Witnessing Erosion of Membership Inference Defenses: Understanding Effects of Data Drift in Membership Privacy

Seung Ho Na
KAIST
Daejeon, South Korea
harry.na@kaist.ac.kr

Kwanwoo Kim
KAIST
Daejeon, South Korea
kw2128@kaist.ac.kr

Seungwon Shin
KAIST
Daejeon, South Korea
claude@kaist.ac.kr

ABSTRACT

Data drift is the phenomenon when the input data distribution in testing time is different from the training time. This strengthens the generalization gap in a model, which is known to severely deteriorate the model's performance. Meanwhile, previous studies state that membership inference attacks (MIA) take advantage of the generalization gap of a machine learning model. By transitive logic, we can deduce that data drift would affect these privacy attacks. In this work, we consider data drift when applied to the privacy threat of MIA. As the first work to explore the detrimental extent of data drift on membership privacy, we conduct a literature review on current MIA defense works under selected dimensions associated with data drift. Our study reveals that not only has data drift never been tested in MIA defense, but there is also no infrastructure to juxtapose data drift with MIA defense. We overcome this by proposing a design for simulating authentic and synthetic data drift and evaluate the benchmark MIA defense methods on various settings. The evaluation shows that data drift strongly enhances the attack success rate of MIA, regardless of defense. In this, we propose MIAdapt, a proof of concept of a MIA defense that allows update in data drift. From this evaluation, we provide security insight into possible solutions in negating the effects of data drift. We hope our work brings attention to the threat of data drift and instigates the development of MIA defense that are adaptable to data drift.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; • **Computing methodologies → Machine learning**.

KEYWORDS

Membership Inference Defense, Data Drift

ACM Reference Format:
Seung Ho Na, Kwanwoo Kim, and Seungwon Shin. 2023. Witnessing Erosion of Membership Inference Defenses: Understanding Effects of Data Drift in Membership Privacy. In *The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), October 16–18, 2023, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3607199.3607224

RAID '23, Oct 16–18, 2023, Hong Kong, China

ACM ISBN 979-8-4007-0765-0/23/10…$15.00
https://doi.org/10.1145/3607199.3607224

1 INTRODUCTION

Times change the behavior of humans and their data, i.e., the input data distribution. This phenomenon called data drift is when data distribution during the training phase is different from testing phase [2, 35]. A real world example of data drift is the immense effect COVID-19 has had on human behavior [33]. This shift in data distribution leads to significant performance degradation in machine learning models [9]. An example is Instacart's online grocery recommendation model [7], where the accuracy dropped from 93% to 61% since COVID-19. This common problem of data drift is caused by the "closed-world" assumption of machine learning that assumes the training and testing data distribution are similar. This assumption is too strong in the real world, inducing notable *generalization gap* in the machine learning model.

In the field of membership inference attacks, generalization gap is known to be one of the leading causes of a deep learning model's vulnerability to membership inference [3, 26, 50, 60]. Membership inference attacks (MIA) are a type of privacy attack on deep learning models that extract the information of whether a data piece was used in training [48]. The inference of membership information is severe threat to privacy because they allow the deduction of personal information based on the model; membership information of a data point in a machine learning model for drug dose prediction leaks information of the data owner's state of health. Noting that data drift can cause generalization gap, by transitive logic, we can deduce that data drift may aid in the performance of MIA. Because data drift is a common and inevitable observation that occurs in the wild, this would escalate the potential threat of MIA.

Meanwhile, MIA defense methods have been studied throughout past literature. The MIA defenses can be categorized by the time they are applied. The defense methods that are applied during training aim for the improved generalization of the model [20, 27, 27, 38, 45, 48, 50]. The idea is that a well generalized model will adapt properly to unseen data; it will have similar performance and behavior on training and testing data, and without any difference in model behavior, MIA can be prevented. MIA defenses that occur after training converge to the concept of training an unprotected model first and then concealing the original prediction vector of this model from the attacker. These methods include logit masking [5, 21, 28, 45, 48] where the logits are altered following a certain rule or optimization algorithm, or they use the trained unprotected model as a reference to smooth out vulnerable features and rebuild a protected model [16, 46, 56].

Data drift may promote MIA, but only after studying its effect on MIA defense methods can we recognize data drift as an established threat to membership privacy. Unfortunately, there is no literature that has addressed this issue of data drift in MIA settings. All the

datasets used in evaluation are randomized datasets that are divided into training and test datasets by random selection from a single pool of data; the training and test datasets are from the same data distribution and display no aspect of data drift. Not only has data drift never been studied with MIA defense, but there is also no plane in which the two can be juxtaposed; there are no suitable datasets to perform evaluation. The datasets used in data drift work primarily consist of stream data [30] and thus are incompatible with the privacy-centric tasks and data formats of MIA settings. This obstacle bears another issue of there being no general method to transform a randomized dataset into a drifted dataset. To properly address data drift in MIA defense, an acceptable MIA-friendly dataset with data drift must be obtained, and to obtain a drifted dataset, a means of implementing data drift in a controllable form must be acquired.

In this work, we acknowledge the potential threat of MIA enhanced by data drift, and confirm the effects of data drift on MIA defense. The first part of this work systematically reviews MIA defense on three critical dimensions that concern data drift. These dimensions are *generalizability*—can the method reduce the effect of data drift, *adaptability*—can the method be updated to new distributions, and *nonrandomization*—did the method test in data drift conditions. From this analysis of MIA defense literature, we learn that *none of the previous MIA defense literature considered data drift.* The second part of this work organizes a detailed evaluation of MIA defenses when exposed to data drift. We provide a data drift design that generates *authentic* and *synthetic* data drift when given a randomized dataset to satisfy the lack of infrastructure for data drift in MIA settings. Our evaluation shows that when the degree of data drift is increased, the attack success rate of MIA increases in all benchmark MIA defense methods, independent to MIA defense. This shows that data drift is capable of penetrating MIA defense methods, and therefore reduces the load of an attacker; all the attacker needs to do is prepare a drifted dataset. Data drift poses as a severe threat of membership privacy in deep learning models. To this end, we offer a MIAdapt, a proof of concept solution that allows updating MIA defense to adapt to data drift. Compared to the benchmark MIA defense methods, MIAdapt best improves the membership privacy of an AI model.

This paper has the following contributions:

- We are the first to question the effect of data drift in MIA, and our analysis alerts the presence of data drift being a potent threat to membership privacy in deep learning models.
- We thoroughly study the past MIA defense literature scrutinized by data-drift-related dimensions, and offer findings contributing to the implementation of data drift.
- We provide a method for generating drifted datasets from randomized datasets and introduce controllable parameters representing the degree of data drift in both authentic and synthetic data drift generation.
- Our evaluations show that data drift penetrate all benchmark MIA defenses in extensive evaluations of various attack methods and datasets. Accordingly, a defense strategy called MIAdapt is crafted to mitigate data drift vulnerabilities.

Our code of data drift generation and evaluation is released here[1] to support future research and reproducibility.

[1]https://doi.org/10.5281/zenodo.6778830

2 BACKGROUND AND RELATED WORK

2.1 Membership Inference Attack

Membership inference attacks (MIA) are a type of privacy attack that takes advantage of the fact that machine learning models are prone to memorizing the training data information [44]. Specifically, MIA intend on inferring the participation of a data point in the training of a machine learning model [48]. MIA can be formally described as given a target data sample x, target model $\mathcal{M}$ trained on $\mathcal{D}_{train}$, and auxiliary dataset $\mathcal{D}_{aux}$:

$$MIA(x, \mathcal{M}, \mathcal{D}_{aux}) = \begin{cases} 1 & \text{when } x \in \mathcal{D}_{train} \\ 0 & \text{when } x \notin \mathcal{D}_{train} \end{cases} \quad (1)$$

Despite the requirement of both x and $\mathcal{M}$ in Equation 1, MIA can be carried out in black-box settings as well, in which the attacker uses the model's output prediction vector $\mathcal{M}(x)$ and $\mathcal{D}_{aux}$ to carry out MIA.

Depending on the construction of the attack model, MIA can be classified into two major approaches of using binary classifier optimization or metric-based thresholding. The main idea of binary classifier optimization methods is to train a neural-network based binary classifier that performs membership inference [48]. These works first train a shadow classifier model that mimics the behavior of the target model on a shadow dataset that is constructed to be from the same distribution as the original train data. By querying data points to this shadow model, the posteriors can be labelled by their respective membership status, and this dataset can be used to train an attack model that successfully infers membership in black-box settings [45]. Variations in training algorithm include utilization of class labels for input features [38] and incorporation of latent features as well as the posteriors for membership inference in the case of white-box settings [39].

Metric-based thresholding methods make membership inferences based on the calculated metric of the prediction vector. By preparing a shadow dataset, these methods query the target model for the prediction vectors of the train and test data points. From these prediction vectors, performance metrics are calculated, where the performance metric $f_{metric} : R^n \rightarrow R$ is a scalar function that maps a logit vector to a rational value. The optimal threshold value that best differentiates train and test data points can be determined for the shadow dataset, which in turn is used for MIA on the target dataset [26, 51]. Compared to the methods using binary classifier optimization, metric-based thresholding methods do not require the relatively computation-heavy and time-costly procedure of optimization, and is therefore a lighter means of MIA [18]. Meanwhile, Song et. al. [50] claims that binary classifier optimization methods are insufficient in proper assessment of membership inference risk and that metric-based thresholding methods show improved MIA performance. Therefore, we select four metric-based thresholding methods for our target attacks in evaluation, described in Section 4.2.

