MIAShield: Defending Membership Inference Attacks via Preemptive Exclusion of Members

Ismat Jarin
ijarin@umich.edu
University of Michigan, Dearborn
United States

Birhanu Eshete
birhanu@umich.edu
University of Michigan, Dearborn
United States

ABSTRACT

In membership inference attacks (MIAs), an adversary observes the predictions of a model to determine whether a sample is part of the model's training data. Existing MIA defenses *conceal the presence of a target sample* through strong regularization, knowledge distillation, confidence masking, or differential privacy.

We propose MIAShield, a new MIA defense based on *preemptive exclusion of member samples* instead of masking the presence of a member. MIAShield departs from prior defenses in that it weakens the strong membership signal that stems from the presence of a target sample by preemptively excluding it at prediction time without compromising model utility. To that end, we design and evaluate a suite of preemptive *exclusion oracles* leveraging model confidence, exact/approximate sample signature, and learning-based exclusion of member data points. To be practical, MIAShield splits a training data into *disjoint subsets* and trains each subset to build an ensemble of models. The disjointedness of subsets ensures that a target sample belongs to only one subset, which isolates the sample to facilitate the preemptive exclusion goal.

We evaluate MIAShield on three benchmark image classification datasets and show that it reduces membership inference to nearly random guess for a wide range of MIAs; achieves far better privacy/utility trade-off compared with state-of-the-art defenses; and remains resilient in the face of adaptive attacks.

KEYWORDS

Membership Inference Attacks, Machine Learning, Privacy

1 INTRODUCTION

One of the main risks of training machine learning (ML) models on privacy-sensitive data (e.g., medical/location data) is models can inadvertently leak information about training data-points —resulting in violation of individual's privacy. One form of information leakage in ML is via membership inference attacks (MIAs) [36]. In a MIA, given a data-point and a model, the adversary's goal is to determine whether the data-point was used to train the model. Such an attack poses realistic threats to the privacy of individuals who contribute data points to models trained on privacy-sensitive data. For instance, suppose a cancer patient's tumor image is used to train a model that classifies images into benign or cancerous. If an adversary with access to the patient's image can infer the presence of the image in the training set of the model, it is a privacy breach.

As documented by prior work [3, 7, 17, 25, 32, 33, 36], the success of MIAs has is largely attributed to a target model's statistically distinguishable behaviors in its predictions on members and non-members of its training data, which in turn is attributed to *overfitting* of models on members. This difference is exploited by an adversary as a strong signal to learn a member/non-member *decision function* (attack model) or some *threshold* to flag members. Based on the information they leverage, MIAs can be *probability-dependent* attacks (use confidence scores predicted for each class) or *label-dependent* attacks (use just the predicted label) [14].

A common thread in existing MIA defenses is the focus on *concealing the presence of a data-point* through strong l_2 regularization, prediction confidence masking, model ensemble, knowledge distillation, or differential privacy. Strong (l_2) regularization methods [24, 27] reduce overfitting and differential privacy-based defenses [1, 15, 16, 30] offer provable privacy guarantees, but they both succeed at the expense of model utility. Distillation-based methods such as [35] achieve better MIA resilience with tolerable utility loss but typically depend on publicly accessible data. Confidence masking methods [18, 42] are ideal to preserve utility with reasonable MIA resilience but are inherently vulnerable to label-dependent MIAs.

In this paper, we step back and ask the question:
If the presence of a target data-point offers a strong signal for MIA, does excluding the data-point without compromising the utility of the model weaken membership signal and hence mitigate the attack?

To systematically investigate this question, we introduce MIAShield—a new defense against MIAs. Instead of basing the defense on masking a target data-point, MIAShield is based on *preemptive exclusion of a target data-point to weaken the membership signal that stems from the presence of the data-point in the training data of a model.* To make MIAShield practical, we split the training data into *disjoint subsets* and train each subset to build an ensemble of models. The disjointedness of subsets ensures that a target data-point belongs to a unique subset —which serves as a precursor for the preemptive exclusion goal. Once the ensemble is operational (e.g., as an MLaaS API), given a data-point, MIAShield first queries an *exclusion oracle* to determine whether the data-point is a member of the training set of one of the models in the ensemble. If so, it excludes the matching model from participating in the ensemble to avoid the *overfitting* of the model to the target data-point. Otherwise, MIAShield uses the whole ensemble to compute prediction on the data-point. By excluding a model that was trained on a target data-point, MIAShield essentially disarms the adversary of its

Proceedings on Privacy Enhancing Technologies 2023(1), 400–416

https://doi.org/10.56553/popets-2023-0024

attack vantage point, i.e., the strong membership signals the model gives away to a MIA adversary on member data-points.

Since the success of MIAShield depends on the effectiveness of the exclusion oracle, we design and evaluate a spectrum of member exclusion strategies. Beginning with a *model confidence-based exclusion* as a baseline (§5.1), we then explore *exact* and *approximate* (data-point) signature-based exclusion strategies (§5.2 – §5.3). Next, we frame the preemptive exclusion goal as a *classifier-based* exclusion strategy (§5.4). Finally, we study a *chain of exclusion oracles* in which MIAShield begins with exact signature-based oracle, then approximate signature-based oracle, and finally resorts to the classifier-based exclusion oracle (§5.5).

In our evaluation, we focus on image classification for two reasons. First, it is the domain where MIA has been extensively studied. Second, it is the domain that has immediate applications to privacy-sensitive data such as medical images and facial recognition data. Against an adversary with black-box access to a deployed model, we evaluate MIAShield on three benchmark image classification datasets against state-of-the-art attacks and defenses. Our results against 7 MIAs consistently suggest that MIAShield drops MIA accuracy to $\approx 50\%$ (random guessing) and incurs no more than 1% drop in model utility compared to the non-private model (§6.5). Compared with 5 prior defenses, MIAShield offers an overall better privacy-utility trade-off with respect to DP-SGD [1], PATE [30], Model-Stacking [33], MemGuard [18], and MMD-MixUp [24] (§6.6). Moreover, MIAShield is resilient in the face of adaptive attacks that exploit side-channel information and knowledge about MIAShield details (§6.7). In summary, we make the following key contributions:

- We propose MIAShield, an exclusion oracle-guided approach that fundamentally rethinks MIA defense by ensuring that a target data-point does not contribute to the MIA signal via the model's predictions on member data-points.
- We show that MIAShield outperforms prior defenses and offers consistently better privacy-utility trade-off.
- We show that MIAShield mitigates two complementary threat models: probability-dependent attacks and label-dependent attacks.
- We demonstrate that MIAShield remains resilient to adaptive attacks that exploit knowledge about our defense.
- We make MIAShield code publicly available at: https://github.com/um-dsp/MIAShield.

2 BACKGROUND

As background, we briefly overview the ML setting we consider (§2.1) and introduce MIAs (§2.2).

2.1 Machine Learning Setting

We focus on supervised ML models. Given a set of labeled training samples $\mathcal{D}^{train} = (X_i, y_i) : i \leq n$ such that X_i is a d-dimensional training example and its corresponding label $y_i \in Y$ (a k-dimensional output space), the model parameterized by model parameter vector θ is denoted as f_θ. Training f aims to minimize the expected loss $J(\theta) = \frac{1}{n}\sum_1^n \mathcal{L}(f_\theta, X_i, y_i)$ over all (X_i, y_i). The loss minimization problem is typically solved using stochastic gradient descent (SGD) vi an iterative update of θ as $\theta_{i+1} = \theta_i - \eta \cdot \Delta_\theta \sum_{i=1}^n \mathcal{L}(\theta_i, X_i, y_i)$, with Δ_θ as the gradient of $J(\theta)$ and η as *learning rate*. A prediction of an input x is $\hat{y} = f_\theta(x) = argmax(y \in Y)$, where y is a k-dimensional vector and each dimension represents the probability of x belonging to the corresponding class. Hereafter, we use f to refer to f_θ.

2.2 Membership Inference Attacks

A MIA is a statistical attack where an adversary aims to infer whether $x \in \mathcal{D}^{train}$. Given a candidate sample x, a model f trained on $\mathcal{D}^{train}$, and adversary's knowledge (about f and $\mathcal{D}^{train}$) denoted by $\mathcal{K}$, the goal of MIA is to determine whether x is used to train f. More formally, MIA is defined as an attack function $\mathcal{A}$ as: $\mathcal{A}(x, f, \mathcal{K}) \rightarrow \{0, 1\}$, where 0 means $x \notin \mathcal{D}^{train}$ and 1 means $x \in \mathcal{D}^{train}$. In the first MIA [36] against ML models and subsequent attacks [33, 38, 43] the attack function $\mathcal{A}$ is typically a binary classifier which is trained based on the fidelity of $\mathcal{K}$ with respect to how much the adversary knows about f and $\mathcal{D}^{train}$. The success of MIAs is attributed to *overfitting* —models are more confident in their predictions on members of their training data. Shokri et al. [36] use multiple *shadow models* (models that imitate the target model) to train an attack model. In a recent MIA, Salem et al. [33] significantly reduce the dependency of [36] on shadow models using as little as none and at most three shadow models.

3 THREAT MODEL

Attacker Capabilities. Via a *black-box* prediction API access to f, the adversary submits a sample x and obtains prediction in a form of *probability vector* y and/or a *label* $\hat{y}$. We anticipate the adversary to launch one of: a) single-step single query attack with access to both y and $\hat{y}$; b) single-step single query attack with access to only $\hat{y}$; c) multi-query attacks with access to only $\hat{y}$; and d) adaptive attacks that leverage side-channel information and/or knowledge of exclusion strategy with access to y and/or $\hat{y}$.

Attacker Knowledge. We assume no access to model parameters, θ, and $\mathcal{D}^{train}$ except a small subset of $\mathcal{D}^{train}$ which the adversary needs any way to launch meaningful MIA. Additionally, we assume that the adversary knows the architecture of target model since state-of-the-art model architectures (e.g., image classification models) are often public knowledge. For probability-dependent attacks, the attacker only follows step (a). For label-dependent baseline attacks, the adversary follows step (b). For label-only augmentation attacks, the adversary follows step (c) and (d), along with the knowledge of target model architecture. For label-only boundary distance attack, the adversary follows step (c) with no knowledge of model architecture.

Defender. We assume the defender has access to a privacy-sensitive training data $\mathcal{D}^{train}$ and their goal is to train and deploy a model that not only offers high prediction accuracy but also is resilient against MIAs that leverage its predictions to infer the presence of a data-point in $\mathcal{D}^{train}$. In addition, the defender may employ accuracy enhancement/recovery operations (e.g., data augmentation) to gain back potential accuracy loss due to splitting $\mathcal{D}^{train}$ into n disjoint subsets.

4 RELATED WORK

We review related work focusing on attacks and defenses. We refer the interested reader to [14] for a comprehensive survey.

4.1 Membership Inference Attacks

Based on the fidelity of predictions returned by a model, we categorize MIAs into *probability-dependent* and *label-dependent* attacks.

Probability-Dependent MIAs rely on probability scores returned by the target model for a given input sample. When ML models are overfitted to their training data, they produce more confident predictions on members. Attackers exploit this statistical predictability of models on members to determine the presence of a target sample in a model's training data. The probability is leveraged by training an attack model $\mathcal{A}$ to predict members/non-members [28, 36] or by computing a threshold τ that winnows members from non-members [43]. When an attack model is trained, shadow models are typically trained over the predictions of every class [36], a single shadow model is trained using top predictions (e.g., top-3, top-1) [33], or a threshold (e.g., of prediction loss) is used without training a shadow model [43].

In **Label-Dependent MIAs** class labels are the only signal available to an adversary. The adversary leverages the sensitivity of member data points to random or adversarially crafted noise.

Gap Attack by Yeom et al. [43] flags a sample as a member if the predicted label is correct, otherwise a non-member. The naive assumption in the Gap Attack is extended by Choquette-Choo et al. [7] in their *label-only* MIA. Around the same time, a closely similar label-only attack was introduced by Li and Zhang [25]. This family of MIAs is realized via two strategies, namely *augmentation attack* and *boundary distance attack*.

