc-30 Global WB

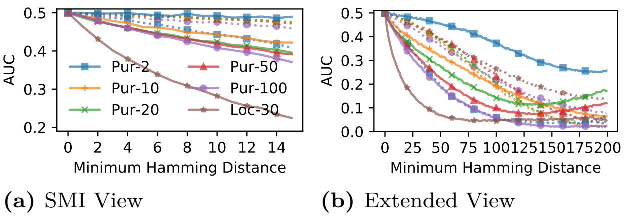
(f) Pur-2 Conf MI **(g)** Pur-2 Loss MI **(h)** Pur-2 Shadow MI **(i)** Pur-2 Local WB **(j)** Pur-2 Global WB

(k) Pur-10 Conf MI **(l)** Pur-10 Loss MI **(m)** Pur-10 Shadow MI **(n)** Pur-10 Local WB **(o)** Pur-10 Global WB

(p) Pur-50 Conf MI **(q)** Pur-50 Loss MI **(r)** Pur-50 Shadow MI **(s)** Pur-50 Local WB **(t)** Pur-50 Global WB

(u) Pur-100 Conf MI **(v)** Pur-100 Loss MI **(w)** Pur-100 Shadow MI **(x)** Pur-100 Local WB **(y)** Pur-100 Global WB

Fig. 8. Increasing AUC of MI attacks with increasing distance of synthetic non-members from the training dataset, with a separation of class labels depending on the size of the DR, for the Location-30, Purchase-2, Purchase-10, Purchase-50, Purchase-100 datasets.



(a) SMI View **(b)** Extended View

Fig. 9. AUC performance on Shadow MI tuned with additional close vectors (dotted lines). The existing Shadow MI results (solid lines) have been mirrored on 0.5 to allow for easier comparison before and after tuning.

F Metrics, Balls and Siblings

Theorem 3 (Metrics)**.** *Let d_1 be a metric on $\mathbb{D}$. Let $\mathbf{x}, \mathbf{x}' \in \mathbb{D}^m$. Then the functions*

1. $d_M(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m} d_1(x_i, x_i')$,
2. $d_E(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{m} (d_1(x_i, x_i'))^2}$,
3. $d_\infty(\mathbf{x}, \mathbf{x}') = \max_{i \in [m]}(d_1(x_i, x_i'))$,

are metrics on the product space $\mathbb{D}^m$. Moreover, for every $\mathbf{x}, \mathbf{x}' \in \mathbb{D}^m$, we have $d_\infty(\mathbf{x}, \mathbf{x}') \leq d_E(\mathbf{x}, \mathbf{x}') \leq d_M(\mathbf{x}, \mathbf{x}')$ [18, §1.6].

Definition 11 (Conserving metric)**.** A metric d is called a conserving metric [18, §1.6] on the product space $\mathbb{D}^m$ if for all $\mathbf{x}, \mathbf{x}' \in \mathbb{D}^m$, we have

$$d_\infty(\mathbf{x}, \mathbf{x}') \leq d(\mathbf{x}, \mathbf{x}') \leq d_M(\mathbf{x}, \mathbf{x}').$$

□

Examples of conserving metrics include the Hamming distance over $\mathbb{D}^m = \{0,1\}^m$, where $d_1(x, x') = |x - x'|$, $x, x' \in \{0,1\}$, the Euclidean distance over $\mathbb{D}^m = [0,1]^m$, where $d_1(x, x') = |x - x'|$, $x, x' \in [0,1]$, and the Manhattan distance (d_M) over $\mathbb{D}^m = [-1,1]^m$, where $d_1(x, x') = |x - x'|$, $x, x' \in [-1,1]$. Henceforth we will assume the metric d to be a conserving metric on $\mathbb{D}^m$.

For any subset $X \subseteq \mathbb{D}^m$, the diameter of X, denoted $\mathsf{diam}_d(X)$ is defined as $\max\{d(\mathbf{x}, \mathbf{x}') \mid \mathbf{x}, \mathbf{x}' \in X\}$.