Why are MIA feasible? The essence of MIA originates from a model's capability to memorize train data. Many papers have formally and empirically discussed overfitting in classifier training to be a factor contributing to successful MIA [3, 26, 50, 60]. Deep learning models are composed of many parameters that oftentimes provide excessive model complexity for the prepared train

dataset size [54]. Overfitting in a model is the state of the model that learns the manifold specific to the train dataset and deteriorates the model's generalization; the model will show a *generalization gap*, which is the contrasting performance in the train and test dataset. Tightly associated with overfitting, generalization gap and its correlation to MIA performance has also been studied [45, 51]. In the end, MIA takes advantage of a model when it shows different behaviors in train and test data.

2.2 Data Drift

Data drift, also called dataset shift, is the concept of a mismatch between training and test data distributions, i.e., a situation in which a model's input distribution changes in testing time [2, 35]. The common assumption of machine learning is a closed-world assumption, where the train data and test data are from the same distribution. But in real-world applications and scenarios, this foundation of machine learning is often violated [31]. Data drift is the observation of this inconsistency and can be formally defined as such:

Definition 1 (Data Drift)**.** For data point $x \in \mathcal{X}$ and its class label $y \in \mathcal{Y}$, data drift is the condition when $P_{tr}(x, y) \neq P_{tst}(x, y)$.

P_{tr} refers to the data distribution at the training time, while P_{tst} refers to the data distribution at the testing time (i.e. time of model deployment). Data drift is defined in terms of joint distributions and leaves it as a general concept, allowing freedom and flexibility in what can be referred to as data drift. From this definition, data drift can be classified into two broad categories of covariate shift and concept drift [31, 35].

Definition 2 (Covariate Shift [24, 47, 58])**.** Given a data point $x \in \mathcal{X}$ that causally determines its class label $y \in \mathcal{Y}$, covariate shift is the case where $P_{tr}(y|x) = P_{tst}(y|x)$ and $P_{tr}(x) \neq P_{tst}(x)$.

Definition 3 (Concept Drift [12, 57, 58])**.** Given a data point $x \in \mathcal{X}$ that causally determines its class label $y \in \mathcal{Y}$, concept drift is the case where $P_{tr}(y|x) \neq P_{tst}(y|x)$ and $P_{tr}(x) = P_{tst}(x)$.

Covariate shift denotes the case where the data distribution (distribution at train P_{tr} and test P_{tst}) changes over time, but their assigned labels remain the same. An example of covariate shift is the changing appearance of trees by season. A tree in springtime will have green leaves while a tree in wintertime will have none; but nonetheless, both are trees. Concept drift refers to the case where a data piece's label changes over time. An example of concept drift might be a specific indicator of a binary file being malicious in the past, but now considered benign.

Data drift is recognized as a major problem hindering modern day machine learning applications. In data drift, the deviated input distribution will no longer abide to the trained decision boundary, causing performance degradation issues [9]. As critical as it is, data drift and its effect is a commonly occurring phenomenon in the real-world. According to the Organisation for Economic Co-operation and Development (OECD), machine learning models predicting global air passenger volumes failed to properly perform in the COVID-19 era [42]. Concept drift in the security venue was also an issue affecting tasks such as malware classification [1, 4, 29] and intrusion detection [32].

Relevance to MIA: Note that data drift caused this erroneous behavior in well-trained models. Due to the test data distribution (e.g., post-COVID-19) being distant from the train data distribution (e.g., pre-COVID-19), a large generalization gap was revealed, depicted in this faulty behavior. Because generalization gap describes the different interpretation of the data by a model in train and testing time, it can be described formally by $P_{tr}(\mathcal{M}(x), y)) \neq P_{tst}(\mathcal{M}(x), y))$. Overfitting and generalization gap was studied to be the main cause of MIA so it can be suggested that *data drift conditions could lead to a greater vulnerability against MIA.*

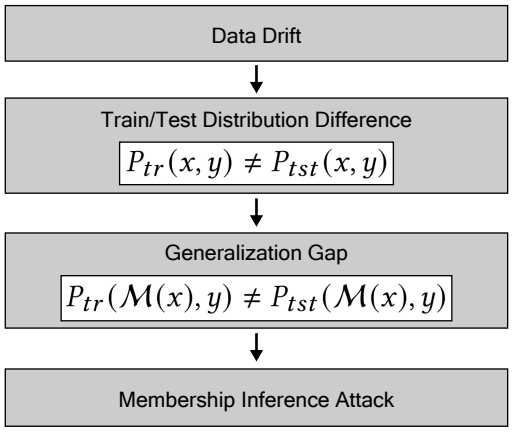


Figure 1: How data drift affects membership inference attacks. Data drift scenarios will lead to a generalization gap in the model, making it prone to membership inference attacks.

The relevance of data drift and MIA is portrayed in Figure 1. Note that the second arrow between "Train/Test Distribution Difference" and "Generalization Gap" is based on an empirical observation of the symptoms of data drift on the machine learning models mentioned above. In this work, we intend on examining this relation between data drift and membership inference and understand the efficacy of MIA defense methods in data drift conditions by covariate shift.

3 DATA DRIFT IN MEMBERSHIP INFERENCE DEFENSE

Due to the fact that data drift may increase the dangers of MIA, it is essential to consider them in the presence of MIA defense methods; if the effect of data drift is potent enough to penetrate MIA prevention schemes, data drift can be deemed a critical vulnerability threatening the private membership information of models. To understand the effect of data drift in MIA defense, we breakdown the past MIA defense literature by three dimensions in consideration of data drift: *generalizability*, *adaptability*, and *nonrandomization*. Each of these dimensions answers the questions of:

(1) Generalizability: Does the defense have means to reduce the effect of data drift?
(2) Adaptability: Can the defense be updated for data drift?
(3) Nonrandomization: Does the evaluation contain any drifted datasets?

3.1 Dimensions in Scrutiny

3.1.1 ***Generalizability*** *of defense models.* A well-generalized model is a model that delivers the same performance in the test

dataset as the train dataset, even when minimizing the training error [40]. Evidently, the generalization gap would be minimal. An ideally generalized model would maintain a low generalization gap ($P_{tr}(\mathcal{M}(x), y) \sim P_{tst}(\mathcal{M}(x), y)$) when the data drifts at a small degree. In this sense, generalization of the model helps dilute the effect of data drift in the model.

Many MIA defense methods provide generalization to their models by using regularizations. Regularizations are the additory methods that make solutions simpler to avoid overfitting, which are typically penalty/complexity terms that are added in the optimization process to prevent overfitting and improve generalization [34, 55]. Classic regularizations, although not meant for MIA defense, were shown to be effective in previous works [20, 27, 45, 48, 50]. These regularizations include L2-norm [41], dropout [52], model stacking [45], early stopping [43, 50], and label smoothing [36].

On the other hand, MIA-specific regularization methods were also developed, with the main purpose of preventing MIA. Adversarial regularization [38] combined the original training loss with the membership inference gain of a dummy attack model. Similar to the training of GAN architecture [10], the target model and the dummy attack model are trained simultaneously in turns. This allows the target model to be regularized by training in the direction of misclassification of the attack model. MMD+Mixup [27] adds the Maximum Mean Discrepancy (MMD) between train and non-train data as a regularization term. The MMD is a kernel based statistical test that measures the distance between two distributions [11].

By retaining regularization in the MIA defense, the target model can be expected to be less affected by data drift. In this aspect, we look into the regularization whereabouts and choice of regularization technique.

3.1.2 ***Adaptability*** *of defense models.* Data drift is a temporal phenomenon which allows reactive measures. Accordingly, a popular approach in overcoming data drift involves the update of models in real-time so that the model is trained continuously on the *current* data, called online learning [17]. However, online learning is applicable only on stream data, which is not the environment used for MIA evaluation nor is the task of membership inference relevant to stream data. Unless a model is retrained from scratch, data drift will affect the model and the model will be vulnerable to MIA for the time being. Because retraining a model is expensive in many aspects, frequent retraining of a model is difficult.

MIA defense, however, does not have to occur simultaneously with model training and can be applied after model training. Only when a defense is applied post-model-training can there be a way to address the condition of data drift into the defense; when the defense is applied during the training of the model, data drift cannot be reflected into the defense. Therefore, the time in which defense is applied with respect to model training is a feature of interest.

The information source that MIA take advantage of are the model weights [37], so defenses strive to obscure the information that can be extracted from the model weights. Defenses that are applied after training can be in the form of logit masking [5, 21, 28, 45, 48] to mitigate black-box attacks. The main idea of logit masking is to hide the full posterior probabilities of a model $\mathcal{M}(x)$ from the attacker. Versions of logit masking include dropping all information except the top-k confidence scores [48] and only returning the label of the prediction [5]. MemGuard [21] takes a different approach and optimizes noise n specific to each data point to maintain the prediction of the target model but be undetected by a neural-net-based attack model. Hanzlik et al. [13] deploys ML models to Isolated Execution Environments and applies noise calculated from the normalized entropy of the logit vector.

