On the Vulnerability of Data Points Under Multiple Membership Inference Attacks and Target Models

Mauro Conti, *Fellow, IEEE*, Jiaxin Li, and Stjepan Picek, *Senior Member, IEEE*

***Abstract*—Membership Inference Attacks (MIAs) infer whether a data point is in the training data of a machine learning model, posing privacy risks to sensitive data like medical records or financial data. Intuitively, data points that MIA accurately detects are vulnerable. Those data points may exist in the data of different target models, each susceptible to multiple MIAs. As such, the vulnerability of data points under multiple MIAs and target models represents a significant challenge. This article defines several metrics reflecting data points' vulnerability and capturing vulnerable data points under multiple MIAs and target models. We implement 77 MIAs, with an average attack accuracy over target models ranging from 0.5 to 0.9, to support our analysis with our scalable and flexible platform, Various Membership Inference Attacks Platform (VMIAP). Based on the results, we observe that MIA has an inference tendency to some data points despite a low overall inference performance. Furthermore, previous approaches are unsuitable for finding vulnerable data points under multiple MIAs and target models. Finally, we explore the impact of retraining target, shadow, and attack models separately on the vulnerability of data points.**

***Index Terms*—Machine learning, privacy, membership inference attack, vulnerable data points, metrics.**

I. Introduction

MACHINE learning, especially with the development of deep learning, promotes many real-world applications, e.g., computer vision, natural language processing, and data mining. With machine learning's frequent practical applications, security and privacy problems arise, including models' fairness [1], [2], adversarial examples [3], [4], and model stealing [5], [6]. A Membership Inference Attack (MIA) detects whether a data point is in the training data of a machine learning model, violating the data points' privacy. MIA became an important topic after the seminal work by Shokri et al. [7]. A successful MIA has serious consequences, especially when the training data is sensitive, like medical data, bank account information, and historical browsing records. Enacting data privacy laws such as the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act (CCPA) in the United States has heightened the community's emphasis on data security and privacy. Consequently, MIAs have garnered significant attention from academia and industry.

It has been empirically demonstrated that some data points in the dataset are more vulnerable to MIA [8], [9], [10]. MIAs generally infer vulnerable data points with a high probability. Long et al. selected data points with fewer neighbors on the combination feature space represented by outputs of reference models as vulnerable ones [8]. In our paper, we denote this method as the neighbors-based method. Furthermore, the authors mentioned identifying vulnerable data points with the outlier detection method is possible. Carlini et al. [11] discussed that vulnerable data points are likelier to be outliers. Therefore, we investigate possible vulnerable data points with an outlier detection method (SUOD) [12] to provide a comparison choice. In 2021, Song et al. defined the privacy risk score of a single data point as the posterior probability of being a member conditioned on the target model's behavior on this data point [9]. After that, Duddu et al. proposed to use the Shapley value to measure data points' susceptibility to MIAs [10]. Those methods obtain vulnerable data points within the training data of one target model and one specific MIA. They do not explore data points' vulnerability under multiple MIAs and different target models. However, more MIAs threaten the target models as current works propose more advanced attacks [11], [13], [14]. For example, previous works proposed various methods [7], [15] to attack models trained with the help of Machine Learning as a Service (MLaaS). Third-party platforms (Google, Amazon, and Azure) train the model based on the uploaded data and publicize the final model as an API, with many users having access to query the API. Therefore, attacking the trained model with multiple MIAs is feasible and relevant.

Training multiple models from one dataset is a common practice in ensemble learning and the previous MIA. For example, Bagging [16] trains several models with multiple training sets, each containing the same number of data points randomly drawn from the dataset with replacement to vote the final prediction. The model structure and hyperparameters could be the same for those models. In the work of Carlini et al. [11], the authors train N shadow models on N subsets of a dataset. The target point (x, y) occurs in $N/2$ subsets. Therefore, they could estimate the metric distributions while the target point is separated in the training and test sets. With the estimated metric distributions of

Received 24 October 2022; revised 28 November 2024; accepted 8 February 2025. Date of publication 17 February 2025; date of current version 11 July 2025. This work was supported by the Chinese Scholarship Council (CSC). *(Corresponding author: Jiaxin Li.)*

Mauro Conti is with the Department of Mathematics, University of Padua, 35131 Padua, Italy, and with the TU Delft, 2628 CD Delft, The Netherlands, and also with the University of Washington, Seattle, WA 98195 USA (e-mail: conti@math.unipd.it).