In *Augmentation Attack*, the adversary observes the sensitivity of a sample to data augmentation techniques such as *rotation* and *translation*. The key insight is that members are often more insensitive to data augmentation than non-members. For example, an image used to train a model will mostly be classified correctly despite slight manipulations (e.g., rotation by 1°). For the attack to be practical, multiple manipulated variants of each sample (e.g., image) are generated and labeled by the target model. Using the labeled augmented data, the adversary then trains a binary attack model that flags samples as members or non-members.

In *Boundary Distance Attack*, the insight is to leverage perturbation methods used in the adversarial examples literature [4, 5, 8, 11, 23, 26] towards distinguishing members from non-members. This is possible because non-members (e.g., model's test samples) are relatively more sensitive to perturbations due to their proximity to the decision boundary of the model most of the time. Based on the magnitude of the perturbation that results in incorrect prediction, the adversary sets a threshold to put apart members from non-members. While adversarial perturbations are attractive in this context, adding random noise has also been shown to be effective especially when the perturbation space is narrow (e.g., in features with limited values or value range) [7].

4.2 Membership Inference Defenses

We present defenses based on regularization, confidence masking, ensemble techniques, and differential privacy.

Regularization techniques such as dropout [39], weight decay [41], and l_2 regularization aim to mitigate overfitting. Nasr et al. [27] train a model with membership privacy via a regularization parameter on an optimization method that achieves minimum utility loss against powerful MIAs. In [24], the defense minimizes the difference between probability vector distribution of the same class for members and non-members by using a new set of regularization methods, thus minimizing the generalization gap. This regularization method is accompanied by a mix-up augmentation technique to further deter an attack and is shown to outperform defenses like MemGuard [18] and DP-SGD [1]. These methods often improve a model's resilience to MIAs at the expense of model utility.

Confidence Masking defenses [18, 42] add noise to prediction confidence scores to conceal the true confidence of the model and hence minimize MIA effectiveness. For instance, providing confidence vectors of top-k classes instead of the complete set of confidence scores. MemGuard [18] adds carefully crafted label-preserving noise to the confidence scores by leveraging adversarial example crafting methods. Confidence masking methods are ideal to preserve utility with reasonable MIA resilience but are vulnerable to label-dependent MIAs such as [25] and [7].

Ensemble Techniques [15, 30, 31, 33, 40] aim to reduce MIA risk by aggregating multiple models' decisions to provide final output. Model stacking [33] uses a 2-layer setup where the first layer contains a neural network and a Random Forest models trained on disjoint subsets of the original training data. The combination of the outputs from the first layer is passed to a Logistic Regression model for a final prediction. PATE [30] and PATEG-G [31] split a dataset into disjoint subsets to train an ensemble of teacher models with sensitive data to provide a noisy majority vote transferred to a student model. Related defenses such as PRICURE [15] also rely on similar ensemble ideas as PATE for noisy ensemble aggregation. We note that PATE [30], PATEG-G [31], and PRICURE [15] are beyond ensemble as they are practically a hybrid of ensemble and differential privacy. SELENA [40], a closely related work to MIAShield, combines ensemble of models trained on random subsets of training samples with self-distillation to train a *protected* model that behaves the same way on members and non-members.

Differential Privacy based methods such as DP-SGD [1] and [6] introduce randomness to the ML training process, while others focus on de-noising [29] or input perturbation [9]. In DP-SGD [1], a differentially private training mechanism is proposed where noise is added to the clipped value of the gradient. When a model is trained with DP using a small privacy budget ϵ, the model does not remember specific user details. As a result, this technique attenuates MIAs and offers strong privacy guarantees. Though such approaches are effective in provably ensuring membership privacy, they suffer from notable utility loss.

MIAShield vs. Existing Defenses. While existing defenses aim at concealing the presence of a MIA target sample, MIAShield takes a fundamentally different approach that is based on the exclusion of the target sample to eliminate the strong membership signal a target sample gives away to an adversary. In §6.6, we present detailed comparison of MIAShield with DP-SGD [1], PATE [30], MemGuard [18], Model-Stacking [33], and MMD-MixUp [24]. The ensemble design of SELENA [40] resembles the training of disjoint subsets in MIAShield. However, random subsets in SELENA may

not be disjoint (hence a target data-point may belong in multiple subsets). Moreover, models excluded from SELENA's ensemble aggregation are based on exact matching, while in MIAShield we design and evaluate a spectrum of exclusion strategies that covers model confidence, exact, approximate, and probabilistic matching.

5 MIASHIELD DEFENSE APPROACH

Defense Intuition. As widely acknowledged by prior work [7, 25, 33, 36], the success of MIA is largely attributed to exploiting the difference in a model's prediction behavior on members and non-members. This difference in turn is attributed to *overfitting*: ML models are often correct and more confident on members of their training data than on non-members. MIAShield fundamentally rethinks MIA defense through preemptive exclusion of members. Unlike prior defenses [1, 18, 24, 30, 33, 35] that base the defense on masking a target data-point, MIAShield instead is based on *preemptive exclusion of a target data-point to weaken the strong membership signal due to the presence of a member data-point.*

Figure 1 shows an overview of MIAShield. First, a sensitive training data $\mathcal{D}$ is split to n disjoint subsets $\mathcal{D}_1 ... \mathcal{D}_n$, from which an ensemble of models $f_1, ..., f_n$ is trained. By making the $\mathcal{D}_i$'s disjoint, MIAShield ensures that a target data-point belongs to one subset only —which goes well with the preemptive exclusion goal. Given a data-point x for prediction, an exclusion oracle eliminates a model f_i that contains x in its training data ($\mathcal{D}_i$) and returns a prediction $y = \Phi(f_1(x), ..., \cancel{f_i(x)}, ..., f_n(x))$ based on an ensemble of $n-1$ models. Φ is an ensemble aggregation function (e.g., majority vote). Otherwise, all n models participate in the ensemble prediction, and it returns $y = \Phi(f_1(x), ..., f_n(x))$. By excluding f_i trained on x, MIAShield disarms the adversary of its attack advantage, i.e., the strong membership signal emitted by the presence of x in f_i's training set $\mathcal{D}_i$. A crucial utility preservation constraint is that the correct label of x is maintained whether the model trained on x is excluded or not. To fulfil this constraint, we leverage data augmentation (i.e., $A(\mathcal{D}_1), ..., A(\mathcal{D}_n)$ in Figure 1) to regain accuracy loss to splitting of $\mathcal{D}$ into n disjoint subsets (details in §6.3).

Since MIAShield's effectiveness depends on the accuracy of the exclusion oracle, the main challenge here is to ensure that the exclusion oracle does not exclude the wrong model (false positive) or fail to exclude the right model (false negative). Towards addressing this challenge, we propose and evaluate a spectrum of member exclusion methods beginning with a baseline exclusion strategy:

- **Model Confidence-Based.** Among $f_1, ..., f_n$, exclude the most confident model on the most-voted prediction. This serves as a naive baseline for it goes well with the root cause of MIAs, i.e., models tend to overfit on member data-points.
- **Exact Signature Matching-based.** Among $f_1, ..., f_n$, identify f_i trained on $\mathcal{D}_i$ s.t. $x \in \mathcal{D}_i$ using deterministic signature matching methods (e.g., cryptographic hash comparison).
- **Approximate Signature Matching-Based.** Whenever $\exists$ $x' \in \mathcal{D}_i$ s.t. $x \approx x'$, exclude the model trained on x' using inexact matching (e.g., based on signatures obtained via perceptual hashing of images [13]). This method is effective especially when the exact matching method yields no match for cases in which the target data-point is not exactly matched but close-enough to other data-points (e.g., genome data, images that differ in one pixel).
- **Classifier-Based.** Treat the exclusion as a classification problem and from a subset of each $\mathcal{D}_i$ learn a function f_{eo} that predicts whether x is a member of one of the $\mathcal{D}_i$'s. This method provides a probabilistic way to determine the model to exclude.
- **Chain of Oracles.** When exact signature matching yields no match, use the approximate signature matching. Resort to classifier-based method as the last line of exclusion only when approximate signature matching yields no match.

Before we elaborate on the aforementioned exclusion strategies, we note that: for probability-dependent attacks, the ensemble aggregation function $\Phi(...)$ returns the average of probability scores of the most voted label and the label itself; for label-dependent attacks, it returns the most voted label.

5.1 Model-Confidence-Based Exclusion (MCE)

An overfitted model is more confident on members and it is well-documented that MIAs are mainly attributed to overfitting [33, 36]. Hence, as a baseline exclusion strategy, we *exclude the most confident model on the most-voted prediction.* In MCE, given a data-point x and models $f_1, ..., f_n$ with corresponding confidence scores $f_1(x) = c_1, ..., f_n(x) = c_n$ on the most-voted prediction, if $argmax(c_1, ..., c_n) = i$, then model f_i is excluded from the ensemble.

A natural question then is whether the most confident model on the most-voted prediction is always the model that was trained on the target data-point. This may not always be the case because models would predict a label with high confidence, but the label may turn out to be the wrong one. This is especially true when, during training, a model picks up spurious correlations instead of truly distinguishing features of a training sample. Even so, establishing MCE as baseline enables us to motivate the need for more accurate alternative exclusion strategies. In §6.5, we compare the effectiveness MCE with the exclusion oracles presented next.

5.2 Exact-Signature-Based Exclusion (ESE)

In ESE, we first compute signature of each sample in $\mathcal{D}_1, ..., \mathcal{D}_n$ using a cryptographic hash function H. This is a one-time offline computation and does not lead to performance bottleneck. Given a target data-point x, we first compute $H(x)$ and search for its match in the hash values of samples in $\mathcal{D}_1, ..., \mathcal{D}_n$. If a match is found (say in $\mathcal{D}_i$) then the corresponding model f_i is excluded from participating in the ensemble prediction. Otherwise, all f_i's participate in computing x's label. This mechanism is deterministic in that if $x \in \mathcal{D}_i$, it is always possible to match its hash value (assuming zero collision).

Three factors contribute to the efficiency of ESE: (1) the hashing algorithm employed (2) the search algorithm and (3) whether the search is parallelized. On (1), H serves the purpose of quickly computing a unique signature for a sample. To that end, it suffices to use faster hash functions (e.g., $MD5$, $SHA-1$) since our goal in employing a hash function here is not conditioned to getting more secure cryptographic hash function. On (2), the choice of the search algorithm and the data structure used to represent the hash values determines how fast one can lookup for a sample. Linear

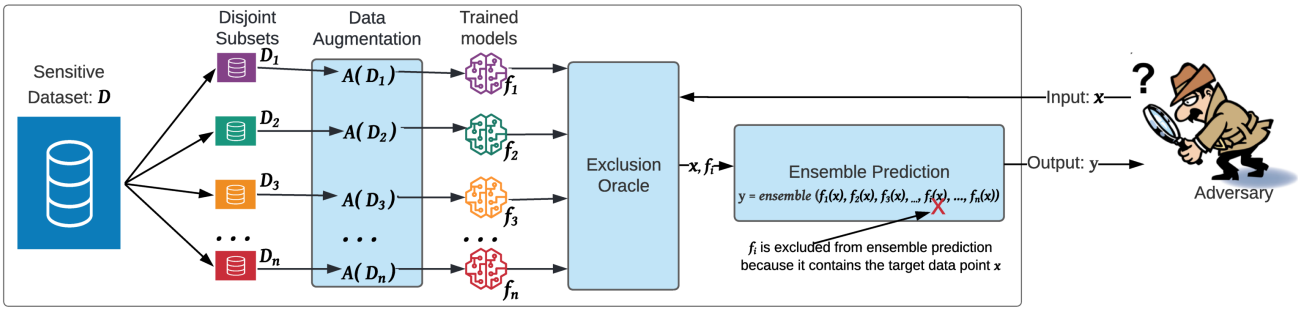


Figure 1: Overview of the MIAShield pipeline.

search takes $O(n)$ (where n is the size of the search space) while binary search will cost by $O(\log n)$). Even better, if a hash-table is used lookup will take $O(1)$. On (3), parallelization of the search significantly speeds up the matching. In our experiments, to speed up the matching, we convert hash values to integers and sort them to ease binary search.

Despite the exactness of ESE and its potential to be high accuracy exclusion, it has two limitations. First, it does not detect members if an adversary slightly modifies inputs (e.g., changes one pixel in an image). Second, it is vulnerable to timing attacks whereby an adversary carefully observes response-time difference of the prediction API on members and non-members.