In addition to logit masking, other methods of MIA defense that are applied post-training are weight pruning [56] and knowledge transfer [46]. Both methods base their MIA protection by first training an unprotected model. Wang et al. [56] optimizes on objectives representing privacy and efficiency to find a subnetwork from the previously trained target classifier. Shejwalkar et al. [46] uses the unprotected model to filter reference data showing minimal privacy leakage, and leverages knowledge distillation [16] and the reference data to train a protected model.

By having the defense applied post-training, it allows the defense method to consider the aspects of data drift in MIA defense.

3.1.3 ***Nonrandomization*** *of datasets.* The main feature of datasets that are looked into are how the datasets were constructed and divided into training and testing datasets. If the train dataset and test dataset were distinguished by a certain rule or pattern (e.g., different time periods), we can consider the train and test datasets to belong to different distributions and therefore contain data drift. On the other hand, when datasets are divided by random selection from a single set into train and test, these datasets can be considered to be from the same distribution when the sample size is large by the central limit theorem. Randomization in dataset construction will not contain data drift in the resulting test dataset; nonrandomized datasets can be expected to contain data drift.

Every MIA defense will eventually be evaluated on staple datasets used in previous MIA works. If the defense methods were evaluated on datasets with aspects of data drift, the results will demonstrate the defense's robustness to data drift. The two main tasks of models attacked by MIA are classification of image and tabular data. From the open-source image and tabular datasets used, most come either prepared into train/test datasets beforehand, or come in single datasets where the security practitioner decides the division into train/test regardless of data type.

Some examples of datasets used in MIA presented in train/test versions are CIFAR-10 [25], CIFAR-100 [25], and Purchases[2]. CIFAR-10 and CIFAR-100 are composed of images of objects and organisms belonging to 10 and 100 classes, respectively. These datasets were divided into training and testing datasets by first collecting images from internet search engines. From this massive collection, the testing dataset was constructed by randomly selecting 1,000/100 images per class. Therefore, the train and test are from the same distribution. Purchases, on the other hand, is a dataset used in a kaggle competition of customer and purchase information. The train and test datasets are collected during different time periods: setting 2013/05/01 as the dividing time, data collected before this time is train data and after is test data. In practice, the provided test dataset cannot be used because it carries empty information (the competition was to predict this missing information using the train dataset), so security practitioners use random sampling from the train dataset [38, 48, 50].

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

Table 1: Examination of membership inference defense literature for data drift aspects in chronological order. Notice all datasets are randomized datasets (shaded red in table) that display no data drift, so none of the defense works display nonrandomization. Works from [21, 23, 38, 45] are the standard comparisons in MIA defense (shaded green in table).

Work	Comp. Number	Regularization	Generalizability	Time of Defense	Adaptability	Datasets	Nonrandomization
[48]	1	L2-Regularization	✓	After Training (Top-k, Softmax)	✓	CIFAR-10, CIFAR-100, MNIST, Purchases, Location, Texas100, UCI Adult	
[38]	6	Dropout, L2-regularization, Adversarial regularization	✓	With Training		CIFAR-100, Purchases, Texas100	
[61]	1	Data Obfuscation	✓	With Training		CIFAR-10	
[45]	7	Dropout, Model-Stacking	✓	With Training		CIFAR-10, CIFAR-100, MNIST, Face, Location, Purchases, Adult, News	
[21]	5	-		After Training (MemGuard)	✓	CH-MNIST, Location, Texas100	
[59]	-	-		After Training (Logit Perturbation)	✓	CIFAR-10, Purchases, Face	
[56]	-	-		After Training (Weight Pruning)	✓	CIFAR-10, CIFAR-100, MNIST, ImageNet	
[50]	4	Early Stopping	✓	With Training		CIFAR-100, Purchases, Location, Texas100	
[46]	1	-		After Training (Knowledge Transfer)	✓	CIFAR-10, CIFAR-100, Purchases, Texas100	
[27]	-	MMD+Mix-up	✓	With Training		CIFAR-10, CIFAR-100, MNIST, Purchases, Texas100	
[13]	-	-		After Training (Entropy-based Noise)	✓	CIFAR-100, MNIST, GTSRB	
[22]	-	Data Augmentation	✓	With Training		CIFAR-10, CIFAR-100, MNIST, Fashion-MNIST	
[53]	-	Ensembling	✓	After Training (Knowledge Transfer)	✓	CIFAR-100, Purchases, Texas100	

Examples of datasets used in MIA settings that require manual division into train and test datasets are Location[3] and Texas100[4]. Location is a tabular dataset of mobile users' location "check-in" information in the Foursquare social network, and Texas100 is a dataset based on archived hospital discharge data of 10 years released by the Texas Department of State Health Services. For both Location and Texas100, a version preprocessed into train and test datasets by random selection from Shokri et al. [48] was used in most works [21, 38, 45, 50]. These datasets were randomized by researchers and display no data drift.

By checking the randomization of evaluation datasets in MIA defense work, we can know if MIA defense has been tested on data drift and therefore the stand of MIA defense against data drift.

3.2 Examining Defense Literature

We investigated the MIA defense literature on the previously mentioned dimensions of data drift, summarized in Table 1. For each work, we report the regularization method, time of defense, and the datasets used in evaluation. Depending on each element, we

[3]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[4]https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm

check the generalizability, adaptability, and nonrandomization, respectively. In addition, we count how many times the work was used as a comparison in the evaluation of other works (the scope being the works listed on the table). The works are listed in time ascending order, with Shokri et al. [48] being the pioneer work of MIA. From this summary, we identify four findings of MIA defense.

-Nonrandomization is not satisfied in any MIA defense work evaluation. This is denoted by the red shading of the nonrandomization column of Table 1. All datasets that were used in defense evaluation were randomized datasets. This means that the defense performance reported in each work is unknown in data drift conditions. Furthermore, this reflects the lack of attention to MIA in data drift conditions. In this work, we intend to adjust this negligence and alert the dangers of data drift in MIA by testing standard MIA defenses in drifted datasets.

-The dataset pool used in evaluation of MIA defense is limited. Adding on to the first speculation, the range of datasets to choose from is not wide. MIA defense literature mainly base their evaluations on the datasets used by Shokri et al. [48]. As a fair and effective means of comparing with previous work this is natural, but it hinders testing in different environments, e.g. data drift. In this work, we include evaluations on a different dataset that can be easily tailored for data drift.

-Generalizability and Adaptability are generally not observed simultaneously. Excluding the first and last entry of Table 1, each defense method relies on either defense with training or defense after training. However, there is no reason for choosing only one of generalizability or adaptability because they are completely orthogonal characteristics of MIA defense; a regularization can be applied in training, and a logit masking can be applied on this trained model afterwards. For this reason, in our work, we set early stopping [50], a regularization method, as a default when training due to its freedom in application with other defense methods (with or after training).

-The benchmarks in MIA defense are adversarial regularization [38], model-stacking [45], MemGuard [21], and early stopping [50]. The comparison numbers of Table 1 show that the above defense methods were compared to most among the literature, shaded in green. Compared to the citation count of a work, counting the number of times a defense method has been directly compared to better depicts the *usage* of a defense work. Therefore, we follow the past literature and use the methods that have appeared most in other defense works as the benchmark when evaluating in data drift conditions. Note that this work intends on studying the effect of data drift on MIA defense in general, therefore the benchmark methods are the best candidates to test this.

4 APPLYING DATA DRIFT

4.1 Designing Data Drift

The category of data drift that is considered in this work is covariate shift. As the first work of incorporating data drift with MIA, we encountered three challenges. We address these challenges by designing authentic and synthetic drift using controllable variables.

4.1.1 Scarcity of Datasets with Data Drift. The first challenge in testing MIA defenses on data drift is the lack of dataset due to the uncommon and unique setting of data drift. As mentioned in Section 2.2, the common assumption of machine learning tasks is a closed world assumption of train and test data being from the same distribution. And as verified by the literature study on MIA defense, all datasets used in previous literature are randomized datasets. Preparing self-collected datasets would be a solution in acquiring a drifted dataset (e.g., purposely collecting test data at a later time), but this option is obligated to the burden of data collection; data collection is a rigorous operation that involves both labor and time. Furthermore, even if the dataset is to be prepared, it needs to abide by the characteristics of datasets used in previous MIA literature (e.g., machine learning task, data type) and be reproducible (unless the dataset is open, it is highly unlikely due to time difference) to be meaningful for our purpose. Therefore, preparing and obtaining drifted datasets that meet the underlying requirements is a difficult task.

Table 2: Make-up of data drift in UTKFace. The superscript denotes the condition in which the dataset is being divided and the subscript denotes the task.

Task	Dataset	Type	Condition	Size
Race Classification	UTK_{rc}^{age}	Train	$age > 20$	18,828
		Test	$age \leq 20$	4,880
Age Classification	UTK_{ac}^{race}	Train	$race \in \{0, 2, 3, 4\}$	19,179
		Test	$race = 1$	4,529

Lu et al. [30] conducted a review on data drift and reported the datasets used among studies handling data drift. Unfortunately, the listed datasets have no intersection with datasets used in MIA works; the listed datasets are specific to stream data, which is not consistent with MIA works. Due to this reason, we choose to manipulate multilabel datasets. By dividing the dataset with a condition on one label and training a target model with a classification task of a different label, data drift can be performed without additional processing of the data. Because we have separated a group of data abiding to a specific condition as the test dataset, the condition of data drift is established ($P(x, y_2|y_1) \neq P(x, y_2|y_1')$). An advantage of this data drift design is that it allows the construction of *authentic* data drift. Although not used in evaluation, notice how this construction can be applied to tabular data as well. By being composed of real data, this way of employing data drift can be considered similar to real data drift.