Bounded Feature Space. We assume $\mathbb{D}$ to be bounded, i.e., $\mathsf{diam}_{d_1}(\mathbb{D}) < \infty$. Since d is a conserving metric it follows that $\mathsf{diam}_d(\mathbb{D}^m) < \infty$, and hence the feature space is also bounded. This is equivalent to saying that for any $\mathbf{x} \in \mathbb{D}^m$, there exists an $R > 0$ such that $\mathbb{D}^m = B_d(\mathbf{x}, R)$ [18, §7.1].

Siblings. Overloading notation, we also define

$$\Phi_i(\mathbf{x}) = \bigcup_{\substack{S \subseteq [m] \\ |S| = i}} \Phi_S(\mathbf{x}),$$

where $1 \leq i \leq m - 1$.

Proposition 1. *Let $1 \leq i \leq m - 1$. Let $r \geq i \times \mathsf{diam}_{d_1}(\mathbb{D})$. Then for every feature vector $\mathbf{x} \in \mathbb{D}^m$, we have $\Phi_i(\mathbf{x}) \subseteq B_d(\mathbf{x}, r)$.*

Proof. Consider any $\mathbf{x}' \in \Phi_i(\mathbf{x})$. Then $\mathbf{x}' \in \Phi_S(\mathbf{x})$, for some $S \subseteq [m]$ where $|S| = i$. Then, as d is a conserving metric,

$$\begin{aligned} d(\mathbf{x}, \mathbf{x}') \leq d_M(\mathbf{x}, \mathbf{x}') &\leq \sum_{j=1}^{m} d_1(x_j, x'_j) = \sum_{j \in S} d_1(x_j, x'_j) \\ &\leq \sum_{j \in S} \mathsf{diam}_{d_1}(\mathbb{D}) = i \times \mathsf{diam}_{d_1}(\mathbb{D}) \leq r. \end{aligned}$$

Hence $\mathbf{x}' \in B(\mathbf{x}, r)$. □

For metrics d_E and d_M, we define d_i to be the restriction of d_E or d_M to i dimensions in a natural way, where $1 \leq i \leq m$.

Proposition 2. *If $\mathsf{diam}_{d_1}(\mathbb{D}) = \delta > 0$, then $\mathsf{diam}_{d_1}(\mathbb{D}) < \mathsf{diam}_{d_2}(\mathbb{D}^2) < \mathsf{diam}_{d_3}(\mathbb{D}^3) < \cdots$.*

Proof. Consider the metric to be d_E. Consider $i = 1$. Then there exist $x, x' \in \mathbb{D}$ such that $\delta = d(x, x')$. Construct the 2-dimensional vectors $\mathbf{x} = (x, x)$ and $\mathbf{x}' = (x', x')$. Then,

$$\begin{aligned} \mathsf{diam}_{d_2}(\mathbb{D}^2) &\geq \sqrt{(d_1(x, x'))^2 + (d_1(x, x'))^2} \\ &= \sqrt{2}\delta > \delta = \mathsf{diam}_{d_1}(\mathbb{D}). \end{aligned}$$

The rest of the proof follows by induction. The case for d_M is similar. □

Proposition 3. *Let $1 \leq i \leq m - 1$. Let $\mathsf{diam}_{d_{i+1}}(\mathbb{D}^{i+1}) > r \geq \mathsf{diam}_{d_i}(\mathbb{D}^i)$, where d_j is d_E restricted to j dimensions. Then,*

1. *For any feature vector $\mathbf{x} \in \mathbb{D}^m$, we have $\Phi_i(\mathbf{x}) \subseteq B_{d_E}(\mathbf{x}, r)$.*
2. *There exists a feature vector $\mathbf{x} \in \mathbb{D}^m$, such that $\Phi_{i+1}(\mathbf{x}) \not\subseteq B_{d_E}(\mathbf{x}, r)$.*

Furthermore, the same holds for the metric d_M, and d_j being d_M restricted to j dimensions.