Jiaxin Li is with the Department of Mathematics, University of Padua, 35131 Padua, Italy (e-mail: jiaxin.li@studenti.unipd.it).

Stjepan Picek is with the Institute for Computing and Information Sciences, Radboud University, 6525 EC Nijmegen, The Netherlands (e-mail: stjepan.picek@ru.nl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TDSC.2025.3543093, provided by the authors.

Digital Object Identifier 10.1109/TDSC.2025.3543093

the target data point, they infer the membership of this target data point in a specific target model. Those N shadow models have the same model structure and hyperparameters for training. Therefore, training multiple models with resampled training sets from the same dataset is possible in Bagging and training N shadow models for the previous MIA. In summary, this work explores whether the vulnerable data point found under one MIA and one target model is also vulnerable to other MIAs and target models. Additionally, the vulnerable data points under one MIA and one target model could have a certain degree of randomness, as the worst accuracy of the binary attack model is 0.5 rather than 0. We eliminate this randomness by exploring multiple target models and MIAs. Finally, previous methods mainly focus on the data point's vulnerability while the data point is in the training data regardless of the test data. However, the vulnerability of the test data is closely related to the performance of MIAs, and exploring vulnerable data points in the test data provides another perspective to understand the vulnerability. Therefore, we propose our method to overcome those gaps.

We explore the vulnerability of data points under multiple MIAs and target models. As an essential step, we attack each target model with multiple MIAs. Note that these target models are trained with resampled subsets of the same dataset by conducting numerous splits. Except for exploratory experiments, they have the same size of data points, identical model structure, and same hyperparameters as default. This approach allows us to eliminate the influence of the effect of size, model structure, and hyperparameters while analyzing a single data point's vulnerability within these target models. In addition, we define eight metrics about the data point's exposure rate and MIA's inference rate, as discussed in Section III, to formally analyze the data point's vulnerability and MIA's inference correctness and tendency. The main reason for new definitions is that previous methods [8], [9], [10] on the vulnerable data points do not provide suitable metrics to evaluate the data point's vulnerability under multiple MIAs and target models from their definitions and our observation (Section VI-E). Moreover, to make it convenient and fast to attack target models with multiple MIAs, we designed and implemented the VMIAP,[1] a scalable and flexible platform to conduct various MIAs against target models. Finally, we employ previous methods and our devised metrics to identify vulnerable data points. It is reasonable that factors that impact the average attack accuracy of MIAs will influence the vulnerability of data points since vulnerability is related to the attack accuracy. To further explore other influencing factors of data point vulnerability, we conduct experiments to retrain target, shadow, and attack models to recapture each data point's vulnerability. Retraining a model in this work means initializing the parameters with a different random seed and shuffling the data during training while keeping the training data, model structure, and hyperparameters unchanged to avoid their influence. The randomness that comes from retraining target, shadow, and attack models separately has different impacts on detecting vulnerable data points, which is reflected in the modification range of metric values for measuring the vulnerability.

Our main contributions are:

1) We define metrics related to the data point's exposure rate and MIA's inference rate to analyze data points' vulnerability and MIA's inference correctness and tendency. Our metrics capture vulnerable data points under multiple MIAs and target models, while previous methods cannot, as shown in Fig. 8. Besides, MIA has an inference tendency to some data points despite a low overall inference accuracy of 0.5, as shown in Table VII.
2) We design and implement a scalable and flexible platform VMIAP for attacking different target models with multiple types of MIAs. We detail the platform in Section IV and compare it with other previous platforms in Table V.
3) The modified average attack accuracy of MIAs on the different overlapping gaps (Fig. 2), datasets, and model structures (Table VI) in Section VI-A shows that the overlapping gap, dataset, and model structure are the main factors affecting attack performance. Retraining target, shadow, and attack models have different impacts on the vulnerability of data points. From the perspective of MIAs, when retraining the target model with a high overfitting level, the vulnerability of data points is less modified than when retraining shadow and attack models.