The multilabel dataset that we use in our work is UTKFace[5], a multilabel dataset of human face images. UTKFace is a large-scale dataset with more than 20,000 images annotated with age, gender, and ethnicity information and can be used for multiple machine learning tasks. We divide the age tag into five age group labels (bracket of 20 years) for the task of age classification, and use the race tag as labels for race classification. Table 2 shows how UTKFace was organized into two data drifted datasets, UTK_{rc}^{age} and UTK_{ac}^{race}. The superscript denotes the condition in which the dataset is being divided and the subscript denotes the task. The superscript $full$ refers to random division (no data drift). For age classification (ac), the dataset is divided depending on the race information, and for

[5] https://susanqq.github.io/UTKFace/

race classification (rc), the dataset is divided depending on the age information. This properly portrays covariate shift because the only aspect of the data we have manipulated is the input distribution (there are no changes in label information).

4.1.2 Lack of Standardized Design of Synthetic Data Drift. The data drift design of Section 4.1.1 only applies to multilabel data, which means that this is inapplicable to any datasets in Table 1. Nonetheless, for a fair and proper analysis of data drift on MIA defense, we must utilize these single label datasets. Therefore, we must be able to apply data drift to single label datasets. Unlike multilabel data, there is only one attribute (label information) pertaining to the data. By dividing the dataset based on this attribute, the label information of the data would be forfeited and the machine learning task cannot be conducted.

Because an authentic form of data drift is not possible for these datasets, data drift was simulated in a synthetic manner. We applied synthetic data drift in a form of a normalization filter. Normalization is a widely used step in data-preprocessing in machine learning [19]. In the case of image data, each pixel value is normalized by the mean μ and standard deviation σ.

$$\hat{x} = \frac{x - \mu}{\sigma} \tag{2}$$

For staple datasets such as CIFAR-10 and CIFAR-100, the mean and standard deviation are well-known and used in normalization. For tabular data, the mean and standard deviation was calculated for each attribute, and the normalization filter was applied to the attribute values. By controlling the mean and standard deviation that a data sample is normalized to, covariate shift is simulated. We drift the data on both mean and standard deviation, to confirm the effect of data drift in familiar single label datasets.

Base Datasets. In this work, we prepare six datasets along with their drifted counterparts. The datasets are UTKFace (UTK_{rc}^{full} for the full UTKFace dataset with race labels, UTK_{ac}^{full} for the full UTKFace dataset with age labels), CIFAR-10, CIFAR-100, MNIST[6], and ADULT[7]. From UTKFace, authentic data drift is generated into datasets UTK_{rc}^{age} and UTK_{ac}^{race}. The drifted counterparts (synthetic) of datasets in CIFAR-10, CIFAR-100, MNIST, ADULT are shown with the superscript d. Note that data drift is generated only in the test dataset.

4.1.3 Controlling Data Drift. By controlling the degree of data drift, a more fine-grained analysis of its effect on MIA defense can be achieved. For the case of authentic data drift, there is no variable that directly controls the degree of data drift. The condition applied in Table 2 is deterministic and cannot be used to lessen or strengthen data drift. Therefore, we reserve data $D_r = \{(x_i, y_i) | (x_i, y_i) \text{ is sampled from } P_{tr}(x, y), (x_i, y_i) \notin D_{tr}\}$ from the training dataset D_{tr} and control the amount of injection into the test dataset D_{tst} to construct a dataset with controlled data drift $D_{cont} = D_r \cup D_{tst}$. By changing the cardinality ratio $d_r = |D_{tst}|/|D_{cont}|$, we can regulate influence of data drift in D_{cont} with ease; larger data ratio from D_{tst} will have stronger data drift.

[6]MNIST is a binary image dataset of digits, also frequently used in MIA. http://yann.lecun.com/exdb/mnist/

[7]ADULT is a tabular dataset of 14 continuous and discrete attributes used to predict income. https://archive.ics.uci.edu/dataset/2/adult

Table 3: Data drift controlling variables.

Control Variable	Value
d_r	0, 0.2, 0.4, 0.6, 0.8, 1.0
d_μ	0, 0.1, 0.2, 0.3, 0.4
d_σ	0, 0.1, 0.2, 0.3, 0.4

On the other hand, controlling synthetic data drift is straightforward. Drift variables d_μ and d_σ are added to the mean μ and standard deviation σ, respectively, during normalization. In Equation 2, $\mu + d_\mu$ would contribute to the lateral movement of data, while $\sigma + d_\sigma$ would contribute to the scaling of the data. Table 3 lists the value of drift variables tested in this work.

4.2 Target MIA

The target attacks that will be used in evaluation are a variety of metric-based thresholding methods. For the following attacks, we adopt the method of calculating class-specific thresholds from Song et al. [50] to further improve MIA accuracy.

4.2.1 MIA based on prediction correctness (A1) [26]. This attack is based on the idea that if the model was trained on a data piece, that data would be correctly predicted. This is a simple baseline for MIA using the generalization gap as reasoning. An adversary will infer a data sample as a member if it is correctly predicted, and if not, a non-member.

$$MIA_{corr}(\mathcal{M}, (x, y)) = \mathbb{1}\{\arg\max_i \mathcal{M}(x)_i = y\} \tag{3}$$

$\mathbb{1}$ denotes the indicator function.

4.2.2 MIA based on prediction confidence (A2) [50, 51, 60]. For data samples that a model has been trained on, the prediction confidence is generally higher than the prediction confidence of non-member data. This form of attack is established on this fact, and considers a data sample to be a member only when the prediction vector has a confidence above a statistically derived class-specific threshold τ_y.

$$MIA_{conf}(\mathcal{M}, (x, y)) = \mathbb{1}\{\mathcal{M}(x)_y \geq \tau_y\} \tag{4}$$

4.2.3 MIA based on prediction entropy (A3) [48, 50]. Training data and test data have been shown to have different prediction entropy distributions [48]. The training procedure of deep learning involves the minimization of model loss, and therefore the prediction vector will be closer to a one-hot encoded vector; the entropy of this vector will be close to 0. Using this as motivation, this attack calculates thresholds of this entropy value, and when the prediction entropy is less than this threshold, the data sample is inferred as a member.

$$MIA_{entr}(\mathcal{M}, (x, y)) = \mathbb{1}\{-\sum_i \mathcal{M}(x)_i \log(\mathcal{M}(x)_i) \geq \tau_y\} \tag{5}$$

4.2.4 MIA based on modified prediction entropy (A4) [50, 53]. As a variation of the MIA based on prediction entropy, this attack addresses the issue of entropy of it not containing any information about the ground truth label; any two one-hot vectors will display the same entropy value of 0 when one is classified correctly

while the other is misclassified. Therefore, the modified prediction entropy is defined as such:

$$Mentr(\mathcal{M},(x,y)) = -(1-\mathcal{M}(x)_y)\log(\mathcal{M}(x)_y) - \sum_{i\neq y}\mathcal{M}(x)_i \log(1-\mathcal{M}(x)_i) \quad (6)$$

Similar to the inference using prediction entropy, member data will have a smaller modified entropy value. When the modified entropy value is less than the derived threshold, the data sample will be inferred as a member.

$$MIA_{Mentr}(\mathcal{M},(x,y)) = \mathbb{1}\{Mentr(\mathcal{M},(x,y)) \leq \tau_y\} \quad (7)$$

4.3 Defense Training

The MIA defense methods that we test data drift on are the benchmark methods of Section 3.2: early stopping (D1) [50], dropout (D2) [52], model-stacking (D3) [45], adversarial regularization (D4) [38], and MemGuard (D5) [21]. Because early stopping is a critical generalization method that is commonly used in deep learning frameworks [15], it was used in all defense methods to end training. The parameter that was monitored for early stopping was the loss in the validation dataset, with a patience of 10 epochs.

The base architecture that was used for defense was ResNet18 [14]. For the dropout defense, a dropout layer was joined before the final linear layer of ResNet18, with dropout rate of 0.5. For the model-stacking defense, the models of the first layer are ResNet18 and ResNet34 [14], ensembled by a logistic regression in the second layer. The inference model used for adversarial regularization and MemGuard have the same architecture of three linear layers of size (1024, 512, 64) with a logistic regression for the final layer.

The architecture was trained with a stochastic gradient descent optimizer with momentum 0.9, weight decay of 5e-4 using a 1cycle learning rate scheduler [49]. The starting learning rate was 0.05 for all methods excluding adversarial regularization, where we used a starting learning rate of 1e-6. We adopted the optimization parameters of Jia et al. [21] for MemGuard, with learning rate 0.1 and $c_2 = 10$ and $c_3 = 0.1$. In data drift, the accuracy of the model (classification performance) is insignificant and can be found in the Appendix.