Proof. For part (1), consider any $\mathbf{x}' \in \Phi_i(\mathbf{x})$. Then $\mathbf{x}' \in \Phi_S(\mathbf{x})$, for some $S \subseteq [m]$ where $|S| = i$. Then,

$$\begin{aligned} d_E(\mathbf{x}, \mathbf{x}') &= \sqrt{\sum_{j=1}^{m} (d_1(x_j, x'_j))^2} \\ &= \sqrt{\sum_{j \in S} (d_1(x_j, x'_j))^2} \leq \mathsf{diam}_{d_i}(\mathbb{D}^i) \leq r. \end{aligned}$$

Hence $\mathbf{x}' \in B_{d_E}(\mathbf{x}, r)$. For part (2), let $\delta = \mathsf{diam}_{d_{i+1}}(\mathbb{D}^{i+1})$. Then their exist $(i+1)$-dimensional vectors $\mathbf{x}', \mathbf{x}'' \in \mathbb{D}^{i+1}$ such that $d_{i+1}(\mathbf{x}', \mathbf{x}'') = \delta$. Furthermore, $d_1(x'_j, x''_j) \neq 0$, for all $j \in [i+1]$. Suppose not, and wlog assume that $d_1(x'_{i+1}, x''_{i+1}) = 0$. Then, we can discard the last element from both vectors, and the resulting i-dimensional vectors have distance δ according to d_i, which is greater than $\mathsf{diam}_{d_i}(\mathbb{D}^i)$; a contradiction. Now, sample any $(m - i - 1)$-dimensional vector from $\mathbb{D}^{m-i-1}$ and append it to both $\mathbf{x}'$ and $\mathbf{x}''$. Let us call the resulting vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. Let $S = \{1, 2, \ldots, i+1\}$. Then, $|S| = i + 1$, and $\mathbf{x}_2 \in \Phi_S(\mathbf{x}_1) \subseteq \Phi_{i+1}(\mathbf{x}_1)$, but

$$\begin{aligned} d_E(\mathbf{x}_1, \mathbf{x}_2) &= \sqrt{\sum_{j=1}^{m} (d_1(x_j, x'_j))^2} \\ &= \sqrt{\sum_{j \in S} (d_1(x_j, x'_j))^2} = \delta > r. \end{aligned}$$

Hence $\mathbf{x}_2 \notin B_{d_E}(\mathbf{x}_1, r)$.

A similar proof holds for the metric d_M. □

Corollary 1. *Let i and $\mathbf{x}$ be as in the statement of the previous proposition. Define $d_1(x, x') = |x - x'|$ for $x, x' \in \mathbb{D}$.*

1. *Let d_H be the Hamming distance on $\mathbb{D} = \{0,1\}^m$. Let $r \geq i$. Then $\Phi_i(\mathbf{x}) \subseteq B_{d_H}(\mathbf{x}, r)$.*
2. *Let d_M be the Manhattan distance on $\mathbb{D} = [-1,1]^m$. Let $r \geq 2i$. Then $\Phi_i(\mathbf{x}) \subseteq B_{d_M}(\mathbf{x}, r)$.*
3. *Let d_E be the Euclidean distance on $\mathbb{D} = [-1,1]^m$. Let $r \geq \sqrt{4i}$. Then $\Phi_i(\mathbf{x}) \subseteq B_{d_E}(\mathbf{x}, r)$.*

G Relationship between Inference Notions

Proof of Theorem 1.

Proof. We essentially show that a membership inference (MI) adversary does not imply a strong membership inference (SMI) adversary, i.e., MI $\not\Rightarrow$ SMI. Let $r > 0$ be fixed. Let $k \geq 2$ be a fixed number of labels. Let $N \gg n$. Sample N points from $\mathbb{R}^m$ such that for all pairs of points $\mathbf{x}, \mathbf{x}'$ in this sample, with $\mathbf{x} \neq \mathbf{x}'$, we have $d(\mathbf{x}, \mathbf{x}') > 3r$.[1] Let us call this sample S_1. For each $\mathbf{x} \in S_1$, assign it an arbitrary label from the k labels and set $c(\mathbf{x})$ to this label. Initialize an empty set S_2. Now for each $\mathbf{x} \in S_1$, sample a random point from $B(\mathbf{x}, r) - \{\mathbf{x}\}$, and add to S_2, and assign it the same label as $\mathbf{x}$, i.e., $c(\mathbf{x})$. Let $S = S_1 \cup S_2$. Notice that every vector in S has precisely one r-neighbour in S. To see this, first note that every vector in S_1 is not an r-neighbour of any other vector in S_1 by construction. Next, we take a vector $\mathbf{x}$ in S_1, and see if it has more than one r-neighbours in S_2. Let $\mathbf{y}$ be the r-neighbour guaranteed by construction. Assume now that $\mathbf{w} \in S_2$ different from $\mathbf{y}$ is another r-neighbour of $\mathbf{x}$. Let $\mathbf{z} \in S_1$ be the r-neighbour of $\mathbf{w}$ in S_1 guaranteed by construction. Then,