For member data-points, the response time depends on where in a model's training data the matching data-point is located, and hence unpredictable. For non-members, on the other hand, the search takes longer (since it must be exhaustive) and about the same for all data-points (because the search space is static).

An adversary may use this predictability to put apart members and non-members by keeping inventory on response time of each query. To disrupt the adversary's pursuit of estimating a response-time threshold that separates members from non-members, we equip MIAShield with two countermeasures: (a) before lookup, we reshuffle the hash values of data-points in each subset $\mathcal{D}_i$ and (b) on member queries, we introduce random delay upper-bounded by the average delay on non-members. In §6.7, we show empirical evidence that demonstrates effectiveness of combining (a) and (b) to make MIAShield resilient in the face of timing attacks.

When an adversary makes minimal modification on a target sample x, ESE oracle misses x which may be almost the same as a member sample x'. For instance, x' differs from x by just a pixel. Attack model-based MIAs (e.g., [28, 36]) that leverage higher confidence of the model on members may succeed in such a scenario because ESE fails to exclude $x' \approx x$. To avoid such a pitfall, next we consider approximate signature matching but still based on hash values such that our exclusion oracle considers an adversary who minimally manipulates a data-point to bypass ESE.

5.3 Approximate-Signature-Based Exclusion (ASE)

ESE is naturally fit for cases where the adversary is unlikely to manipulate a target data-point. When the adversary performs slight manipulations (e.g., single-pixel change, adjusting brightness of an image), ESE results in a false negative (misses a member data-point that very slightly differs from a target data-point). To make MIAShield resilient to such slight label-preserving manipulations, we turn to *perceptual hashing* methods which have been widely used to search for similar images in domains like digital forensics, cybercrime analysis, and image search engines [13].

Perceptual hashing algorithms generate a fingerprint for each image so that similar-looking images are mapped to the same or similar hash code. Unlike conventional cryptographic hashing algorithms such as MD5 and SHA1 that generate distinct hash values for slightly modified inputs, perceptual hashing is designed to tolerate small perturbations so that a slightly manipulated image still produces similar hash values.

Given an image x, a perceptual hashing function H_p produces a binary string as the hash code: $h = H_p(x), h \in \{0/1\}^l$, where $\{0/1\}^l$ represents a binary string of length l. For a given data-point x, our exclusion oracle uses H_p (e.g., *pHash*) to compute $x' = H_p(x)$ and matches it against similar hash values of images in $\mathcal{D}_1 ... \mathcal{D}_n$. The key advantage over the ESE oracle is that ASE is now able to match x (which may have been modified by an adversary via operations such as rotation, pixel change, or brightness change) with visually similar data-points in $\mathcal{D}_1 ... \mathcal{D}_n$.

When doing perceptual hashing-based matching, a certain threshold is set based on a distance metric between two hash values. A widely used distance metric is the *normalized Hamming distance* —which measures the number of different bits between the two hash strings divided by the length of the hash string. The normalized Hamming distance value falls within [0,1].

While ASE addresses the limitation of ESE pertinent to slightly manipulated samples, an adversary with knowledge of the usage of perceptual hashing can still attempt to bypass it. For instance, one potential threat is to incrementally manipulate (e.g., rotate, perturb) a sample and watch for noticeable *change* in predictions (e.g., probability score changes). When it happens, such observable change can lead to the disclosure of the hamming distance threshold. We will revisit this threat in §6.7.

5.4 Classifier-Based Exclusion (CBE)

Complementary to ESE and ASE, in CBE we frame member data-point exclusion as a statistical learning objective, where the underlying data distribution in $\mathcal{D}_1, ..., \mathcal{D}_n$ is leveraged as a basis to predict which model to exclude. To this end, in CBE we treat the

exclusion of a data-point as probabilistic prediction task for which we train an exclusion oracle model that, given a data-point, predicts the to-be-excluded model. We first identify features to characterize each data-point of $\mathcal{D}$. A feature vector is composed of a subset of features (X_{PCA}) of data-points in $\mathcal{D}$, a confidence vector (C) and a label (y) returned by a model trained on $\mathcal{D}$. In the following, we expand on what comprises of the feature vector $[X_{PCA}, C, y]$.

Subset of Features (X_{PCA}): Instead of using the whole feature set of data-points in $\mathcal{D}$, the defender takes advantage of the white-box access to $\mathcal{D}$. Hence, we perform principal component analysis (PCA) on $\mathcal{D}$ to determine a subset of the features (which we call X_{PCA}) such that X_{PCA} contains m features ($m < d$, where d is feature dimension of a sample in $\mathcal{D}$). The reason behind taking a subset of the features is twofold. First, we aim for a lightweight model that does not take long to train. Second, we aim to minimize what the exclusion oracle inherits from $\mathcal{D}$ —if there is overfitting inherent to $\mathcal{D}$, focusing on the more 'robust' features obtained via PCA limits the likelihood of propagating the overfitting to the training set of the exclusion oracle.

Confidence Score Vector (C): For each x in $\mathcal{D}_i$, a model trained on $\mathcal{D}$ returns a confidence score vector of the same dimension as the number of classes k, which is a probability distribution over k classes. We include $C = [c_1, ..., c_k]$ in our feature vector such that $\Sigma_{i=1}^{k} c_i = 1$. We note that C resembles the confidence vector used in training an attack model in MIAs (e.g., [28, 36]). However, in our case C is just a part of our feature vector, while in typical shadow models-based MIAs it is the decisive part of the feature vector in the training of the attack model.

Predicted Label (y): To further enrich our feature vector, we also add the predicted label $y \in \{1, ..., k\}$. The rationale for using y is that in prior work [36] it has been shown that there is positive correlation between MIA and the output label.

Model Index (l): This is the target label for our exclusion oracle model. The label is an integer in the range $[1, n+1]$, where $1 = f_1$, ..., $n = f_n$, and $n+1$ means none of the f_i's among $f_1, ..., f_n$ is excluded (i.e., all models participate in the ensemble).

Our reader may wonder "how CBE differs from an attack model of typical MIA?". On the surface, CBE appears to be yet another binary member/non-member attack model similar to the likes of Shokri et al. [36]. Compared with the MIA adversary, we argue that the defender is in a more advantageous position. Specifically, the defender has full access to the training data $\mathcal{D}$, the disjoint subsets $\mathcal{D}_1, ..., \mathcal{D}_n$, and additional information such as each model's overfitting score. All these details are typically unavailable to a black-box MIA adversary that we consider in our threat model.

An intuitive approach to build CBE is to train it on a combination of features from each $\mathcal{D}_i$ (i.e., members) and an $(n+1)^{th}$ class (i.e., non-members). In our case, this approach turned out to be less effective in correctly labeling non-members that come from outside the test set. To overcome this, we use a threshold-based mechanism that leverages probability scores on member data-points.

The CBE oracle is built in two steps. In step-1, we train a model f_{eo} on feature vectors of only members of the form $[X_{PCA}, C, y]$ based on a subset of each $\mathcal{D}_i$ in $\mathcal{D}_1, ..., \mathcal{D}_n$. Model f_{eo} predicts one of $1...n$. For instance, $f_{eo}(x) = 3$ means $x \in \mathcal{D}_3$.

In step-2, we establish an exclusion threshold τ_{eo} through empirical observation of the confidence scores produced by f_{eo} on members (coming from $\mathcal{D}_i$'s) and non-members (coming from test/validation set). The key intuition behind the estimation of τ_{eo} is that for non-members, confidence scores returned by f_{eo} will be lower compared to the member counterparts because f_{eo} tends to be more confident on its training samples. Now, for a given data-point x, we compute $y_l = f_{eo}(x)$, where y_l is an n-dimensional probability score vector (n = number of models in the ensemble). To compute the final label (model-index l) for exclusion oracle, we follow the following condition: If $max(y_l) \geq \tau_{eo}$, the model index is computed as $l = argmax(y_l)$ where $l \in (1, ..., n)$, otherwise the label l is $n+1$. Over a sample of possible threshold values in the range $[0, 1]$, we compute the final exclusion oracle accuracy Acc_{eo} over the aforementioned condition and fix the τ_{eo} value for which Acc_{eo} is maximum and use it for future exclusions.

5.5 Chain of Exclusion Oracles (COE)

Inherent limitations of hash value-based exclusion oracles (ESE and ASE) may result in missing a member data-point. This may, in effect, misguide MIAShield to be tricked by a MIA adversary with knowledge about either the hashing algorithms and/or Hamming distance threshold (for the perceptual hashing case). To mitigate this threat, in COE we chain the oracles in the order ESE→ASE→CBE such that CBE is the last resort if ESE→ASE yields no match. Our reader may wonder as to the benefit of chaining the exclusion oracles in such a sequence. The benefit is that a (very small) percentage of member data-points is likely to be missed by ESE and ASE and may be correctly flagged by the CBE. Why? Because, unlike ESE and ASE which rely on exact and approximate signature of data-points, respectively, CBE learns membership signals based on the underlying features of members' data distribution. Note that COE may inherit the timing attack risk from ESE and ASE. The two countermeasures we introduced in §5.2 (reshuffling and bounded delay on member queries) remain as effective countermeasures to hinder the success of timing attack.

6 EVALUATION

We evaluate MIAShield on 3 image classification datasets against 7 attacks and compare it with 5 related defenses. We also evaluate its resilience in the face of adaptive attacks. Our evaluation is guided by the following research questions:

- **RQ1**: How do the exclusion oracles compare among each other and between probability-dependent and label-dependent attacks?
- **RQ2**: How does MIAShield compare with state-of-the-art MIA defenses on utility-privacy trade-off?
- **RQ3**: How resilient is MIAShield in the face of adaptive attacks that strive to bypass it by leveraging side-channels and knowledge about the exclusion oracles?

6.1 Datasets and Partitioning

We use three benchmark datasets: CIFAR-10 [20], CIFAR-100 [21], and CH-MNIST [19], which we briefly describe next.

Dataset	$\mathcal{D}^{train}$	$\mathcal{D}^{test}$	n	$\mathcal{D}^{train}_{EO}$	$\mathcal{D}^{test}_{\text{MIAShield}}$
CIFAR-10 [20]	50K	10K	5	2.5K×n (members) + 5K (non-members)	5K (members) + 5K (non-members)
CIFAR-100 [21]	50K	10K	4	2.5K×n (members) + 5K (non-members)	5K (members) + 5K (non-members)
CH-MNIST [19]	4K	1K	4	500×n (members) + 500 (non-members)	500 (members) + 500 (non-members)

Table 1: Dataset partitioning in the evaluation of MIAShield.

CIFAR-10 [20] consists of 60K color images of 10 classes. Each image is $32 \times 32 \times 3$ pixels. The target classes include 10 object images (e.g., airplane, truck, automobile, dog, bird, frog).

CIFAR-100 [21] has 100 classes, 600 images each. Per class, there are 500 training images and 100 test images. The classes include different object names, for example, fishes (e.g., aquarium fish, flatfish), flowers (e.g., orchids, poppies) etc.

CH-MNIST [19] contains samples of histology tiles from patients with colorectal cancer and has eight target classes. It contains 5K images in total. The size of each image is 64×64 pixels.

Partitioning of Datasets: As shown in Table 1, for each dataset, given a train set $\mathcal{D}^{train}$ (members) and $\mathcal{D}^{test}$ (non-members), we split $\mathcal{D}^{train}$ into n disjoint subsets ($\mathcal{D}^{train}_1$, ..., $\mathcal{D}^{train}_n$) such that each $\mathcal{D}^{train}_i$ has $\frac{|\mathcal{D}^{train}|}{n}$ samples. To train the CBE oracle, we use $\approx$ 25% of the samples in each $\mathcal{D}^{train}_i$ to collectively represent n models from the ensemble. To represent the $(n+1)^{th}$ model (i.e., non-member), we use $\approx$ 50% of samples from $\mathcal{D}^{test}$. Combining the two, we get the $\mathcal{D}^{train}_{EO}$ column in Table 1. Inline with prior work [24, 25, 33, 36], we use a balanced number of members $\mathcal{D}^{test}_{mem} \subset \mathcal{D}^{train}$ and non-members $\mathcal{D}^{test}_{non-mem} \subset \mathcal{D}^{test}$ (the $\mathcal{D}^{test}_{\text{MIAShield}}$ column in Table 1). Balancing members and non-members is crucial to establish 50% (random guess) MIA accuracy as a baseline. To avoid potential bias, we ensure that $\mathcal{D}^{test}_{mem}$ and $\mathcal{D}^{train}_{EO}$ are disjoint.