Membership Inference Setup: Defense models were trained using half of the training set (target train dataset) to distinguish between the target train dataset and shadow train dataset. The shadow train dataset is reserved to use as the data collected by the attacker and accordingly, the thresholds used in the attacks (Section 4.2.1-4.2.4) would be calculated using the shadow train dataset. Note that the data used in training is not drifted data and the drifted dataset only pertains to the non-member data used in evaluating MIA accuracy. Using the drifted data in updating the MIA defense to reflect data drift is done in MIAdapt, our attempt at providing an adaptable MIA defense.

4.4 Leveraging Adaptability with MIAdapt

MIA defenses that observed adaptability (Table 1) are methods that have the defense mechanism applied *after* the training of the model, hence possess the opportunity to update the defense in response to data drift. However, these defense methods are not implemented for the purpose of update; these defenses cannot be applied as is and need to be adjusted and patched to be able to accept drift information. We propose MIAdapt, a realization and proof of concept (POC) that updates the defense to be effective in drifted data. MIAdapt adds robustness to the model by incorporating drifted data in optimizing the resulting logit vector without modifying the machine learning model itself. MIAdapt is based on MemGuard [21], which is a work dedicated to solving the MIA Defense Problem (Appendix A). To reflect the presence of data drift into the noise optimization, we need to introduce a condition that restricts the resulting sum to be from the drifted distribution—the input distribution at the time of defense is drifted causing generalization gap ($P_{tr}(\mathcal{M}(x),y) \neq P_{tst}(\mathcal{M}(x),y)$). Therefore, we define the problem specific to drifted settings.

Definition 4 (Drifted MIA Defense Problem)**.** Given a decision function g of the defense classifier, a confidence budget ϵ, $\mathbf{s} = \mathcal{M}(x)$ for data (x,y), we aim to find a randomized noise addition mechanism $\mathcal{R}^*$ solving the following optimization problem:

$$\begin{aligned} \arg\min_{\mathcal{R}} \quad & \mathcal{R}^* = |E_{\mathcal{R}}(g(\mathbf{s}+\mathbf{n})) - 0.5| \\ \text{subject to} \quad & \arg\max_j s_j + n_j = \arg\max_j s_j \\ & E_{\mathcal{R}}(d(\mathbf{s},\mathbf{s}+\mathbf{n})) \leq \epsilon \\ & s_j + n_j \geq 0, \forall j \\ & \sum_j s_j + n_j = 1 \\ & (\mathbf{s}+\mathbf{n},y) \sim P_{tst}(\mathcal{M}(x),y) \end{aligned} \quad (8)$$

The objective function and first four constraints are imported from the original MIA Defense Problem, and the final constraint is the distinction caused by data drift in Definition 4. Despite its necessity, the constraint as is cannot be assured; the origin distribution of a single data point cannot be known.

Eliminating constraint via change of variables: $\mathbf{s}+\mathbf{n}$ is a term that appears in the objective function as well as all the constraints of the Equation 8. However, the presence of drift does not affect any of the constraints. In other words, these constraints are held independent of the fact that $\mathbf{s}+\mathbf{n}$ follows the drifted distribution. For example, the sum of the logit vector is 1 by definition, regardless of drift. Therefore, the only instance that the final constraint affects is in the objective function: the objective function is the only place that we must consider the presence of drift. g is a decision function that is trained to classify membership. By retraining the decision function on the drifted distribution of data, the decision function can be wired to interpret and process the input information as data from the distribution $P_{tst}(\mathcal{M}(x),y)$. Therefore to eliminate the unworkable final constraint, g is substituted by g^*, the decision function trained on the drifted data. With the final constraint removed and indirectly enforced by g^*, Equation 8 effectively becomes congruent to the original MIA Defense Problem.

Implementing MIAdapt: Due to the change of variables to g^*, the implementation of noise optimization is the same as MemGuard. The only difference would be that the decision function is trained on drifted data. Because MIAdapt is implemented by the defender, access to the train dataset is assumed and the logit outputs of

this data through the model $\mathcal{M}$ is labeled 1. The drifted dataset is prepared first using the methods of Section 4.1, which is then fed into $\mathcal{M}$ for the logit information and is labeled 0. The assembling of this negative label data is essentially the supply of $P_{tst}(\mathcal{M}(x), y)$. Following this composition of train data for decision function g^*, MIAdapt follows the same optimization procedure of MemGuard in Section 4.3.

5 RESULTS

5.1 Data Drift enhances MIA

5.1.1 Evaluation on randomized datasets. The MIA accuracy of the datasets and their drifted counterparts for each defense are shown in Table 4. Overall, the results of randomized datasets (for each pair of rows, the upper row) are lower than previous literature [21, 38, 45, 48, 50]. This reflects our decision of employing early stopping (D1) in all defense methods. Because early stopping nearly assures the prevention of overfitting in the target model, it shows adequate defense capabilities. When early stopping is used with other defense methods, for most occasions the MIA performance decreases. This suggests the synergy of defense methods. In particular, MIA accuracy decreases when early stopping is used with other defense methods that show generalizability (D2, D3, D4). For early stopping done on MemGuard (D5), it is less reliable than the previous defense methods and sometimes induce MIA gain.

5.1.2 Evaluation on datasets with data drift. Compared to the MIA performance on the original randomized data sets, the drift-generated datasets have noticable gain in MIA performance. Excluding the single result of UTK_{ac}^{full} and its drifted dataset on prediction correctness attacks, all results for (dataset, attack, defense) triple show an increase in MIA attack performance when tested on the drifted data, a sign of increased MIA vulnerability. The performance of each attack averaged on all defense methods can be seen in each rightmost column of Table 4. The best average gain of all MIA on each dataset is 0.075 for UTK_{rc}^{age}, 0.089 for UTK_{ac}^{race}, 0.17 for CIFAR-10^d, 0.125 for CIFAR-100^d, 0.26 for MNIST^d, and 0.035 for ADULT^d. Note that this performance increase is not due to a change in attack methodology, but solely the effect of data drift—a change in test dataset. Furthermore, this average gain is not biased to a certain weak-performing defense method because the MIA defense methods do not demonstrate any particular defense improvement for each MIA. This shows that **data drift plays a significant factor in enhancing MIA vulnerability, even when the target model is defended with benchmark MIA defenses.**

The single result that did not show increased MIA attack performance was UTK_{ac}^{full} against data drift for MIA based on prediction correctness (A1) (Table **??**). Statistically this attack method showed to be the most unaffected by data drift. For each attack method, the average gain of MIA vulnerability was 0.051 for A1, 0.141 for A2, 0.131 for A3, and 0.139 for A4. The relatively inferior improvement in MIA vulnerability that data drift caused in A1 is mainly due to it being the most naive form of MIA based on prediction correctness. Because the datasets UTK_{rc}^{age} and MNIST^d show a relatively minor model performance impairment (explained more in Section 6.1), the target model was able to perform its task correctly and therefore the attack would mislabel the drifted data samples as train data (case of UTK_{ac}^{full} and UTK_{ac}^{race} for A1).

MIA by prediction confidence (A2) and modified entropy (A4) are the most enhanced attacks by data drift. The resulting MIA accuracy gain by synthetic data drift (CIFAR-10^d, CIFAR-100^d, MNIST^d, and ADULT^d) is shown to be more prominent than gain by authentic drift (UTK_{rc}^{age} and UTK_{ac}^{race}). This reflects the gap between authentic data drift generation, where the data is unprocessed, and synthetic data drift, where the data was processed by artificial perturbations. However, as a concept that cannot be expected nor can the degree of data drift be predicted, our design of data drift satisfies the role of covariate shift simulation in the test data.

5.2 Controlling the Degree of Data Drift

Using the control variables of Table 3, we varied the degree of data drift to the resulting test dataset and evaluated MIA defenses (Figure 2). Due to the excessive variations in (dataset, defense, and attack) settings, we fix the dataset and report the effect of changing the degree of data drift with comparison to the MIA (Figure 2a-2d) and the MIA defense (Figure 2e-2h). Two instances from both authentic and synthetic data drift generation can be seen. In this analysis, synthetic data drift is generated from covariate shift by mean and covariate shift by standard deviation.

As expected, strengthening the degree of data drift increases the MIA attack accuracy a larger amount. This effect is stronger in synthetic data drift generation (Figure 2c, 2d). Unlike other controllable variables, for a larger value of d_μ, a larger leakage of membership information (Figure 2c) occurs. The rest of the controllable variables maintain a near-linear relationship with MIA success rate. In addition, the MIA success rates in varying d_μ and d_σ show to be nearly identical among the target attacks. This is because the operation of applying synthetic data drift is standardized. On the other hand, varying the cardinality in authentic data drift introduces random samples to the test dataset. Compared to changing the normalization parameters, this is bound to be noisy, hence the wider dispersion of MIA success rate per attack. Out of the target MIA, MIA by prediction modified entropy is the highest achieving attack in all cases.

When comparing the defense methods, we see a similar pattern; with stronger drift, there is stronger MIA success rate. Like the observation of d_μ in Figure 2c, Figure 2g shows that for a higher d_μ, the leakage of membership information becomes greater. Finding this trend in all target MIA and all MIA defense, d_μ can be seen as the most influential variable to induce data drift. Unfortunately, the MIA defense methods have little difference in defense capability of data drift. There is no specific defense candidate that varies less with the data drift control variable. In other words, **no defense benchmark is superior to the others, and all of them have a monotonic relationship with data drift**.