$$\begin{aligned} d(\mathbf{x}, \mathbf{z}) &\leq d(\mathbf{x}, \mathbf{w}) + d(\mathbf{w}, \mathbf{z}) \\ \Rightarrow d(\mathbf{x}, \mathbf{z}) &\leq r + r = 2r, \end{aligned}$$

a contradiction. Next, we will look at vectors in S_2. We will check if any vector from S_2 has more than one r-neighbour in S_1. Then, we will check if the vectors in S_2 have any r-neighbours in S_2. This exhausts the cases.

Let $\mathbf{y}$ be the r-neighbour in S_2 of some $\mathbf{x} \in S_1$. This is true by construction. Let $\mathbf{z}$ be some other vector in S_1. Then, $d(\mathbf{x}, \mathbf{y}) \leq r$, and $d(\mathbf{x}, \mathbf{z}) > 3r$. Therefore,

$$\begin{aligned} &d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \\ &\Rightarrow 3r < d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \\ &\Rightarrow 3r < r + d(\mathbf{y}, \mathbf{z}) \\ &\Rightarrow 2r < d(\mathbf{y}, \mathbf{z}), \end{aligned}$$

hence $\mathbf{y}$ is not an r-neighbour of any other $\mathbf{z}$ in S_1. Now consider some $\mathbf{w} \in S_2$ not equal to $\mathbf{y}$. Assume to the contrary that $d(\mathbf{y}, \mathbf{w}) \leq r$. Let $\mathbf{z}$ be the r-neighbour of $\mathbf{w}$ in S_1 (again by construction, it should exist). Then,

$$\begin{aligned} &d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{w}) + d(\mathbf{y}, \mathbf{z}) \\ \Rightarrow &d(\mathbf{x}, \mathbf{z}) \leq r + r + r = 3r, \end{aligned}$$

which is a contradiction.

Let $\mathbb{D}^m = S$. Define the distribution $\mathcal{D}$ as the uniform distribution over S. Sample a dataset $X \leftarrow \mathcal{D}^n$. Define a classifier h_X which given a point $\mathbf{x}$ in X, assigns its label $c(\mathbf{x})$ to all vectors within the ball $B(\mathbf{x}, r)$, i.e., all r-neighbours of $\mathbf{x}$ have the constant label. The classifier h_X, when queried for a point $\mathbf{x} \in X$, simply outputs the label $c(\mathbf{x})$. For any point $\mathbf{x} \notin X$, it checks if there is some $\mathbf{x}' \in X$ such that $d(\mathbf{x}', \mathbf{x}) \leq r$. If yes, it returns the label $c(\mathbf{x}')$. Otherwise, it returns an arbitrary label from the k labels.

Now consider an MI adversary $\mathcal{A}$ which given $(\mathbf{x}, c(\mathbf{x}))$, queries h_X with $\mathbf{x}$, and outputs 1 (member) if $h_X(\mathbf{x}) = c(\mathbf{x})$ and 0 (non-member) otherwise. Let us calculate the probabilities in:

$$\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0],$$

which define the adversary's advantage (Definition 7). If $\mathbf{x}$ is a member, then the adversary does not make a mistake, as the label returned by h_X is exactly the label $c(\mathbf{x})$ by construction. Therefore,

$$\Pr[b' = 1 \mid b = 1] = 1.$$

Now consider the other probability, i.e., $\Pr[b' = 1 \mid b = 0]$. The adversary could erroneously output $\mathbf{x}$ as a member either if its r-neighbour was in X, or if its r-neighbour was not part of X, but the classifier gives it the correct label by chance. Thus

$$\begin{aligned} \Pr[b' = 1 \mid b = 0] = &\left(1 - \left(\frac{2N-2}{2N-1}\right)^n\right) \\ &+ \left(\frac{2N-2}{2N-1}\right)^n \left(\frac{1}{k}\right) \end{aligned}$$