6.2 Models

Non-Private Model. For all datasets, we use the AlexNet [22] CNN architecture shown in Appendix A (Table 4). The model is trained on $\mathcal{D}^{train}$ using *Adam* optimizer and categorical cross-entropy as loss function. Number of epochs for CIFAR-10, CIFAR-100, and CH-MNIST is 60, 130, and 200, respectively. Batch size = 128 and learning rate = 0.01 for all datasets.

MIAShield Ensemble Models. Based on the dataset split in Table 1, we train MIAShield models from the disjoint subsets using the same AlexNet architecture in Appendix A (Table 4).

Exclusion Oracle Model. In step-1, we train a Random Forest (RF) classifier on $\mathcal{D}^{mem}_{EO}$ where $\mathcal{D}^{mem}_{EO} = [X^{mem}_{PCA}, C, y, l]$ as described in §5.4. It is noteworthy that $\mathcal{D}^{mem}_{EO}$ contains 2.5K×n samples for CIFAR-10 and CIFAR-100 and 0.5K×n samples for CH-MNIST, where n is the number of models. In step-2, we query the trained RF model with both D^{mem}_{EO} and $D^{non-mem}_{EO}$ and then calculate Acc_{EO} over 20 threshold samples $t \in (0, 1)$, where the threshold value starts with 0 and is incremented by $\frac{(1-0)}{20}$ up to 1. The threshold value that led to the maximum exclusion oracle accuracy is chosen as the final threshold for evaluation. Accordingly, for CIFAR-10, CIFAR-100 and CH-MNIST, the threshold values are 0.38, 0.52, and 0.47, respectively.

Model Name	Acc. (Aug.)	Acc (No-Aug.)
f_1	65.72%	56.51%
f_2	64.52%	57.52%
f_3	63.1%	57.48%
f_4	65.42%	59.08%
f_5	66.46%	55.74%

Table 2: CIFAR-10: accuracy pre- and post-augmentation.

Model Name	Acc. (Aug)	Acc. (No-Aug)
f_1	37.02%	27.22%
f_2	34.86%	25.1%
f_3	37.52%	26.86%
f_4	36.68%	27.2%

Table 3: CIFAR-100: accuracy pre- and post-augmentation.

Attack Models. For probability-dependent attacks, we use [38] and use the Threshold attack, the Logistic Regression (LR) attack, and the Multi-Layer Perceptron (MLP) attack based on [33].
For label-dependent attacks, we use the Gap Attack [43] as the baseline. For label-only augmentation attacks, we follow the original work [7] and use a shallow neural network to train the attack model with 2 hidden layers (10 neurons each). We set batch size as 32 and epochs to 60 to train the attack model.

6.3 Experimental Setup and Evaluation Metrics

Dataset Partitioning. As shown in Table 1, we split CIFAR-10, CIFAR-100, and CH-MNIST into $n = 5$, $n = 4$, and $n = 4$ disjoint subsets. In effect, models trained on individual subsets tend to be less accurate compared to one trained on the original dataset. To address this side-effect and gain back accuracy lost to dataset splitting, we leverage data augmentation [37]. Specifically, we apply horizontal flip, width shift and height shift (by 0.1), 10° rotation, and zoom by 0.2%. We note that for CH-MNIST we did not apply augmentation as we experimentally observed that it results in accuracy degradation instead of accuracy gain. Tables 2 and 3 show that, for CIFAR-10 and CIFAR-100, our models gain an average accuracy of 7.78% and 9.96%, respectively, with data augmentation.

Exclusion Oracles. For MCE, we use the probability vector of each prediction to identify the most confident model. For ESE, we use *SHA-1* hashing from the *hashlib* module of Python. For ASE, we use *phash* perceptual hashing from Python's *imagehash* library with Hamming distance threshold = 14. For CBE, we train a Random Forest classifier based on the dataset split in Table 1. We empirically fix the PCA value of 4. As a result, instead of 3072 features on

CIFAR-10 and CIFAR-100, we use $PCA\{(32, 32 \times 3), component = 3\} = 3{\times}32{\times}3 = 288$ and for CH-MNIST $PCA\{(64, 64), component = 3\} = 3 \times 64 = 192$, principal features per image.

Probability-Dependent Attacks. We use Threshold (Th), Logistic Regression (LR), and Multi-Layer Perceptron (MLP) attacks by Salem et al. [33] as implemented in Tensorflow-Privacy [2].

Label-Dependent Attacks. We use the attack by Choquette-Choo et al. [7] which shares similarities with the attack by Li and Zhang [25]. Specifically, we use Gap attack (their baseline in [7]), Data Augmentation attack, and Decision Boundary Distance attack. For Gap attack, as in the original work [43], we label 0 (non-member) if the target model predicts the wrong class, otherwise we use 1 (member) for correct predictions.

For data augmentation attack, given a data-point, the target model (serving as shadow model) is queried multiple times using augmented images to create a labeled training data for the attack model. We use rotation and translation techniques to issue multiple queries. For rotation of CIFAR-10 samples, we use $r = 4^\circ$ inline with prior work for which $r \in [1^\circ, 15^\circ]$ is established as 'safe' range, and for our settings we found $r = 4$ results in the highest attack model accuracy. As in the original work, we issue 3 queries (with $r = 0^\circ$, $r = 4^\circ$, and $r = -4^\circ$). For CIFAR-100 and CH-MNIST, the best performing r values are 5° and 6° respectively. For translation attack, a pixel bound d is such that, |i|+|j|=d, where we translate the image $\pm(i)$ pixels horizontally and $\pm j$ pixels vertically. For this attack, again following prior work [7], we select translation bound $d = 1$ that yields highest attack model accuracy.

For decision boundary attack, we use the random noise attack and follow a similar setup as the original work [7]: a sample x is predicted as member if the distance $d(x, y) > d_\tau(x, y)$ where $d(x, y)$ is the data-point's l_2 distance from the target model's boundary. To calculate this distance, $d_\tau(x, y)$, we evaluate accuracy of the shadow model (i.e., target model) h on N number of queries where $x^i_{adv} = x + \mathcal{N}(0, \sigma^2.I)$ using isotropic Gaussian noise [10]. We choose optimal number of queries, $N = 250$ based on the highest attack accuracy we obtained after empirically exploring candidate query budget values in $(100, 250, 350, 500)$.

6.4 Evaluation Metrics

We use the following metrics to quantitatively assess the effectiveness of MIAShield:

- **Exclusion Oracle Accuracy** (Acc_{EO}): percentage of correctly excluded samples out of samples submitted to an exclusion oracle.
- **Model Test Accuracy**: percentage of test samples correctly predicted by a model.
- **Generalization Gap**: difference between model accuracy over training samples and test samples.
- **Attack AUC**: Area Under the Curve of attack ROC. The higher the AUC, the more successful the attack.
- **Attack Advantage**: maximum difference between TPR (correctly flagged members) and FPR (non-members flagged as members). It quantifies privacy leakage induced by the attack. The larger its value, the higher the privacy leakage.

6.5 MIAShield Overall Effectiveness

Model Test Accuracy vs. Attack AUC. The ideal utility/privacy trade-off for MIAShield is when Model Test Accuracy remains almost the same as undefended model's accuracy and Attack AUC is close to 50% (near random guess). Compared to the undefended model, across the three datasets and for the three probability-dependent attacks (Figure 2) and four label-dependent attacks (Figure 3), MIAShield reduced MIA AUC to $\approx$ 50% (random guess). This happened with insignificant accuracy loss, as evident from the very narrow horizontal margin between the blue circles (undefended model) and MIAShield exclusion oracles in Figures 2 and 3. Note, however, that MCE (our baseline exclusion oracle) is an exception here. Since the most confident model may not always be the model trained on the target data-point, it is not surprising that MCE results in a comparatively higher utility loss and Attack AUC.

Model Test Accuracy vs. Attack Advantage. On probability-dependent attacks (Appendix C: Figure 12) and label-dependent attacks (Appendix C: Figure 13), compared to the undefended model, for all three datasets MIAShield oracles consistently achieve the best combination of utility (very close to undefended model's accuracy) and attack advantage (very close to zero). These results are consistent with the ones in Figures 2 and 3 and confirm that MIAShield oracles not only result in lowest aggregate attack AUC, but also lowest privacy leakage when measured via attack advantage. Also notice that except the baseline exclusion oracle (MCE), the remaining four exclusion oracles (ESE, ASE, CBE, and COE) are not only significantly better performing but also comparable among each other despite their complementary exclusion intuitions.

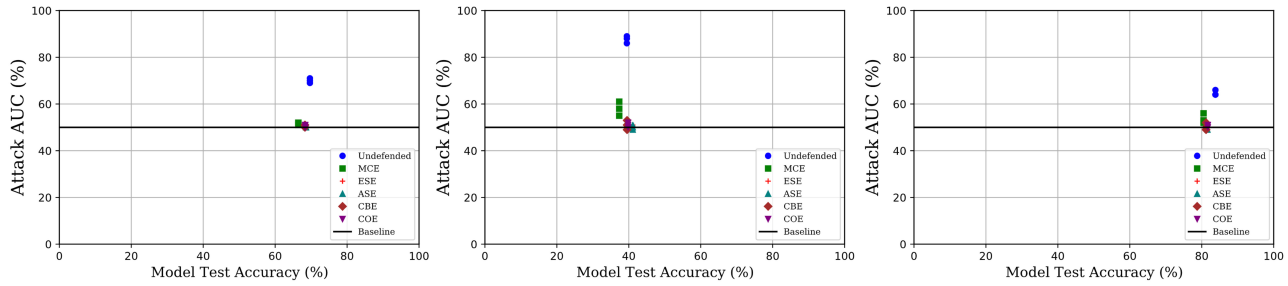
Results across Figures 2, 3, 12, and 13 point us to two high-level insights. First, using the preemptive exclusion strategy, eliminating the membership signal that is attributed to a target data-point and its neighborhood significantly reduces the effectiveness of MIAs with insignificant utility loss. Second, the membership signal does not always stem from the most confident model in the ensemble.

With respect to **RQ1**, overall MIAShield achieves significantly better privacy-utility trade-offs compared to the undefended model. Among the exclusion oracles, ESE, ASE, CBE, and COE offer consistently better privacy-utility trade-offs than the MCE baseline oracle while ASE is the overall winner with CBE and COE oracles ranking close second.

6.6 Comparison with Prior Defenses

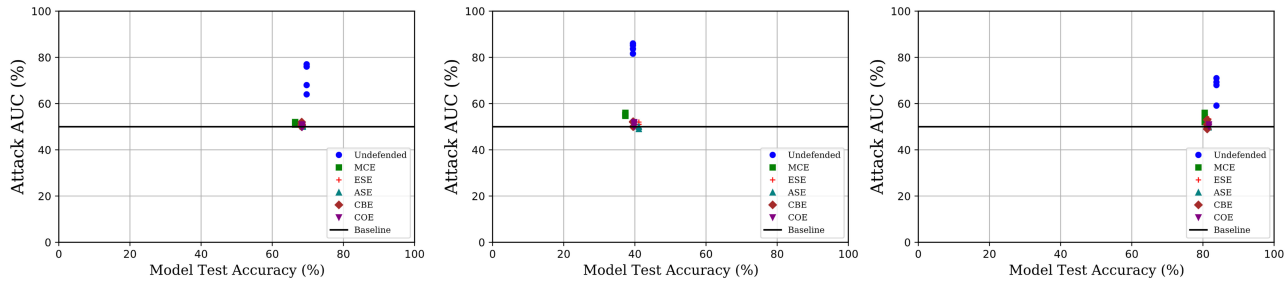
To answer **RQ2**, we compare MIAShield's best version (oracle: ASE) with 5 defenses in 4 categories, published in the 2016–2021 timeframe. Among differential privacy-based defenses, we use DP-SGD [1] and PATE [30]. Among ensemble learning-based methods, we use Model-Stacking [33]. From confidence masking-based defenses, we use MemGuard [18]. Among strong regularization-based defenses, we use MMD+Mixup [24]. Details of each defense's setup including ϵ choice for DP-SGD and PATE appear in Appendix B.