5.3 MIAdapt Evaluations

By taking into account the shifted input data distribution, MIAdapt shows to be less affected by data drift and provides stronger defense against MIA than the benchmark MIA defense methods of Table 5 in most attack settings. In some cases, this effect is to a significant extent, practically nullifying MIA (MIA attack success rate being near

Table 4: MIA accuracy on multiple datasets. Each dataset pair denotes the randomized dataset and the drifted dataset. The shaded rows show the attack success rates of the drifted datasets. Bold values show the larger MIA accuracy between the pair of datasets. UTK_{rc}^{age} and UTK_{ac}^{race} denote the case of $d_r = 1.0$ and CIFAR-10^d, CIFAR-100^d, MNIST^d, and ADULT^d denote the case of $d_\sigma = 0.4$. All forms of MIA on drifted datasets show an increase in MIA vulnerability.

Table 4a: MIA by prediction correctness attack (A1).

	D1	D2	D3	D4	D5	Average
UTK_{rc}^{full}	0.534	0.528	0.508	0.506	0.534	0.522
UTK_{rc}^{age}	0.584	0.604	0.602	0.588	0.607	**0.597**
UTK_{ac}^{full}	0.516	0.525	0.523	0.513	0.516	**0.519**
UTK_{ac}^{race}	0.498	0.515	0.495	0.533	0.511	0.510
CIFAR-10	0.519	0.515	0.518	0.515	0.519	0.517
CIFAR-10^d	0.615	0.609	0.640	0.603	0.615	**0.616**
CIFAR-100	0.550	0.539	0.503	0.541	0.550	0.537
CIFAR-100^d	0.639	0.627	0.592	0.655	0.638	**0.630**
MNIST	0.503	0.502	0.503	0.502	0.503	0.503
MNIST^d	0.537	0.524	0.542	0.515	0.537	**0.531**
ADULT	0.496	0.497	0.500	0.495	0.500	0.498
ADULT^d	0.500	0.546	0.500	0.548	0.500	**0.519**

Table 4b: MIA by prediction confidence attack (A2).

	D1	D2	D3	D4	D5	Average
UTK_{rc}^{full}	0.537	0.532	0.547	0.507	0.538	0.532
UTK_{rc}^{age}	0.556	0.594	0.591	0.614	0.596	**0.590**
UTK_{ac}^{full}	0.522	0.526	0.536	0.512	0.523	0.524
UTK_{ac}^{race}	0.592	0.612	0.644	0.612	0.603	**0.613**
CIFAR-10	0.518	0.515	0.515	0.514	0.519	0.516
CIFAR-10^d	0.687	0.675	0.689	0.691	0.686	**0.686**
CIFAR-100	0.536	0.530	0.517	0.534	0.539	0.531
CIFAR-100^d	0.655	0.645	0.646	0.680	0.654	**0.656**
MNIST	0.500	0.504	0.508	0.505	0.502	0.504
MNIST^d	0.836	0.807	0.875	0.786	0.837	**0.828**
ADULT	0.496	0.500	0.502	0.496	0.496	0.498
ADULT^d	0.580	0.592	0.538	0.584	0.590	**0.577**

Table 4c: MIA by prediction entropy attack (A3).

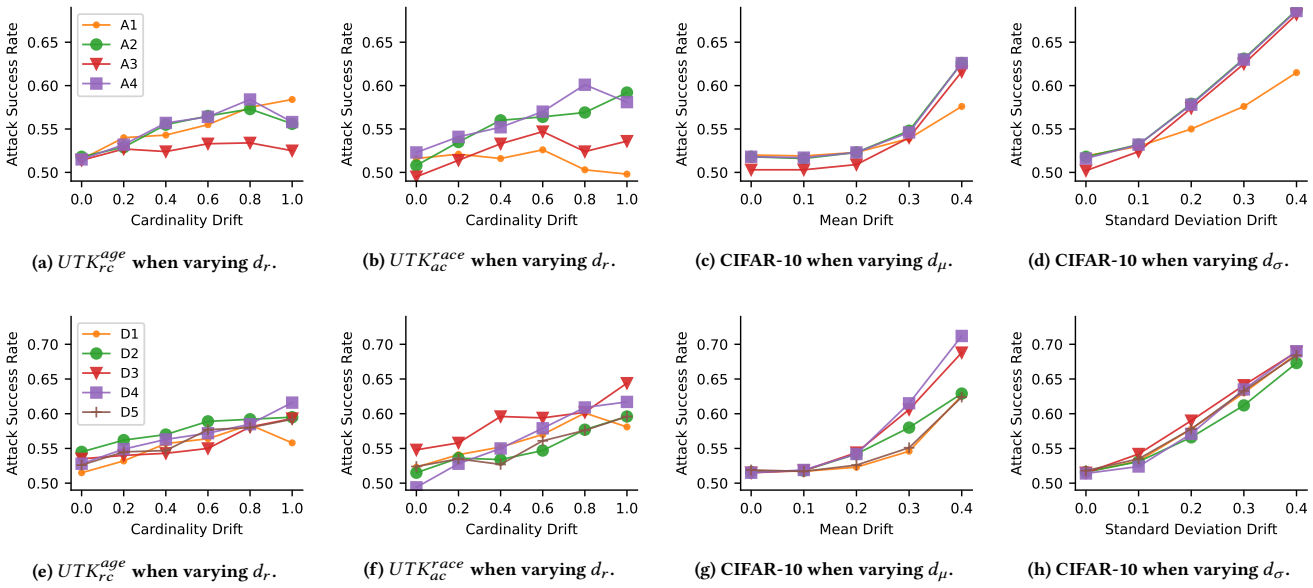
	D1	D2	D3	D4	D5	Average
UTK_{rc}^{full}	0.505	0.509	0.523	0.499	0.505	0.508
UTK_{rc}^{age}	0.525	0.557	0.546	0.549	0.544	**0.544**
UTK_{ac}^{full}	0.503	0.511	0.508	0.506	0.505	0.507
UTK_{ac}^{race}	0.536	0.592	0.602	0.615	0.572	**0.583**
CIFAR-10	0.502	0.504	0.499	0.500	0.503	0.502
CIFAR-10^d	0.682	0.666	0.631	0.686	0.680	**0.669**
CIFAR-100	0.507	0.498	0.523	0.491	0.505	0.505
CIFAR-100^d	0.616	0.615	0.613	0.584	0.617	**0.609**
MNIST	0.500	0.506	0.507	0.503	0.501	0.503
MNIST^d	0.837	0.808	0.876	0.787	0.838	**0.829**
ADULT	0.496	0.500	0.499	0.498	0.496	0.498
ADULT^d	0.610	0.592	0.520	0.591	0.610	**0.585**

Table 4d: MIA by prediction modified entropy attack (A4).

	D1	D2	D3	D4	D5	Average
UTK_{rc}^{full}	0.537	0.531	0.547	0.508	0.536	0.532
UTK_{rc}^{age}	0.558	0.595	0.593	0.616	0.592	**0.591**
UTK_{ac}^{full}	0.525	0.520	0.537	0.511	0.527	0.524
UTK_{ac}^{race}	0.581	0.596	0.644	0.617	0.596	**0.607**
CIFAR-10	0.516	0.516	0.515	0.514	0.518	0.516
CIFAR-10^d	0.686	0.673	0.689	0.690	0.684	**0.684**
CIFAR-100	0.539	0.531	0.517	0.537	0.540	0.533
CIFAR-100^d	0.650	0.642	0.646	0.673	0.653	**0.653**
MNIST	0.496	0.507	0.508	0.503	0.501	0.503
MNIST^d	0.834	0.806	0.875	0.785	0.835	**0.827**
ADULT	0.496	0.500	0.502	0.496	0.496	0.498
ADULT^d	0.585	0.592	0.518	0.604	0.585	**0.577**

0.5) for MNIST^d. The bold numbers show which method is more effective as a defense, and in the majority of dataset/attack pairs, MIAdapt was the most effective in decreasing the MIA success rate. Furthermore, in all the instances that a different defense method was more effective (e.g., CIFAR-100^d with attack A1), MIAdapt was the next most effective out of all the other defenses. For the case of MNIST^d, MIAdapt has shown to decrease the MIA success rate by an average of 0.248, the largest reduction rate observed in all drifted datasets. These results indicate that the retraining and substitution of the decision function g^* proved to be an adequate enforcer for the data drift condition of the Drifted MIA Defense Problem.

There did exist some occasional underperformance when compared to early stopping (D1). By definition, D1 is the training approach with the fewest iterations, indicating faster convergence and potentially *reduced exposure of data samples for memorization* by the model. The observation that MIAdapt may occasionally lag behind D1 in terms of performance highlights the need for further investigation. Relying soley on reduced exposure as the optimal defense strategy with data drift may be deemed suboptimal as it only pertains to a subset of cases (3 out of 24 cases). Exploration of MIAdapt along with reduced iterations in training could improve effectiveness against MIA in the context of data drift. Nevertheless, **MIAdapt remains the leading candidate in MIA defense during data drift and represents the possibility of being able**



(a) UTK_{rc}^{age} when varying d_r. (b) UTK_{ac}^{race} when varying d_r. (c) CIFAR-10 when varying d_μ. (d) CIFAR-10 when varying d_σ.