II. Background and Threat Model

The first four subsections present the necessary background information about notations, machine learning, previous methods, and the Membership Inference Attack Game. Finally, we discuss the threat model.

A. Notation

We summarize in Table I the notations we use in the rest of the paper.

B. Machine Learning

We consider the target models based on neural networks and train the attack model as classical or ensemble models to implement MIAs. We introduce those types of models in this section.

Neural Network. The target model we focus on is a supervised model for classification tasks. The target dataset D_{target}, drawn from the underlying distribution $\mathbf{D}$, comprises many data points. The x and y represent the data point's feature vector and label, respectively. The unique values of y indicate categories of data points. Before training, we divide the target dataset into two sub-datasets D_{target}^{train} and D_{target}^{test} for training and testing the target model f_{θ_t}, which is a neural network. The f_{θ_t} obtains the feature vector x and predicts its label based on the maximum probability it belongs to each category. The training of f_{θ_t} makes its outputs for data points in D_{target}^{train} close to their ground truths. The learning algorithm and loss function guide the adjustment of the model's parameters θ_t. The iterative update of θ_t is along the opposite direction of the gradient to minimize the loss of a

[1][Online]. Available: https://github.com/fight-think/Various_Membership_Inference_Attacks_Platform_open_source

TABLE I
NOTATIONS

Notation	Description
$\mathbf{D}$	data distribution
D_{target}	contains two sub-datasets D^{train}_{target} and D^{test}_{target}
D^{train}_{target}	training data of the target model
D^{test}_{target}	test data of the target model
D_{shadow}	contains two sub-datasets D^{train}_{shadow} and D^{test}_{shadow}
D^{train}_{shadow}	training data of the shadow model
D^{test}_{shadow}	test data of the shadow model
D_{re_shadow}	holds two sub-datasets $D^{train}_{re\_shadow}$ and $D^{test}_{re\_shadow}$
$D^{train}_{re\_shadow}$	relabel D^{train}_{shadow} with the target model
$D^{test}_{re\_shadow}$	relabel D^{test}_{shadow} with the target model
$z = (x, y)$	one data point from the dataset
f_θ	model f_θ with parameters θ
$f_\theta(x)$	output of the model f_θ on x
θ_t	parameters of the target model
θ_s	parameters of the shadow model
θ_{re_s}	parameters of the relabeled shadow model
$r(z)$	privacy risk score of data point z
E	extra knowledge the adversary holds
I	adversary's utilization strategy on knowledge E
$A_{f_{\theta_t},E,I}(x, y)$	adversary's prediction on (x, y)
n	the number of MIAs for each target model
k	the number of target models for each run, also equals the split numbers of the original dataset
$MT(x, y)$	the number of target models whose training data contains (x, y) among k target models
$NMT(x, y)$	the number of target models whose test data contains (x, y) among k target models
MN	the number of data points that occurred in the training data of k target models
NMN	the number of data points that occurred in the test data of k target models
$b_j(x, y)$	the membership ground truth for (x, y) in j-th target model of k target models
$b'_{(j,i)}(x, y)$	the membership prediction of i-th MIA for (x, y) in j-th target model
$B()$	the indicator function, $B(true) = 1$, $B(false) = 0$

batch of data points in D^{train}_{target}. The model will reuse D^{train}_{target} for updating parameters several times. After the training of f_{θ_t}, the model will evaluate on D^{test}_{target}, which has no overlapping data points with D^{train}_{target}. The prediction accuracy of f_{θ_t} on D^{test}_{target} presents the model's generalization ability. The model with a high prediction accuracy on D^{test}_{target} is more effective in practice.