Overall Comparison. Plots of model utility against attack AUC (Figures 4 and 5) and utility against attack advantage (Appendix C: Figures 14 and 15) consistently suggest that MIAShield (green square) offers better privacy-utility trade-off compared with prior defenses that cover four families of techniques. Next, we provide a



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 2: Model Test Accuracy vs. Attack AUC for all MIAShield exclusion oracles against probability-dependent attacks.



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 3: Model Test Accuracy vs. Attack AUC for all MIAShield exclusion oracles against label-dependent attacks.

highlight of pairwise comparison of MIAShield with each defense (much detailed pairwise comparison appears in Appendix D).

MIAShield vs. MemGuard. Overall, MemGuard preserves utility while MIAShield results in negligible utility loss. On probability-dependent attacks, MIAShield offers much lower attack AUC (Figure 4) and attack advantage (Appendix C: Figure 14). On label-dependent attacks, while MemGuard offers nearly zero MIA mitigation, MIAShield drops attack AUC (Figure 5) to almost random guess with much lower attack advantage (Appendix C: Figure 15).

MIAShield vs. Model-Stacking. Overall, MIAShield outperforms Model-Stacking on CIFAR-10 and CIFAR-100 while they are comparable on CH-MNIST. On utility vs. attack advantage, however, MIAShield significantly outperforms Model-Stacking on all datasets and both attack types (Appendix C: Figures 14 and 15).

MIAShield vs. DP-SGD. Across all three datasets and all seven attacks, MIAShield and DP-SGD are equally able to drop attack AUC to ≈ random guess. However, in all cases MIAShield offers orders of magnitude better model utility than DP-SGD. This is evident from the horizontal model utility gaps between green squares (MIAShield) and red crosses (DP-SGD) in Figures 4 and 5, which makes MIAShield an overall winner on utility-privacy trade-off.

Since the choice of the privacy budget ϵ determines privacy/utility trade-off for DP-SGD, for fair comparison, we experimented with ϵ values $10^1 - 5 \times 10^3$ to fix the privacy/utility trade-off. In Figures 6a (CIFAR-10) and 6b (CIFAR-100) we show privacy/utility comparison between DP-SGD with different ϵ choices and MIAShield with the ASE oracle. Our comparison above is based on DP-SGD with $\epsilon = 10^3$ (for both CIFAR-10 and CIFAR-100) because as can be seen from Figure 6a DP-SGD offers comparable attack AUC with MIAShield but offers a much lower utility than MIAShield. We note that the other seemingly obvious choice is DP-SGD with $\epsilon = 5 \times 10^3$, which offers much better utility. However, it does so at a significant cost of privacy (notice increase in Attack AUC for both Figure 6a and 6b). Therefore, in the interest of fair comparison, we chose DP-SGD ($\epsilon = 10^3$) instead because attack AUC is much close to random guess (which in turn is close to MIAShield).

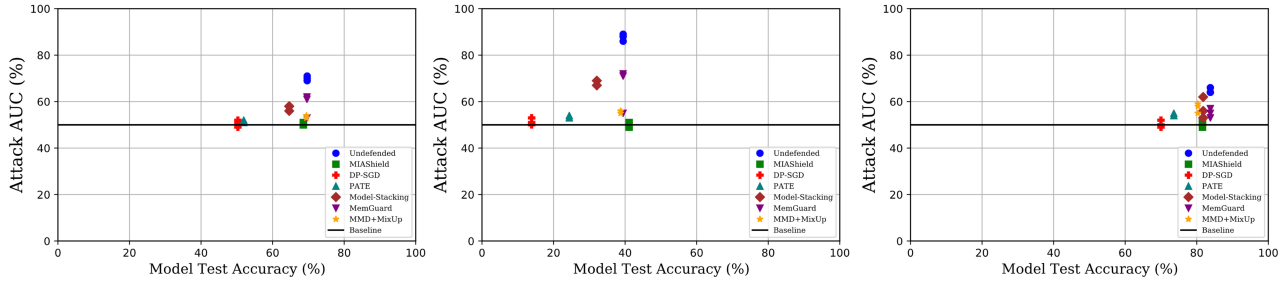
MIAShield vs. PATE. Overall, MIAShield outperforms PATE on both utility against attack AUC and utility against attack advantage. With respect to DP-SGD, however, Figures 4 and 5 consistently suggest that PATE is overall better, but still underperforms MIAShield across the board.

MIAShield vs. MMD-MixUp. For CIFAR-10, MMD-MixUp and MIAShield offer close-enough privacy-utility trade-offs. On CIFAR-100 and CH-MNIST, however, MIAShield outperforms MMD-MixUp with lower attack AUC and attack advantage, again suggesting MIAShield offers overall better utility-privacy trade-off.

With regards to **RQ2**, our results consistently suggest that MIAShield offers much better privacy-utility trade-off than prior defenses across all datasets on both probability-dependent and label-dependent attacks.

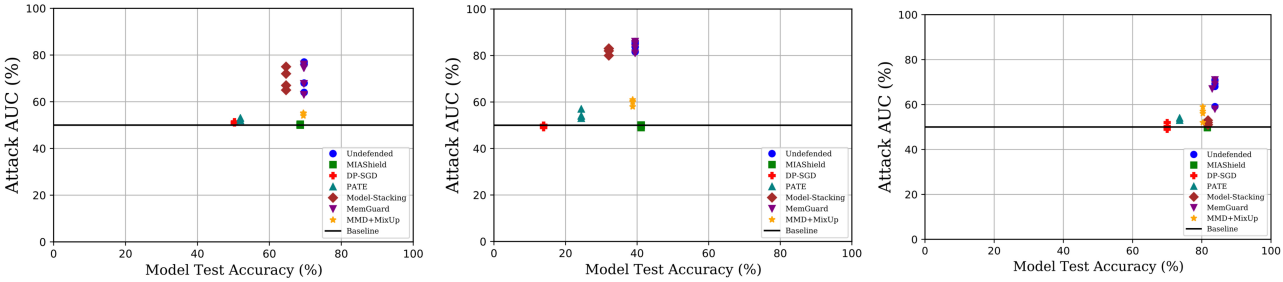
6.7 Robustness Against Adaptive Attacks

Our evaluation so far considered an adversary who either has no knowledge of the internal workings of MIAShield or manipulates inputs within MIAShield detection threshold. To answer **RQ3**, we



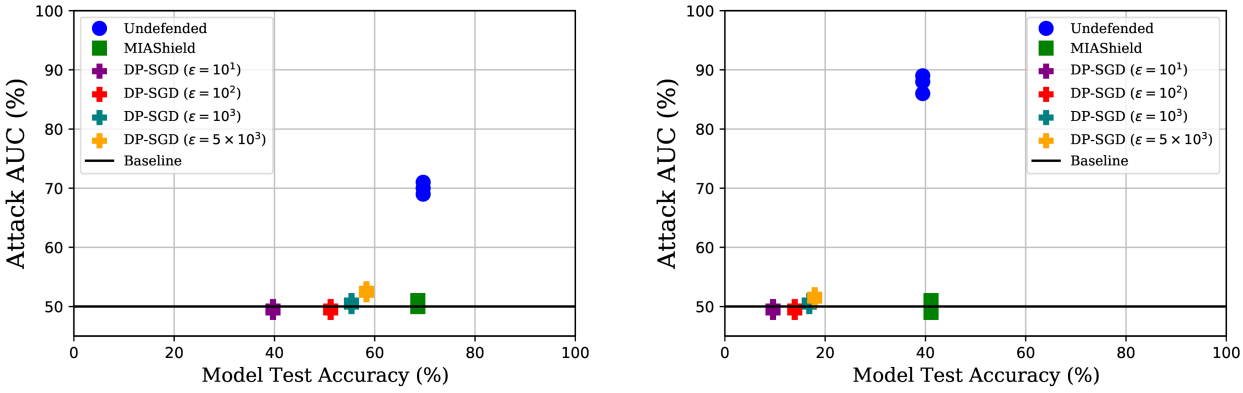
(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 4: MIAShield vs. related work on Model Test Accuracy vs. Attack AUC against probability-dependent attacks.



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

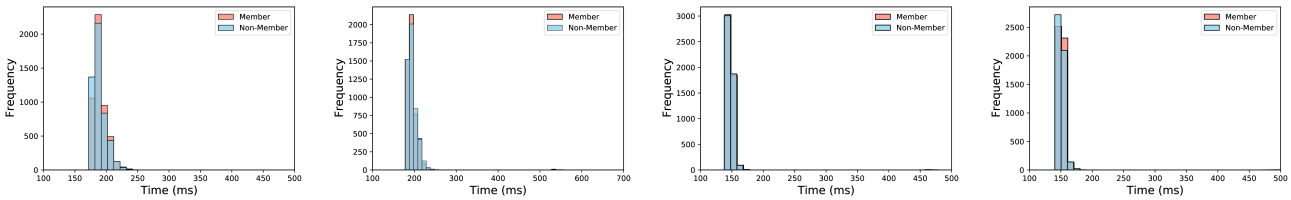
Figure 5: MIAShield vs. related work on Model Test Accuracy vs. Attack AUC against label-dependent attacks.



(a) CIFAR-10 (b) CIFAR-100

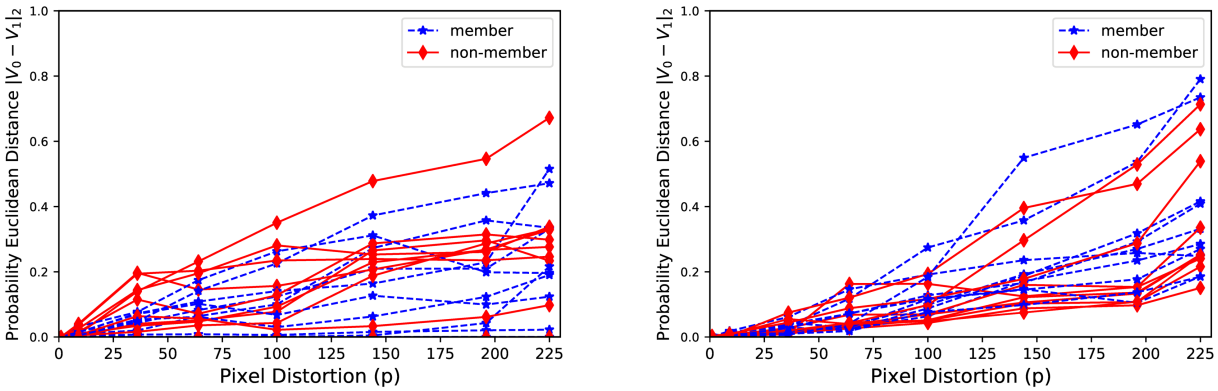
Figure 6: Privacy/Utility trade-off comparison between MIAShield and DP-SGD (over a range of ϵ choices) against probability-dependent attacks for CIFAR-10 and CIFAR-100.

now consider case (d) of attacker capabilities from our threat model described in §3. Adaptive attacks leverage side-channel information and/or knowledge of the exclusion oracles to bypass MIAShield. We consider three types of adaptive attacks. First, a *timing attack* that leverages the fact that MIAShield uses a certain exclusion strategy, but the adversary may be unaware of the type of exclusion oracle used. Second, a *sudden probability change attack* that observes visible change(s) to probability scores of predictions of a progressively manipulated data-point as potential indicators of exclusion. Third, a *manipulative attack* that leverages knowledge about exclusion oracles and parameters (e.g., hamming distance threshold for ASE) and performs batch augmentation or batch perturbation of samples to reduce MIAShield's resilience.



(a) CIFAR-10 (Oracle: ESE) (b) CIFAR-10 (Oracle: ASE) (c) CIFAR-100 (Oracle: ESE) (d) CIFAR-100 (Oracle: ASE)

Figure 7: Timing attack risk comparison between members and non-members.



(a) CIFAR-10 (b) CIFAR-100

Figure 8: Sudden probability change monitoring against incremental pixel distortion for CIFAR-10 and CIFAR-100.