[1] There can be many such vectors, which can be found using a greedy algorithm [8]. For instance, if $\mathbb{D} = \{0,1\}$, $r = 1$, and d is the Hamming distance, then the Gilbert-Varshamov bound states that there are at least

$$\frac{2^m}{\sum_{i=0}^{3} \binom{m}{i}},$$

vectors with minimum Hamming distance $> 3r = 3$ [7, 8].

$$= 1 - \left(1 - \frac{1}{2N-1}\right)^n \left(\frac{k-1}{k}\right)$$

Subtracting this from the above, we see that the advantage is

$$\left(1 - \frac{1}{2N-1}\right)^n \left(\frac{k-1}{k}\right)$$

By Bernoulli's inequality [16], we have

$$\left(1 - \frac{1}{2N-1}\right)^n \geq 1 - \frac{n}{2N-1},$$

and noting that $N > n$, we get $2N - 1 \geq 2n$. And therefore,

$$1 - \frac{n}{2N-1} \geq 1 - \frac{n}{2n} = \frac{1}{2}.$$

Finally, we get the advantage of at least $\frac{1}{2}\frac{k-1}{k}$, which is a constant.[2] However, the same adversary if used as a subroutine in Experiment 2, will always output 1 if queried on $\mathbf{x}$ and its r-neighbour, since every r-neighbour of a member $\mathbf{x} \in X$, is assigned the true label (even if it is not in X, by construction). Hence, the resulting adversary has no advantage in the sense of SMI. □

Proof of Theorem 2.

Proof. Consider an SMI adversary $\mathcal{B}$ which is given $\mathbf{x}$. SMI chooses a random index, or alternatively, a random index set S of cardinality 1. The adversary $\mathcal{B}$ constructs $\mathbf{x}^* = \phi_S(\mathbf{x})$ and gives it to $\mathcal{A}$. Upon receiving $\mathbf{x}'$ from $\mathcal{A}$, the adversary $\mathcal{B}$ checks if $\mathbf{x}' = \mathbf{x}$. If yes, it returns 1. Else it returns 0. The advantage of adversary $\mathcal{B}$ is

$$\begin{aligned}
&\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0] \\
&= \Pr[\mathrm{Exp}_{\mathrm{AI}}(\mathcal{A}, h_X, 1, n, \mathcal{D}) = 1 \mid b = 1] \\
&- \Pr[\mathrm{Exp}^*_{\mathrm{AI}}(\mathcal{A}, h_X, 1, n, \mathcal{D}) = 1 \mid b = 0] \\
&\leq \Pr[\mathrm{Exp}_{\mathrm{AI}}(\mathcal{A}, h_X, 1, n, \mathcal{D}) = 1 \mid b = 1] \\
&- \Pr[\mathrm{Exp}_{\mathrm{AI}}(\mathcal{A}, h_X, 1, n, \mathcal{D}) = 1 \mid b = 0] + \epsilon(r) \\
&= \delta + \epsilon(r).
\end{aligned}$$

In the above, $\Pr[\mathrm{Exp}^*_{\mathrm{AI}}(\mathcal{A}, h_X, 1, n, \mathcal{D}) = 1 \mid b = 0]$ denotes the version of Experiment 4, where $\mathbf{x} \leftarrow \mathcal{D}$ in Step 3 is replaced with $\mathbf{x}_0 \leftarrow\$ X, \mathbf{x} \leftarrow B_d(\mathbf{x}_0, r)$, according to the distribution induced by $\mathcal{D}$, and $\epsilon(r)$ is the advantage of a distinguisher who distinguishes between the two distributions. Under the indistinguishable neighbour assumption 5, we assume this to be negligible for small r. □