Classical Model: SVM [17] and Linear Regression [18] are two classical models in machine learning. SVM aims to find the best decision boundary to separate data points into different categories while maximizing the distance between the decision boundary and the nearest data points of each category. Linear regression is a statistical method used to model the relationship between one or more input variables and a continuous output variable, which assumes a linear relationship between the inputs and the output. It aims to find the best-fitting line or hyperplane that minimizes the loss. Since classifying members and non-members is a binary task, it is feasible to use those two classical models.

Ensemble Model: Apart from Bagging mentioned in Section I, we utilize XGBoost [19] to train the attack model. XGBoost iteratively trains the weak leaner, typically the decision tree, to correct errors of the previous predictions by minimizing a loss function measuring the difference between current predictions and ground truth. The formal name of this process is Gradient Tree Boosting. XGBoost applies L1 and L2 regularizations related to the weights of learners to reduce overfitting. To enable the operation on billions of examples in distributed or memory-limited settings, XGBoost uses a novel tree learning algorithm for handling sparse data, and a theoretically justified weighted quantile sketch procedure enables handling instance weights in approximate tree learning.

C. On Finding Vulnerable Data Points

Previous works mention or utilize the neighbors-based method, SUOD, privacy risk score, and Shapley value to find vulnerable data points. Thus, we briefly introduce those concepts.

Neighbors-Based Method: The MIA generally performs better on the overfitted target model. To overcome the low performance of MIA on a well-generalized model, Long et al. [8] propose to select a few vulnerable data points from the training data and implement MIA against them. The primary step for selecting is counting the number of neighbors of each data point on the combination feature space represented by outputs of reference models. They choose the data points with fewer neighbors as the vulnerable ones.

SUOD: Outlier detection with multiple unsupervised detectors is common for unlabeled data and stability-required scenarios [20]. SUOD [12] is a three-module acceleration framework that leverages random projection, pseudo-supervised approximation, and balanced parallel scheduling for the scalability of outlier detection. The random projection module generates lower subspaces for high-dimensional datasets while preserving distance relationships. The pseudo-supervised approximation module approximates fitted unsupervised models by lower-cost supervised regressors for fast prediction of unseen data. The balanced parallel scheduling module forecasts models' training and prediction costs with high confidence, so the scheduler assigns a nearly equal number of tasks among workers for efficient parallelization. With those three modules, SUOD expedites the training and prediction with many unsupervised detection models for outlier detection.

Privacy Risk Score: Song et al. [9] define the privacy risk score (r) of an input (z) with respect to f_{θ_t} as the posterior probability of being a member of D^{train}_{target} conditioned on f_{θ_t}'s behavior ($O(f_{\theta_t}, z)$) on the input. We denote the privacy risk score as $r(z) = P(z \in D^{train}_{target} | O(f_{\theta_t}, z))$. They further expand this expression based on Bayes' theorem for a convenient computation.

Shapley Value: The Shapley value [10] is a concept originating from cooperative game theory. It provides a fair and systematic way to allocate each player's contribution cooperatively. In machine learning, the Shapley value is used to estimate the contribution of each training data point to the model's utility. The computation of the Shapley value for a data point estimates the influence of that record on the model's utility using the leave-one-out training approach. The formula for Shapley values involves assessing the marginal contribution of each data record to the model's utility when added to a randomly chosen subset of the training data.

D. Membership Inference Attack Game

We use the definition of the Membership Inference Attack Game presented by Yeom et al. [21], and followed in the works of Carlini et al. [11] and Jayaraman et al. [22].

Definition 1 (Membership Inference Attack Game): The game between a challenger C and an adversary A:

1) The challenger samples a training dataset D^{train}_{target} from the underlying distribution **D** and trains a target model f_{θ_t} based on D^{train}_{target}.
2) The challenger flips a bit b. If $b = 0$, the challenger samples a data point (x, y) from the distribution **D**. If $b = 1$, the challenger randomly selects a data point (x, y) from the training dataset D^{train}_{target}.
3) The challenger sends (x, y) to the adversary.
4) The adversary queries the target model f_{θ_t} and has some extra knowledge E. Then, the adversary outputs the prediction $b' = A_{f_{\theta_t},E,I}(x, y)$ with a specific utilization strategy I of knowledge E.
5) Outputs 1 if $b = b'$, else 0.