(e) UTK_{rc}^{age} when varying d_r. (f) UTK_{ac}^{race} when varying d_r. (g) CIFAR-10 when varying d_μ. (h) CIFAR-10 when varying d_σ.

Figure 2: The change in MIA success rate when varying the degree of data drift in multiple scenarios. The upper graphs compare the attacks in the defense setting of early stopping (D1) and the lower graphs compare the defense methods in the attack setting of MIA based on prediction modified entropy (A4). Graphs on the rest of the cases can be found in link.

to utilize current data (that is from different distribution from the train data) to update the defense mechanism without retraining the whole model.

Drifted Data Emphasize Themselves: The primary reason behind the increased vulnerability of membership inference due to data drift is rooted in the presence of drifted data, which exhibits a markedly different logit distribution compared to the training data. The attacks discussed in Section 4.2 pertain to threshold-based MIAs, wherein a metric has been empirically proven to demonstrate a distinction between trained and untrained data.

As the intensity of data drift strengthens, the distributions of the training data and drifted data become more disparate, resulting in a well-defined threshold between these two distributions (i.e. drifted data emphasize themselves similarly to outliers). MIAs take advantage of this separation which lead to increased MIA performance. Consequently, this explains the effectiveness of MIAdapt in reducing the potency of MIAs. By incorporating drifted samples into the computation of these thresholds, MIAdapt effectively lessens the separation between distributions, thus making the problem of membership inference more challenging. As a result, MIAdapt demonstrates itself to be an adaptable and functional defense mechanism.

Table 5: MIA success rate on MIAdapt. The evaluations of MIAdapt are shown on the second row of each dataset and are compared to the defense method that shows strongest defense capability for each dataset/attack pair. The bolded instances mark the prominent defense for each setting.

Dataset	A1	A2	A3	A4
UTK_{rc}^{age}	0.584 (D1)	**0.556** (D1)	0.525 (D1)	**0.558** (D1)
	0.573	0.579	**0.523**	0.579
UTK_{ac}^{race}	**0.498** (D1)	0.592 (D1)	**0.536** (D1)	0.581 (D1)
	0.538	**0.588**	0.579	**0.580**
CIFAR-10^d	0.609 (D2)	0.675 (D2)	0.631 (D3)	0.673 (D2)
	0.542	**0.575**	**0.568**	**0.575**
CIFAR-100^d	**0.592** (D3)	0.645 (D2)	0.584 (D4)	0.642 (D2)
	0.618	**0.624**	**0.581**	**0.621**
MNIST^d	0.515 (D4)	0.786 (D4)	0.787 (D4)	0.785 (D4)
	0.497	**0.502**	**0.554**	**0.502**
ADULT^d	0.500 (D1)	0.538 (D3)	0.520 (D3)	0.518 (D3)
	0.500	**0.510**	**0.514**	**0.510**

6 SECURITY INSIGHT

6.1 Verifying Drift Design

The extent of drift should be considered through our design of data drift, i.e., an appropriate or realistic scale of drift needs to be applied to the randomized data. It is important in this work to not overestimate data drift for a more accurate outlook on the effect of data drift to MIA defense. We assess drift extent by observing the model accuracy on the drifted data. In practice, accuracy on the drifted data also serves as an indicator to dispose the model and train a new one in perspective to a service operator [6, 8]. A severe accuracy degradation would imply excessive data drift and signal for retraining of the model, rather than updating the adaptable

defense. In our work, threshold for accuracy performance loss will be when it degrades by more than 20%.

Table 6 shows the model performance on drifted data. For all drifted data, the performance degradation does not exceed our selected threshold of 20%. The greatest performance loss (P.L.) was 0.193 when the CIFAR-10 dataset was drifted by $d_\sigma = 0.4$. This amount of data drift is tolerable enough to proceed without retraining the model, but troublesome enough that the defense should be updated by MIAdapt. In the case of a less drastic extent of drift ($d_\sigma = 0.3$), the largest performance loss was 0.127, much below the threshold for model retraining. This verifies that the values of control variables (Table 3) used for data drift simulation were appropriate and not an overestimated amount.

Furthermore, one crucial finding through verifying our drift design is that the degree of model performance loss does not equate to, or have direct relations to, the extent of data drift on MIA vulnerability. We look at the cases of UTK_{ac}^{race} and $MNIST^d$: the two drifted datasets that show the smallest model performance loss of 0.003 and 0.058 respectively (Table 6). However, these two datasets have different results in MIA vulnerability shown in Table 4. UTK_{ac}^{race} shows an average MIA increase of 0.060, while $MNIST^d$ shows an average MIA increase of 0.251, which is more than 1.5 times the original MIA success rate.

Data drift intensely affected the MIA vulnerability in the dataset $MNIST$, while it did not have that strong of an effect in the dataset UTK_{ac}^{full}, despite the fact that the drifted variants of the datasets both showed minor model performance loss. This means that the current practice of "model performance loss" being an indicator of data drift may lead to false negatives. In other words, **even if the model performance loss does not show to be significant to a service operator, data drift may already be present and leave the model susceptible to MIA**. As the threshold condition for retraining itself may not be sufficient in resisting against data drift, this brings more emphasis to an adaptable and updatable defense such as MIAdapt. Because MIAdapt is a relatively low-cost updatable method compared to retraining a full model, it can be consistently used in periodic updates. By updating a model's MIA defense periodically, the false negative problem that the threshold condition contains can be avoided.

6.2 Defense Options in Data Drift

As our results indicate, data drift induces increased vulnerability to MIA attacks. As a moderator of a model who wants to counteract this, there are three available options. The first option is to maintain an up-to-date model, and this can be accomplished by frequent retraining of the model with the current data. As shown in Section 6.1, waiting for noticable performance degradation to retrain a model may overlook the MIA vulnerability of models that are less affected by data drift. Therefore the training needs to occur on a periodic schedule. Also called offline learning, it is a default solution for negating effects of data drift [35]. However, this option is expensive in terms of resources, time, and cost.

The next options accept data drift circumstances and deal with the MIA aspect. One option would be to secure the information of train data. An adversary requires an auxiliary dataset to initiate MIA. This auxiliary dataset illustrates the shadow data in which both

Table 6: Model performance on authentic and synthetic data drift. P.L. is the largest performance loss due to drift.

Table 6a: Model performance on drifted datasets by d_r

Dataset	0.0	0.2	0.4	0.6	0.8	1.0	P.L.
UTK_{rc}^{age}	0.719	0.706	0.670	0.606	0.621	0.561	0.158
UTK_{ac}^{race}	0.662	0.665	0.690	0.659	0.685	0.704	0.003

Table 6b: Model performance on drifted datasets by d_σ.

Dataset	0.0	0.1	0.2	0.3	0.4	P.L.
CIFAR-10^d	0.886	0.863	0.824	0.770	0.693	0.193
CIFAR-100^d	0.668	0.642	0.593	0.541	0.488	0.180
MNIST^d	0.988	0.983	0.970	0.953	0.920	0.058
ADULT^d	0.840	0.835	0.818	0.785	0.760	0.080

binary classifier optimizations and metric-based thresholding MIA use to learn the data characteristics of train and test data. When the moderator secures the information of train data distribution to not be leaked, composing shadow train data would be challenging. This is a fundamental prevention of MIA, and is therefore an idealistic solution to MIA in general. However, there are times when the train data cannot be completely isolated; models may be trained on public data. These cases allow the collection of shadow train data, and therefore MIA become viable.

The last option is to employ an adaptive MIA defense. As shown by MIAdapt (Table 5), the possibility of a defense method that adapts to data drift has been confirmed. As a prototype of an adaptive MIA defense, updating a model's defense is relatively easier than retraining the model. Whenever a sufficiently large batch of drifted data is collected, the model can be updated to neutralize the effect of data drift. By acknowledging the severe effects of data drift in membership privacy, we hope that our prototype will serve as an example that instigates further research on adaptive MIA defenses in negating data drift. Future research can focus on advancing methods of collecting and applying distribution data to MIA defense using our design of data drift.

6.3 Limitations

While we study the effects of covariate shift, concept drift is not explicitly addressed in this study. This omission stems from the complexities associated with our primary focus on MIA. The existing literature on MIAs primarily revolves around the extraction of private information from machine learning models employed for *classification tasks*. In the context of classification tasks, variations within the label distribution can be considered as elaborated in Section 4.1, but assumptions concerning the addition or removal of classes (concept drift) lie beyond the scope of classification; a classification model is inductive therefore physically cannot classify a data sample with a new label that it does not already know. Consequently, the implications of concept drift on MIA are left as a subject for future work.

Despite the success in MIA mitigation during data drift for some datasets, judging strictly on MIAdapt as a practical solution opens concerns; performance-wise, MIAdapt was not optimal for all cases

and serves its purpose only as a POC for data drift adaptability. In addition, by being based on MemGuard, MIAdapt also bears the inherent shortcoming that the noise optimization takes place only after the model is able to calculate the resulting logit vector on a data sample. This means that MIAdapt needs to perform optimization for every single data sample that is requested, which will cause latency issues in deployment. It is hoped that further contributions from others will expand and strengthen this concept of an adaptable defense mechanism.