H Miscellaneous Results

Relationship between AUC and Advantage. The MI advantage from Definition 7 denoted $\mathrm{Adv}_{\mathrm{MI}}(\mathcal{A}, h_X, n, \mathcal{D})$ can be empirically estimated as $\mathrm{TPR}(\tau) - \mathrm{FPR}(\tau)$[3] with τ denoting the threshold parameter of the given classifier h_X and $\mathrm{TPR}(\tau)$ and $\mathrm{FPR}(\tau)$ denoting the True Positive Rate and False Positive Rate respectively at τ. The AUC-ROC statistic captures the aggregate performance of the classifier h_X for all possible values of the threshold τ and is computed as $\mathrm{AUC} = \int_{\mathrm{FPR}(\tau)=0}^{1} \mathrm{TPR}(\tau) d(\mathrm{FPR}(\tau)) = \int_{x=0}^{1} \mathrm{TPR}(\mathrm{FPR}^{-1}(x))dx$.

When $\mathrm{Adv}_{\mathrm{MI}}(\mathcal{A}, h_X, n, \mathcal{D})) = \mathrm{Adv}_m$ for all possible values of τ (*i.e.* Advantage is same for all values of the threshold parameter), the AUC is computed as $\int_{x=0}^{1} (\mathrm{FPR}(\mathrm{FPR}^{-1}(x)) + \mathrm{Adv}_m)dx = \frac{1}{2} + \mathrm{Adv}_m$. Thus, $\mathrm{AUC} - \frac{1}{2}$ equals the advantage from Definition 7. Even when the advantages vary with τ, $\mathrm{AUC} - \frac{1}{2}$ is a good approximation for the average advantage.

Similarly, the Advantage in the strong membership inference definition, $\mathrm{Adv}_{\mathrm{SMI}}(\mathcal{A}, h_X, r, n, \mathcal{D})$ can be empirically estimated as $\mathrm{TPR}(\tau) - \mathrm{FPR}(\tau)$ as long as $B_d(\mathbf{x}_0, r)$ is assumed to have a small number of samples from X, i.e., in general $B_d(\mathbf{x}_0, r)$ would contain more elements outside of X.

Average Manhattan Distance. Let $\mathbb{D}^m = [-1, 1]^m$. Given a vector $\mathbf{x} \in \mathbb{D}^m$, we want to find the Manhattan distance d_M between $\mathbf{x}$ and a vector $\mathbf{y} \in \mathbb{D}^m$, each of whose elements is sampled uniformly at random from the set $\mathbb{D} = [-1, 1]$. Define the distance as α_m. Consider first $m = 1$. Then, α_1, the expected Manhattan distance between x and y, can be defined as

$$\alpha_1 = \frac{1}{R} \int_{-1}^{+1} \int_{-1}^{+1} |x - y| \, dx \, dy,$$

where $R = 4$ is the area of the square $[-1, 1] \times [-1, 1]$. Integrating the above we get,

$$\begin{aligned}
\alpha_1 &= \frac{1}{4} \int_{-1}^{+1} \left(\int_{-1}^{y} (y - x) \, dx + \int_{y}^{+1} (x - y) \, dx \right) dy \\
&= \frac{1}{4} \int_{-1}^{+1} (y^2 + 1) \, dy
\end{aligned}$$

[2] Note that if the adversary just guesses randomly, the advantage is 0. This is significantly greater than 0.

[3] i.e., $\Pr[b' = 1 \mid b = 1] = \frac{\Pr[b'=1 \wedge b=1]}{\Pr[b=1]} = \mathrm{TPR}$ and $\Pr[b' = 1 \mid b = 0] = \frac{\Pr[b'=1 \wedge b=0]}{\Pr[b=0]} = \mathrm{FPR}$

$$= \frac{1}{4} \cdot \frac{8}{3} = \frac{2}{3}.$$

By independence, we get $\alpha_m = m\alpha_1 = 2m/3$. For $m = 5$, we get $\alpha_5 = 10/3 \approx 3.33$. Thus, we set $\alpha = 3.33$ as the benchmark for a random guess with 5 missing features in the CIFAR dataset.