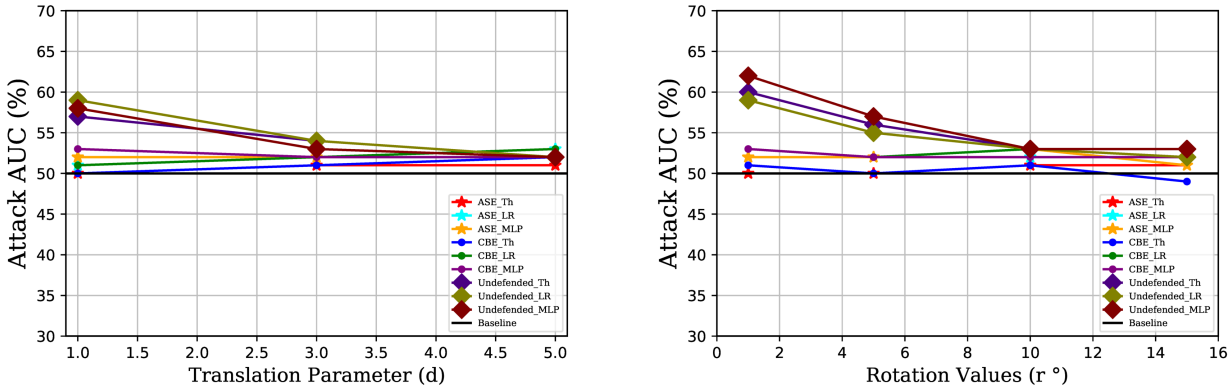
Timing Attack. For this attack, ESE, ASE, and COE are the target oracles. Assuming that one of these oracles is behind the prediction API, the adversary keeps track of prediction response times over a batch of members and non-members and then uses the observations to empirically distinguish members from non-members. The attacker's motivation here is that: with no remedy against timing attacks and assuming no fluctuation in response latency, on average, MIAShield takes longer to respond to non-member samples.

To evaluate the effectiveness of mitigations we introduced in §5.2 (reshuffling and bounded delay on members), Figure 7 compares histogram plots of response time between 5K members and 5K non-members for ESE and ASE oracles on CIFAR-10 and CIFAR-100. *Our results suggest that the adversary is overall unable to distinguish between members (red bars) and non-members (blue bars).*

Sudden Probability Change Attack. The goal of this attack is to incrementally manipulate a data-point and observe output probability vectors in pursuit of sudden change (spike/decrease) in the probability values. The intuition is that such sudden changes in probability would typically happen for members as a potential indication of an exclusion of the model that was trained on the data-point, which would imply inference of a member. To explore the feasibility of such an attack, for a randomly picked member/non-member, we apply Gaussian noise-based incremental pixel distortion. The pixel distortion attack works as follows: for each test sample, we apply incremental manipulation by adding random Gaussian noise to different numbers of pixels of images. For example, $p = 1$ implies we add noise to only one randomly selected pixel, while $p = 100$ means we add noise to 100 randomly chosen pixels.

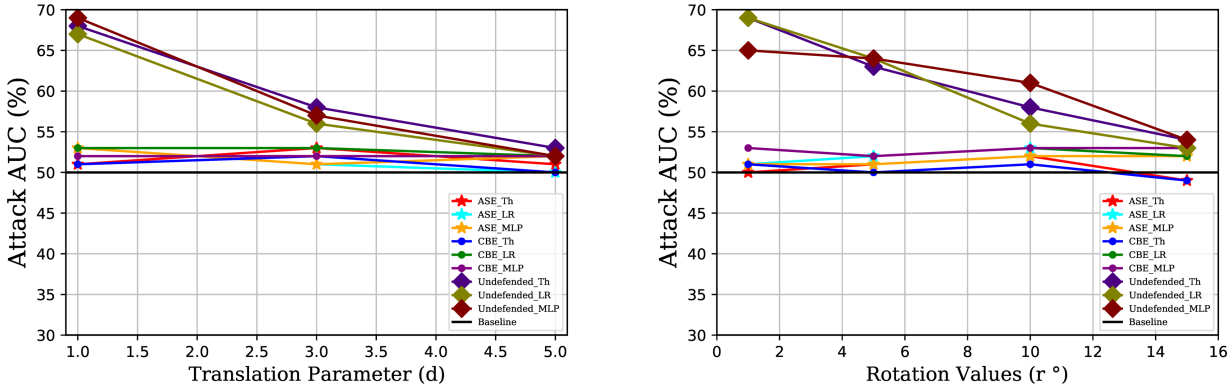
For each incremental iteration of pixel distortion, we compute the *difference* between probability score vector of current and previous manipulation iteration. The goal is to determine whether a "sudden change" in probability scores is observed and hence gives a signal of exclusion of the model that contains the data-point. The results of this experiment are shown in Figures 8a (for CIFAR-10) and 8b (for CIFAR-100). On the x-axis is the pixel distortion value p (the number of randomly selected pixels to which Gaussian noise is added). On the y-axis is the probability score vector distance between iteration $i-1$ and i of pixel distortion. The distance is computed as $||V_i - V_{i-1}||_2$ for V_i and V_{i-1} as the predicted probability vectors of distortion iteration i and $i-1$, respectively. The dimension of V is the same as the number of class labels of the dataset at hand. For CIFAR-10, for both members and non-members we pick one sample at random from each of the 10 classes. Similarly, for CIFAR-100, we randomly pick one sample each from 10 randomly picked distinct classes of the 100 classes.

From Figures 8a and 8b, looking at the y-axis values for both CIFAR-10 and CIFAR-100, for an adversary that is after a "sudden change" in probability scores, the members and non-members line



(a) Translation **(b) Rotation**

Figure 9: Augmentation-based probability-dependent attacks against ASE and CBE oracles on CIFAR-10.



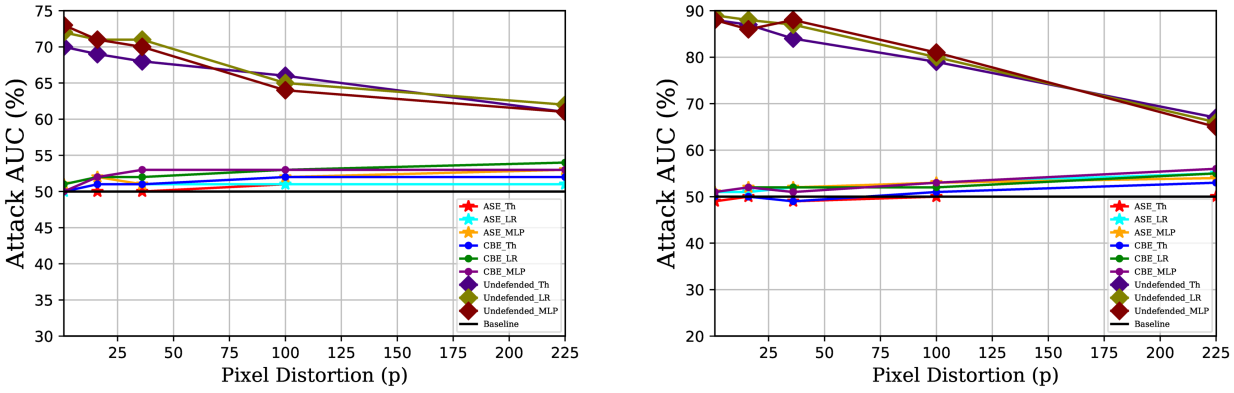
(a) Translation **(b) Rotation**

Figure 10: Augmentation-based probability-dependent attacks against ASE and CBE oracles on CIFAR-100.

plots do not particularly point to a triggering signal to call for inference. For some members/non-members, probability difference increases as distortion intensity increases, but this does not necessarily imply exclusion. Examining first sudden changes of members vs. non-members (i.e., left-most non-zero value for distortion parameter p), it is again not apparent that there is a pattern that is uniquely attributable to members to result in high confidence inference. *The conclusion is that sudden probability change may not necessarily be attributed to members.*

Manipulative Attack. Beyond incremental manipulation of a single sample, we also examine incremental manipulation on a batch of member/non-member data-points to measure the aggregate impact of incremental manipulation on attack AUC. For this purpose, we use translation and rotation per the setup in §6.3, pixel distortion as in the sudden probability change attack, and perturbation as used in label-only boundary-distance attack (§6.3).

Rotation and Translation-based batch manipulation: To push MIAShield beyond the augmentation values set in our experimental setup (§6.3), we use r spanning $1-15$ for rotation and d spanning $1-5$ for translation. Figure 9 (CIFAR-10) and Figure 10 (CIFAR-100) show attack AUC comparison for the three probability-dependent attacks (Th, LR, and MLP) between the undefended model and MIAShield with ASE and CBE oracles. Looking at the undefended model (top 3 lines for both Figures), attack AUC drops as manipulations increase for both rotation and translation. For MIAShield (ASE or CBE oracle), attack AUC remains consistently lower than the undefended model and close to the random guess baseline despite the increased manipulations. Given the augmented training of MIAShield's ensemble, such resilience to augmentation perturbations may seem expected. However, in this experiment we pushed the manipulations far enough and our results suggest that attack AUC does not necessarily improve with more manipulation.



(a) CIFAR-10 **(b) CIFAR-100**

Figure 11: Pixel distortion-based probability-dependent attacks against ASE and CBE oracles on CIFAR-10 and CIFAR-100.

It is worth noting that, on the one hand, as the manipulations increase, the exclusion oracle accuracy kept decreasing. On the other hand, the attack model accuracy kept decreasing because the probability signal returned from the ensemble turned out to be too noisy for the adversary —resulting in low attack AUC. Besides, recall that by splitting the original dataset into disjoint subsets and then leveraging data augmentation when we train MIAShield ensemble, we are already creating a first line of (empirical) defense. When the augmented ensemble labels the highly manipulated samples of the adversary, the probability scores returned are noisy enough to lead to a poorly performing attack model (hence low attack AUC). A key takeaway here is that more manipulation may not necessarily translate to more accurate attack. In fact, it instead results in increase in the percentage of incorrect predictions which seems to misguide the attack model.

Pixel distortion-based batch manipulation: Figure 11 suggests that as pixel distortion increases, attack AUC shows a slight increase, but not too potent to bypass MIAShield. Although the models tend to leak more members on CIFAR-100 for a higher pixel distortion size ($p = 225$), the leakage is still low compared to the undefended model and other related works. As pixel distortion increases, while the exclusion oracles tend to misclassify members as non-members, the attack model, on the other hand, is fed with weak signals that stem from wrong predictions —resulting in attack AUC degradation.

Perturbation-based batch manipulation: Based on label-dependent attacks (via boundary distance attack as described in §6.3), multiple queries (i.e., 250 in our case) are required to initiate the attack. With each query, a small random noise within the limit of $N(0, \sigma^2)$ is added to the image for manipulation. From Appendix E: Table 5, we observe that, for all the three datasets, both the attack AUC and advantage is near the baseline. This suggests that perturbation-based attacks, while they can degrade model utility, are not overall successful against MIAShield. The model utility degradation is inherited by the attack model via incorrect predictions.

With regards to **RQ3**, our results overall suggest that MIAShield is resilient in the face of (i) timing attacks, sudden probability change attacks, and both augmentation-based and perturbation-based manipulative attacks. We also note that increased manipulation does not necessarily translate to increased attack AUC.

7 CONCLUSION AND FUTURE WORK

MIAShield defends ML models against MIAs through a preemptive exclusion of member data-points. By excluding a model that was trained on a member data-point, MIAShield eliminates the strong membership signal the data-point gives away to a MIA adversary. Our extensive evaluations on three image classification datasets, three probability-dependent attacks, four label-dependent attacks, and comparison with five prior defenses consistently suggest that MIAShield significantly reduces MIA accuracy to nearly random guess with insignificant utility loss. We also demonstrate that MIAShield is resilient to a range of adaptive attacks that exploit side-channels and knowledge about the defense to bypass it.

MIAShield faces a limitation that leaves room for further research. Although overall attack AUC remains low for manipulation-based adaptive attacks, it comes at a cost of utility loss because the manipulations force the MIAShield ensemble to return wrong predictions. While wrong predictions result in a less accurate attack model and hinders the membership inference goal, we also recognize the fact that legitimate users may also manipulate inputs and submit to MIAShield prediction API (e.g., to cloak images against facial recognition in the wild [34]). In this context, preserving model utility is crucial for the model to remain useful.