If $b = b'$, the game's output is 1, which means the adversary wins this game on a single data point. Otherwise, the challenger successfully defends. In realistic experiments, we act as both the challenger and the adversary to test the performance of adversaries. The evaluation depends on the result of a specific adversary on a large number of data points.

E. Threat Model

As per the definition of the Membership Inference Attack Game in the previous section, the adversary has access to the target model f_{θ_t} and some extra knowledge E. The adversary aims to win the Membership Inference Attack Game on data points as often as possible. If the adversary obtains extra knowledge E and changes the utilization way I to information, we regard it as a new type of MIA in our paper. The basic ideas of MIAs follow previous works. Then, we change the details of MIAs' implementations to construct variants of previous MIAs, which extends possible related MIAs to evaluate our newly defined metrics. In this way, we obtain 54 different MIAs to explore the vulnerability of data points. To formally explain each MIA's threat model, we expose multiple MIAs' extra knowledge and utilization strategies, which distinguish different MIAs.

Extra knowledge: Given one data point (x, y), the adversary queries the target model f_{θ_t} and obtains its output. Then, the adversary returns the prediction $(A_{f_{\theta_t},E,I}(x, y))$ about the membership of this data point. Following the strategy of the shadow model [7], the adversary obtains a shadow dataset D_{shadow}, which has no overlapping data points with the training and test data of the target model, to train and test a shadow model f_{θ_s} for imitating the behavior of the target model. Besides, the adversary knows the model structure and hyperparameters of the target model f_{θ_t}. In summary, the label y, the target model's output $f_{\theta_t}(x)$, the shadow dataset D_{shadow}, the model structure, and the target model's hyperparameters are the adversary's extra information to infer the membership of the data point (x, y).

Basic process: The adversary trains the shadow model with the extra knowledge to mimic the behavior of the target model.

TABLE II
ELEVEN BASIC FEATURES

Number	Basic Features
①	target model's output vector $f_{\theta_t}(x)$
②	parameters' gradients on x
③	target model's logits (output before SoftMax) on x
④	maximum probability value in the output vector
⑤	CrossEntropyLoss in the Pytorch implementation
⑥	probability of ground truth
⑦	entropy of the output vector
⑧	normalization value of the output vector's entropy
⑨	Mentr value defined by Song et al. [9]
⑩	hingle loss mentioned in [11]
⑪	logit-scaled confidence defined in [11]

To better imitate it, the adversary also relabels the shadow dataset by evaluating it with the target model and uses this relabeled shadow dataset (D_{re_shadow}) to train an additional model, which we call relabeled shadow model ($f_{\theta_{re\_s}}$) in this work. With a shadow or relabeled shadow model, the adversary extracts the attack feature to distinguish the training and test data. The attack feature is a metric directly or indirectly calculated from the model's output. For the shadow or relabeled shadow model, the adversary knows the membership of each data point in the shadow or relabeled shadow dataset. Therefore, the adversary constructs the attack dataset with the attack feature and the membership label. With the attack dataset, the adversary trains the attack model or selects the threshold to determine the membership of the data point. More explanation about the process of implementing multiple MIAs is in Section IV.

Utilization strategies: The utilization strategies are related to obtaining the attack feature from the model's output and using it to predict membership. For different MIAs in this work, there is a subtle difference in the attack feature, utilization of the attack feature, or the source of the attack dataset (shadow model or relabeled shadow model). We derive the attack features by amalgamating or directly selecting from eleven basic features. Table II provides those eleven basic features.