7 CONCLUSION

In this work, we provided the first discussion of data drift with MIA. Our literature review on MIA defense tells us that data drift has not been considered throughout past works and therefore unprepared for the effects of data drift. To accommodate data drift in MIA, we implemented a design of generating data drift: authentic data drift controlled by cardinality ratio d_r from multilabel datasets and synthetic data drift controlled by the normalization variables d_μ and d_σ from single label datasets. Our evaluation shows that data drift enhances MIA and penetrates the benchmark MIA defenses. We promote the usage of MIAdapt, our POC on updating MIA defense that shows notable mitigation performance on multiple drifted datasets. We hope these results emphasize the risks current MIA defenses are exposed to and that they highlight the need to consider data drift in future MIA defense research.

ACKNOWLEDGMENTS

This research was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921).

REFERENCES

[1] Hyrum S Anderson and Phil Roth. 2018. Ember: an open dataset for training static pe malware machine learning models. *arXiv preprint arXiv:1804.04637* (2018).
[2] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 443–448.
[3] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 343–362.
[4] Yizheng Chen, Shiqi Wang, Dongdong She, and Suman Jana. 2020. On training robust {PDF} malware classifiers. In *29th USENIX Security Symposium (USENIX Security 20)*. 2343–2360.
[5] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International conference on machine learning*. PMLR, 1964–1974.
[6] Luke Diliberto. 2021. *When Should You Retrain Machine Learning Models?* https://www.phdata.io/blog/when-to-retrain-machine-learning-models/
[7] Domino. 2022. *How COVID-19 Has Infected AI Models.* https://www.dominodatalab.com/blog/how-covid-19-has-infected-ai-models
[8] Emeli Dral. 2021. *When to Retrain an Machine Learning Model? Run these 5 checks to decide on the schedule.* https://www.kdnuggets.com/2021/07/retrain-machine-learning-model-5-checks-decide-schedule.html
[9] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.
[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.
[12] David J Hand. 2006. Classifier technology and the illusion of progress. *Statistical science* 21, 1 (2006), 1–14.
[13] Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, Michael Backes, and Mario Fritz. 2021. Mlcapsule: Guarded offline deployment of machine learning as a service. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3300–3309.
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[15] Reinhard Heckel and Fatih Furkan Yilmaz. 2020. Early Stopping in Deep Networks: Double Descent and How to Eliminate it. In *International Conference on Learning Representations*.
[16] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).
[17] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online learning: A comprehensive survey. *Neurocomputing* 459 (2021), 249–289.
[18] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2021. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* (2021).
[19] Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. 2020. Normalization techniques in training dnns: Methodology, analysis and application. *arXiv preprint arXiv:2009.12836* (2020).
[20] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. 2021. Practical blind membership inference attack via differential comparisons. *arXiv preprint arXiv:2101.01341* (2021).
[21] 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*. 259–274.
[22] Yigitcan Kaya and Tudor Dumitras. 2021. When Does Data Augmentation Help With Membership Inference Attacks?. In *International conference on machine learning*. PMLR, 5345–5355.
[23] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2020. On the effectiveness of regularization against membership inference attacks. *arXiv preprint arXiv:2006.05336* (2020).
[24] Mark G Kelly, David J Hand, and Niall M Adams. 1999. The impact of changing populations on classifier performance. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. 367–371.
[25] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[26] Klas Leino and Matt Fredrikson. 2020. Stolen Memories: Leveraging Model Memorization for Calibrated {White-Box} Membership Inference. In *29th USENIX Security Symposium (USENIX Security 20)*. 1605–1622.
[27] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. 5–16.
[28] Zheng Li and Yang Zhang. 2020. Label-leaks: Membership inference attack with label. *arXiv preprint arXiv:2007.15528* (2020).
[29] Martina Lindorfer, Matthias Neugschwandtner, and Christian Platzer. 2015. Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis. In *2015 IEEE 39th annual computer software and applications conference*, Vol. 2. IEEE, 422–433.
[30] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.
[31] Ankur Mallick, Kevin Hsieh, Behnaz Arzani, and Gauri Joshi. 2022. Matchmaker: Data Drift Mitigation in Machine Learning for Large-Scale Systems. *Proceedings of Machine Learning and Systems* 4 (2022), 77–94.
[32] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In *Network and Distributed Systems Security (NDSS) Symposium*.
[33] Youssef Miyah, Mohammed Benjelloun, Sanae Lairini, and Anissa Lahrichi. 2022. COVID-19 Impact on Public Health, Environment, Human Psychology, Global Socioeconomy, and Education. *The Scientific World Journal* 2022 (2022).
[34] Reza Moradi, Reza Berangi, and Behrouz Minaei. 2020. A survey of regularization strategies for deep models. *Artificial Intelligence Review* 53, 6 (2020), 3947–3986.
[35] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45, 1 (2012), 521–530.
[36] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems* 32 (2019).
[37] Seung Ho Na, Hyeong Gwon Hong, Junmo Kim, and Seungwon Shin. 2022. Closing the Loophole: Rethinking Reconstruction Attacks in Federated Learning from a Privacy Standpoint. In *Annual Computer Security Applications Conference*. 332–345.
[38] 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*. 634–646.

[39] 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)*. IEEE, 739–753.

[40] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. *Advances in neural information processing systems* 30 (2017).

[41] Andrew Y Ng. 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*. 78.

[42] OECD. 2020. *COVID-19 and the aviation industry: Impact and policy responses*. https://www.oecd.org/coronavirus/policy-responses/covid-19-and-the-aviation-industry-impact-and-policy-responses-26d521c1/

[43] Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, 55–69.

[44] Maria Rigaki and Sebastian Garcia. 2020. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646* (2020).

[45] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed Systems Security Symposium 2019*. Internet Society.

[46] Virat Shejwalkar and Amir Houmansadr. 2021. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9549–9557.

[47] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.

[48] 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)*. IEEE, 3–18.

[49] Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. SPIE, 369–386.

[50] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.

[51] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 241–257.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[53] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. 2021. Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture. *arXiv preprint arXiv:2110.08324* (2021).

[54] Sergios Theodoridis and Konstantinos Koutroumbas. 2006. *Pattern recognition*. Elsevier.

[55] Vladimir Vapnik. 1999. *The nature of statistical learning theory*. Springer science & business media.

[56] Yijue Wang, Chenghong Wang, Zigeng Wang, Shanglin Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. 2020. Against membership inference attack: Pruning is all you need. *arXiv preprint arXiv:2008.13578* (2020).

[57] Gerhard Widmer and Miroslav Kubat. 1996. Learning in the presence of concept drift and hidden contexts. *Machine learning* 23, 1 (1996), 69–101.

[58] Keisuke Yamazaki, Motoaki Kawanabe, Sumio Watanabe, Masashi Sugiyama, and Klaus-Robert Müller. 2007. Asymptotic bayesian generalization error when training and test distributions are different. In *Proceedings of the 24th international conference on Machine learning*. 1079–1086.

[59] Ziqi Yang, Bin Shao, Bohan Xuan, Ee-Chien Chang, and Fan Zhang. 2020. Defending model inversion and membership inference attacks via prediction purification. *arXiv preprint arXiv:2005.03915* (2020).

[60] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 268–282.

[61] Tianwei Zhang, Zecheng He, and Ruby B Lee. 2018. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860* (2018).

A MIA DEFENSE PROBLEM

Definition 5 (MIA Defense Problem [21])**.** Given a decision function $g : \mathbf{s} \rightarrow [0, 1]$ of the defense classifier, a confidence budget ϵ, a true confidence score vector $\mathbf{s}$, the defender aims to find a randomized noise addition mechanism $\mathcal{R}^*$ solving the following optimization problem:

$$\begin{aligned} \arg\min_{\mathcal{R}} \quad & \mathcal{R}^* = |E_{\mathcal{R}}(g(\mathbf{s}+\mathbf{n})) - 0.5| \\ \text{subject to} \quad & \arg\max_j s_j + n_j = \arg\max_j s_j \\ & E_{\mathcal{R}}(d(\mathbf{s}, \mathbf{s}+\mathbf{n})) \leq \epsilon \\ & s_j + n_j \geq 0, \forall j \\ & \sum_j s_j + n_j = 1 \end{aligned} \tag{9}$$

The first constraint maintains the logit label decision after noise addition. The second constraint restricts the magnitude of noise $\mathbf{n}$ from growing too large. The third constraint only allows positive logit values and the final constraint makes sure the addition of the noise adds up to 1 because it is a logit vector. The noise $\mathbf{n}$ is optimized with the restraints so that the decision of the defender classifier g is flipped, which would effectively mean that the output logit has evaded MIA.

B ACCURACY OF MODELS

Table 7: Model accuracy on multiple datasets.

	D1	D2	D3	D4	D5
UTK_{rc}^{full}	0.667	0.687	0.572	0.623	0.667
UTK_{ac}^{full}	0.621	0.667	0.623	0.701	0.621
CIFAR-10	0.891	0.886	0.865	0.898	0.891
CIFAR-100	0.670	0.641	0.453	0.547	0.670
MNIST	0.989	0.991	0.990	0.991	0.989
ADULT	0.810	0.797	0.834	0.793	0.810

Table 7 shows the model accuracy of our defense methods. The performance of these models are on-par with the reported performance of previous works. Because our work concerns data drift (test data distribution is different from train data distribution), the model accuracy is not critical, and only used to assure that a model was trained properly.