Moreover, MIAShield could be further improved through future work that either hardens the exclusion oracles against deceptive manipulations or explores more rigorous evaluation of the exclusion oracles' sensitivity to larger and arbitrary manipulations.

ACKNOWLEDGMENTS

We are grateful to our shepherd Simon Oya and the anonymous PETS reviewers whose feedback immensely improved this paper.

REFERENCES

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. ACM, 308–318.

[2] Galen Andrew, Steve Chien, , and Nicolas Papernot. 2020. TensorFlow Privacy. *https://github.com/tensorflow/privacy* abs/2002.08570 (2020). arXiv:2002.08570

[3] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium, USENIX Security 2019*. 267–284.

[4] Nicholas Carlini and David A. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. 39–57.

[5] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. 2020. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*. 1277–1294.

[6] Junjie Chen, Wendy Hui Wang, and Xinghua Shi. 2021. Differential Privacy Protection Against Membership Inference Attack on Machine Learning for Genomic Data. In *Biocomputing 2021: Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, January 3-7, 2021*. WorldScientific.

[7] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. 2021. Label-Only Membership Inference Attacks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 1964–1974.

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting Adversarial Attacks With Momentum. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 9185–9193.

[9] Kazuto Fukuchi, Quang Khai Tran, and Jun Sakuma. 2017. Differentially Private Empirical Risk Minimization with Input Perturbation. In *Discovery Science - 20th International Conference, DS 2017, Kyoto, Japan, October 15-17, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10558)*. Springer, 82–90. https://doi.org/10.1007/978-3-319-67786-6_6

[10] Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin D. Cubuk. 2019. Adversarial Examples Are a Natural Consequence of Test Error in Noise. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2280–2289.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. 2012. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* 13 (2012), 723–773.

[13] Qingying Hao, Licheng Luo, Steve T. K. Jan, and Gang Wang. 2021. It's Not What It Looks Like: Manipulating Perceptual Hashing based Applications. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 69–85.

[14] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. 2021. Membership Inference Attacks on Machine Learning: A Survey. *CoRR* abs/2103.07853 (2021).

[15] Ismat Jarin and Birhanu Eshete. 2021. PRICURE: Privacy-Preserving Collaborative Inference in a Multi-Party Setting. In *IWSPA@CODASPY 2021: ACM Workshop on Security and Privacy Analytics, Virtual Event, USA, April 28, 2021*. ACM, 25–35.

[16] Ismat Jarin and Birhanu Eshete. 2022. DP-UTIL: Comprehensive Utility Analysis of Differential Privacy in Machine Learning. In *CODASPY '21: Twelfth ACM Conference on Data and Application Security and Privacy, Baltimore, MD, USA, April 24–27, 2022*. ACM.

[17] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. 2021. Revisiting Membership Inference Under Realistic Assumptions. *Proc. Priv. Enhancing Technol.* 2021, 2 (2021), 348–368.

[18] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM, 259–274.

[19] Kaggle. 2018. Colorectal Histology MNIST. https://www.kaggle.com/kmader/colorectal-histology-mnist

[20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. CIFAR-10 (Canadian Institute for Advanced Research). http://www.cs.toronto.edu/~kriz/cifar.html

[21] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. CIFAR-100 (Canadian Institute for Advanced Research). http://www.cs.toronto.edu/~kriz/cifar.html

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 1106–1114.

[23] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial Machine Learning at Scale. *CoRR* abs/1611.01236 (2016). arXiv:1611.01236

[24] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership Inference Attacks and Defenses in Classification Models. In *CODASPY '21: Eleventh ACM Conference on Data and Application Security and Privacy, Virtual Event, USA, April 26-28, 2021*. ACM, 5–16.

[25] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 880–895.

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *CoRR* abs/1706.06083 (2017). arXiv:1706.06083

[27] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. ACM, 634–646.

[28] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 739–753.

[29] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2020. Improving Deep Learning with Differential Privacy using Gradient Encoding and Denoising. *CoRR* abs/2007.11524 (2020).

[30] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *ICLR 2017*.

[31] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[32] Md. Atiqur Rahman, Tanzila Rahman, Robert Laganière, and Noman Mohammed. 2018. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.* 11, 1 (2018), 61–79.

[33] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

[34] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2020. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*. USENIX Association, 1589–1604.

[35] Virat Shejwalkar and Amir Houmansadr. 2021. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 9549–9557.

[36] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 3–18.

[37] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.

[38] Liwei Song and Prateek Mittal. 2021. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*. USENIX Association, 2615–2632.

[39] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[40] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. 2022. Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture. In *31st USENIX Security Symposium, USENIX Security 2022, August 10-12, 2022*. USENIX Association, Boston, MA, 1433–1450.

[41] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2018. Towards Demystifying Membership Inference Attacks. *CoRR* abs/1807.09173 (2018).

[42] Ziqi Yang, Bin Shao, Bohan Xuan, Ee-Chien Chang, and Fan Zhang. 2020. Defending Model Inversion and Membership Inference Attacks via Prediction Purification. *CoRR* abs/2005.03915 (2020).

[43] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*. IEEE Computer Society, 268–282.

Layer Type	Layer Parameters
	Input $d1 \times d2 \times d3$
Convolution	$48 \times 3 \times 3$ strides=(2, 2), padding = same, activation = ReLU
Max-Pooling	Poolsize= 2×2 strides=(2, 2)
Batch-Normalization	
Convolution	$96 \times 3 \times 3$ strides = (2, 2), padding = same, activation = ReLU
Max-Pooling	Poolsize = 2×2 strides = (2, 2)
Batch-Normalization	
Convolution	$192 \times 3 \times 3$ padding = same, activation = ReLU
Convolution	$192 \times 3 \times 3$ padding = same, activation = ReLU
Convolution	$256 \times 3 \times 3$ padding = same, activation = ReLU
Max-Pooling	Poolsize= 2×2 strides = (2, 2)
Batch-Normalization	
Flatten	
Fully connected, Dropout	512, 0.5
Fully connected, Dropout	256, 0.5
Fully connected	num-classes
Activation	softmax

Table 4: AlexNet model architecture used for all datasets. For CIFAR-10 and CIFAR-100, $d1 \times d2 \times d3 = 32 \times 32 \times 3$ and for CH-MNIST it is $64 \times 64 \times 1$. Values for num-classes is 10 for CIFAR-10, 100 for CIFAR-100, and 8 for CH-MNIST.

APPENDIX

A. Model Architecture

Table 4 shows details of model architecture used for all datasets.

B. Related Defenses Setup

DP-SGD: We use TensorFlow Privacy [2] based on DP-SGD [1] on the same model architecture (Table 4) for the non-private and MIAShield models. The model parameters, i.e, batch size, number of epochs, and learning rate are similar as well. For CIFAR-10, clipping parameter = 1.5, noise multiplier = 0.223, and $\epsilon = 10^3$. For CIFAR-100, noise multiplier = 0.248 and privacy budget $\epsilon = 10^3$. For CH-MNIST, use noise multiplier = 0.52 and $\epsilon = 10^2$.

PATE: Based on the original implementation of PATE [30], we use 40 and 25 teacher models for CIFAR-10 and CIFAR-100 and ϵ values are in the range $[0.01, 10^2]$. For CH-MNIST, we use 10 teacher models while the ϵ range is similar. For a fair comparison with MIAShield, we pick the ϵ and number of teacher models value that offers the best privacy/accuracy trade-off. For probability-dependent attacks, we note that our implementation of PATE returns the top-1 noisy aggregated confidence score that receives the majority vote (confidence score of the final label only). On the contrary, for the label-dependent attacks, a class label is returned that receives a majority vote by the teacher ensemble after noisy aggregation. The model architecture and training parameters are similar to the non-private model for these settings as well.

Model-Stacking: We follow the original implementation by Salem et al. [33]. In particular, for the first layer of the stack, we use two models (the same architecture as Table 4 and a random forest classifier). We train each model with $\frac{|D^{train}|}{2}$ samples. Finally, we train a third logistic regression classifier as a meta-model that uses the outputs of the first two models as a training dataset to produce the final inference. We train two baseline models with disjoint datasets as in the original work, i.e., for the CIFAR-10 and the CIFAR-100, each model is trained with 2.5K samples. For CH-MNIST, each model is trained on 2K samples.

MemGuard: We reuse the implementation by Choquette-Choo et al. [7]. Thus, we use the best performing ϵ value while masking the confidence vectors. For CIFAR-10, CIFAR-100, and CH-MNIST, we use noise parameters as 10^{-3}, 10^{-2}, and 10^{-4}, respectively.

MMD-Mixup: We follow the setup of the original paper [24]. This technique first uses the mix-up data augmentation technique, in which an image is constructed from two training images to mask individual training samples from exposure to inference. Secondly, for training, they use MMD-Regularization [12] to reduce the difference between confidence score distributions between members and non-members, hence MMD regularization is added as a training loss function to achieve this target. MMD score specifically calculates the distance between softmax output of training (member) / validation (non-member) examples in the same class, where this defense aims to minimize the loss.

C. Model Test Accuracy vs. Attack Advantage

Figures 12 and 13 show the utility vs. attack advantage plots of MIAShield exclusion oracles while Figures 14 and 15 show utility vs. attack advantage trade-off comparison between MIAShield and five prior defenses.

D. Detailed Comparison with Related Defenses

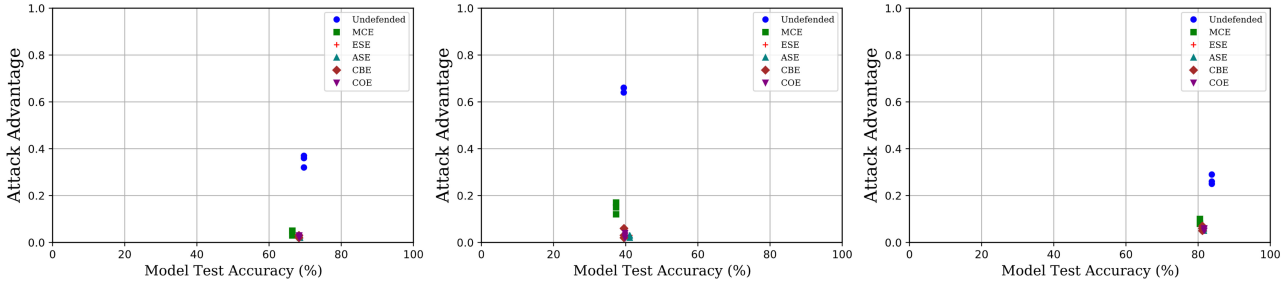
MIAShield vs. MemGuard [18]

Probability-Dependent Attacks: From Figure 4, for CIFAR-10 and CIFAR-100, MemGuard is in the upper right direction compared to MIAShield. Even though MemGuard offers a slightly higher utility compared to MIAShield, it still suffers from high attack AUC. From Figure 14, for CIFAR-10 and CIFAR-100, MemGuard lies in the upper right region while MIAShield stays in the lower-left region near baseline attack AUC, which indicates that MIAShield provides more privacy-utility tradeoffs compared to MemGuard.

Label-Dependent Attacks: From Figure 5, MemGuard is very close to the undefended model, suggesting comparable utility as the undefended model. On attack AUC, however, MemGuard overlaps with the undefended model's attack AUC —indicating that MemGuard offers zero privacy against label-only attacks (note that this is even true for Gap attack, which does not involve manipulation). On the contrary, MIAShield mitigates MIA near random guess. On the x-axis, the distance between MIAShield and MemGuard is very low, which shows that MIAShield offers almost similar utility as MemGuard. From Figure 15, we observe that MemGuard suffers from high privacy leakage compared to MIAShield.

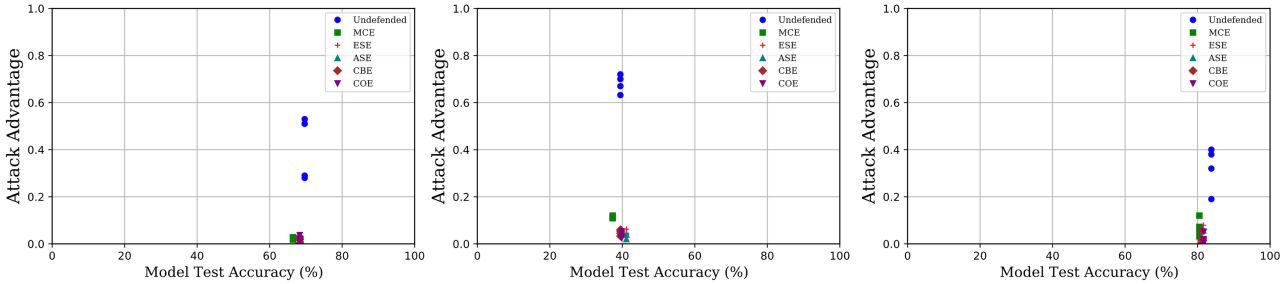
MIAShield vs. Model-Stacking [33]

Probability-Dependent Attacks: From Figure 4, Model-Stacking points lie slightly higher and left compared to MIAShield, which indicate relatively lower privacy-utility trade-off compared to MIAShield.



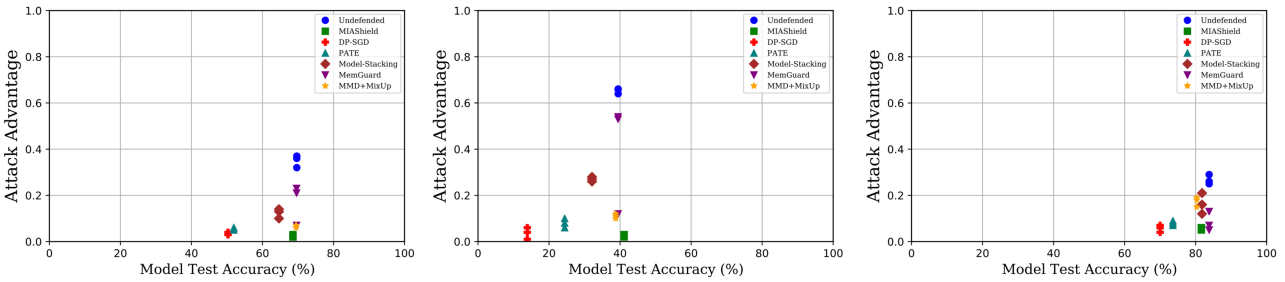
(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 12: Model Test Accuracy vs. Attack Advantage for MIAShield exclusion oracles against probability-dependent attacks.



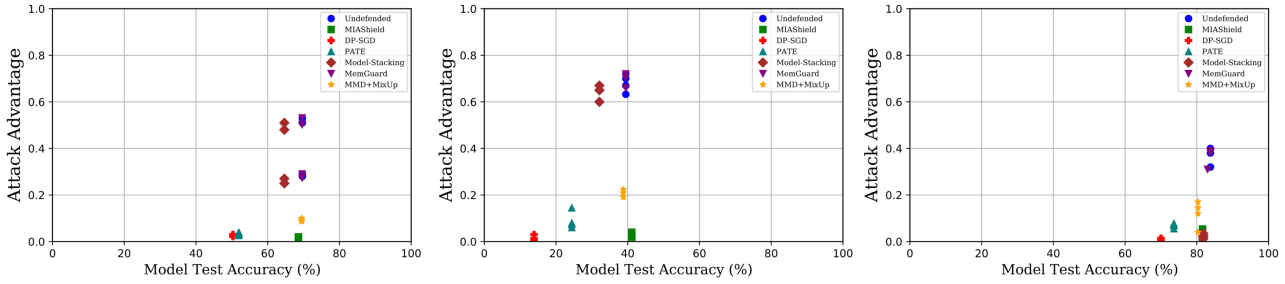
(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 13: Model Test Accuracy vs. Attack Advantage for all MIAShield exclusion oracles against label-dependent attacks.



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 14: MIAShield vs. related work on Model Test Accuracy vs. Attack Advantage against probability-dependent attacks.



(a) CIFAR-10 (b) CIFAR-100 (c) CH-MNIST

Figure 15: MIAShield vs. related work on Model Test Accuracy vs. Attack Advantage against label-dependent attacks.

Model-Stacking also under-performs on attack advantage vs. attack AUC (Figure 14) compared to MIAShield. Given that Model-Stacking aims to conceal membership signals via training two models on subsets of the training set and then a meta-model is trained based on the output of two models. If the original dataset is overfitted, the approach is by design vulnerable to MIA. In addition, splitting the dataset into subsets also results in accuracy loss, unless measures such as data augmentation are taken.

Label-Dependent Attacks: From Figure 5, for CIFAR-10 and CIFAR-100, Model-Stacking points lie upper left compared to MIAShield. For both datasets, MIAShield offers better privacy-utility trade-offs than Model-Stacking. For CH-MNIST, the points overlap —showing comparable privacy-utility trade-off for both MIAShield and Model-Stacking. Similarly, Figure 15 shows that Model-Stacking results in higher attack advantage and higher attack AUC as opposed to MIAShield which shows way lower on both.

MIAShield vs. DP-SGD [1]

Probability-Dependent Attacks: From Figure 4, for CIFAR-10, CIFAR-100, CH-MNIST, MIAShield and DP-SGD reduce attack AUC from $\approx 71\%$ (CIFAR-10) and 89% (CIFAR-100) to $\approx 50\%$ (random guess). On test accuracy, however, DP-SGD results in accuracy loss of $\approx 20\%$, $\approx 25\%$, and $\approx 14\%$, while MIAShield incurs orders of magnitude lower accuracy loss of $\approx 1\%$, $\approx -1.5\%$ and $\approx 2\%$, on CIFAR-10, CIFAR-100, CH-MNIST, respectively. For nearly the same attack AUC performance as DP-SGD, MIAShield introduces $\approx 19\%$, $\approx$ 26.5%, and $\approx$ 14% less utility loss on CIFAR-10, CIFAR-100, and CH-MNIST, respectively. DP-SGD [1] provides strong privacy-utility tradeoffs against MIAs but at the expense of model utility. The remarkably low utility loss in MIAShield stems from the ensemble of disjoint subsets and the use of data augmentation to gain back accuracy loss when splitting the original dataset into disjoint subsets. From Figure 14, MIAShield and DP-SGD provide strong privacy as they both achieve the lowest AUC and attack advantage. For CH-MNIST, attack advantage is comparatively higher for both methods ($\approx$ 0.05). *Label-Dependent Attacks:* From Figures 5 and 15, we see that MIAShield and DP-SGD compare the same as in probability-dependent attacks.

MIAShield vs. PATE [30]

Probability-Dependent Attacks: PATE suffers from larger attack advantage compared to DP-SGD and MIAShield over all datasets (attack advantage is in the range $0.06 - 0.1$ for datasets and attacks). Though it provides less accuracy loss compared to DP-SGD (within $\approx 10\%$ to $\approx 15\%$), MIAShield still outperforms PATE both in terms of privacy and utility loss. In Figure 4, although PATE's attack AUC is near baseline, it is way below MIAShield on test accuracy. In Figure 14, PATE is slightly on the upper right side compared to MIAShield, which implies more privacy leakage for PATE.

Label-Dependent Attacks: In Figure 5, PATE shows higher attack AUC and attack advantage compared to MIAShield. Though PATE provides noisy vote counts as the final label, it still reveals membership signals as it does not exclude any vulnerable model as MIAShield does. Hence, unlike MIAShield, the teacher model that overfits training samples still participates in the noisy vote. Besides, CH-MNIST suffers from a slightly higher privacy leakage compared to the other two datasets due to the limited dataset size which results in a smaller number of teacher models. In Figure 15, we find that PATE offers lower MIA resilience compared to MIAShield.

MIAShield vs. MMD-MixUp [24]

Probability-Dependent Attacks: From Figure 4, for CIFAR-10, MMD-Mixup is close to MIAShield, which suggests that both defenses offer nearly the same privacy-utility trade-offs. On the contrary, for CIFAR-100 and CH-MNIST, MMD-MixUp offers close-enough utility as MIAShield but offers less MIA mitigation due to its higher attack AUC. Figure 14 also suggests MMD-MixUp allows relatively larger attack advantage and attack AUC, especially on CIFAR-100 and CH-MNIST.

Label-Dependent Attacks: As can be seen from Figure 5, for CIFAR-10, MMD-Mixup is once again close-enough to MIAShield suggesting that MIAShield and MMD-MixUp both offer comparable privacy-utility trade-off (though MMD-MixUp shows a bit higher attack AUC). However, for CIFAR-100 and CH-MNIST, MMD-MixUp offers a comparable utility akin to MIAShield, although its attack AUC is higher than MIAShield (especially for rotation, translation, and boundary distance attacks). The attack AUC vs. attack advantage plot (Figure 15) points to the same conclusion as the probability-dependent attacks.

EO Type	Dataset	Mnp.	EO Acc.	Test Acc	Train Acc.	Attack Type	Attack AUC	Attack Adv.
Undefended	CIFAR-10	0, 4, 1, .961	*None*	69.66	98.87	GAP,RA,TA,BA	.640, .760, .770, .680	.280, .510, .530, .290
MCE	CIFAR-10	0, 4, 1, .546	36.24	66.44	69.30	GAP,RA,TA,BA	.510, .510, .520, .510	.021, .018, .028, .026
ESE	CIFAR-10	0, 4, 1, .961	99.99	68.59	67.90	GAP,RA,TA,BA	.504, .503, .501, .503	.008, .027, .021, .021
ASE	CIFAR-10	0, 4, 1, .961	100	68.59	67.90	GAP,RA,TA,BA	.504, .500, .501, .500	.008, .014, .012, .020
CBE	CIFAR-10	0, 4, 1, .961	99.23	68.28	67.52	GAP,RA,TA,BA	.500, .505, .510, .520	.002, .021, .019, .030
COE	CIFAR-10	0, 4, 1, .961	99.93	68.38	67.52	GAP,RA,TA,BA	.495, .502, .505, .510	.007, .018, .020, .037
Undefended	CIFAR-100	0, 5, 1, .996	*None*	39.46	97.98	GAP,RA,TA,BA	.816, .852, .860, .837	.632, .700, .720, .670
MCE	CIFAR-100	0, 5, 1, .996	40.26	37.34	50.20	GAP,RA,TA,BA	.550, .560, .551, .547	.110, .113, .120, .109
ESE	CIFAR-100	0, 5, 1, .996	100	41.12	39.93	GAP,RA,TA,BA	.490, .520, 510, .510	.040, .045, .032, .063
ASE	CIFAR-100	0, 5, 1, .996	99.99	41.12	39.93	GAP,RA,TA,BA	.490, .501, .500, .500	.040, .020, .020, .039
CBE	CIFAR-100	0, 5, 1, .996	99.02	39.52	37.90	GAP,RA,TA,BA	.521, .520, .522, .510	.033, .045, .054, .060
COE	CIFAR-100	0, 5, 1, .996	98.95	39.82	39.93	GAP,RA,TA,BA	.510, .517, .520, .500	.030, .024, .041, .053
Undefended	CH-MNIST	0, 6, 1, .984	*None*	83.80	99.60	GAP, RA, TA,BA	.591, .710, .680, .693	.190, .400, .320, .380
MCE	CH-MNIST	0, 6, 1, .984	40	80.50	85.80	GAP, RA, TA,BA	.530, .520, .540, .560	.059, .032, .072, .120
ESE	CH-MNIST	0, 6, 1, .984	100	81.60	81.05	GAP, RA, TA,BA	.501, .500, .505, .534	.003, .003, .017, .079
ASE	CH-MNIST	0, 6, 1, .984	100	81.60	81.05	GAP, RA, TA,BA	.501, .497, .500, .521	.003, .003, .012, .057
CBE	CH-MNIST	0, 6, 1, .984	99.65	81.20	81.00	GAP, RA, TA,BA	.500, .510, .490, .530	.005, .015, .020, .056
COE	CH-MNIST	0, 6, 1, .984	99.89	81.70	81.00	GAP, RA, TA,BA	.500, .510, .500, .510	.004, .013, .020, .052

Table 5: MIAShield against label-dependent attacks.