There are classifier-based and threshold-based methods for using the attack feature to predict membership. In the classifier-based method, the adversary trains one classifier to infer membership with the attack dataset. Here, we select four types of classifiers: SVM, Linear Regression, XGBoost, and a shallow MultiLayer Perceptron (MLP) [23]. Those four types of classifiers contain classical, ensemble, and neural network models, which bring diversity to MIAs and make comparison and evaluation straightforward by only changing the classifier type. Table III lists the attack features utilized in the classifier-based method. The labels from ① to ⑪ are from Table II. Each row in the table represents one attack feature combined with eleven basic features in Table II. Generally, we connect multi-valued feature ① with other single-valued features (④ to ⑪) as the attack features. There are fifteen attack features for the classifier-based method. We regard the change in the type of classifier as a new type of MIA. Therefore, there are 60 (15×4) classifier-based MIAs.

For the threshold-based method, the adversary finds a suitable comparison threshold for the single-valued attack feature. Eight

TABLE III
FIFTEEN ATTACK FEATURES OF THE CLASSIFIER-BASED METHOD

Attack Features
①
②
③
Ⓢ
①, ②
①, ③
①, Ⓢ
②, ③
②, Ⓢ
③, Ⓢ
①, ②, ③
①, ②, Ⓢ
①, ③, Ⓢ
②, ③, Ⓢ
①, ②, ③, Ⓢ

The labels (① to ⑪) are from Table II. For convenience, we denote features from ④ to ⑪ as Ⓢ.

basic features labeled from ④ to ⑪ are possible attack features. ① to ③ are excluded as the multi-valued feature unsuitable for threshold comparison. Besides, the source of those eight basic features could be the shadow or relabeled shadow models. Therefore, there are 16 (8×2) threshold-based MIAs. Besides, we include the gap MIA, proposed by Yeom et al. [21], for comparison. The gap MIA classifies the data point that the target model correctly predicts its class as a member. Otherwise, this MIA regards it as a non-member. In summary, we currently consider 77 (60+16+1) MIAs. The previous works do not cover all the details about 77 MIAs' implementations. Specifically, we use more than one type of attack model, combine the output vector and other basic features as the attack features, and utilize the relabel strategy for threshold-based methods. This means that we expand MIAs from previous works. The expansion explores more possible MIAs for our analysis and provides more for the label-only condition with the relabel strategy.

III. NEW METRICS FOR DATA POINT AND MIA

We give a brief introduction in Section III-A. Section III-B formulates metrics for the data point's exposure rate. Section III-C interprets metrics for MIA's inference rate.

A. Brief Outline

Previous works detect vulnerable data points under one MIA and one target model. They focus on improving the overall attack accuracy rather than analyzing the vulnerability of each data point. From the analysis of Section VI-E and their definitions, we find they are unsuitable for describing vulnerable data points under multiple MIAs and target models. Therefore, we define four metrics for the data point's exposure rate to assess the data points' vulnerability. Two new metrics are suitable for measuring the vulnerability of the data point under multiple MIAs and target models. The other two metrics are basic elements for those two metrics. Besides, we formulate four metrics related to MIA's inference rate to investigate the inference correctness and tendency of the MIA. We define the notations of the following formulas in Tabel I.

B. Data Point's Exposure Rate

For the data point (x, y) in the training data of j-th target model, we define the percentage of correct inferences among n MIAs as its Member Exposure Rate (MER), formulated in (1).

$$MER_j(x,y) = \frac{\sum_{i=1}^{n} B(b'_{(j,i)}(x,y) == 1)}{n}. \quad (1)$$

In k random divisions of the dataset, the data point (x, y) is in the training data of $MT(x, y)$ target models. We average the $MER_j(x, y)$ values among those target models to define the Average Member Exposure Rate (AMER), formulated in (2).

$$AMER(x,y) = \frac{\sum_{j=1}^{MT(x,y)} MER_j(x,y)}{MT(x,y)}. \quad (2)$$

If this data point (x, y) is in the test data of j-th target model, we define the percentage of correct inferences among n MIAs as its Non-Member Exposure Rate (NMER). The difference between NMER and MER in (1) is the judgemental condition changes to $b'_{(j,i)}(x, y)$ equals 0. Similarly, we average the $NMER_j(x, y)$ values among $NMT(x, y)$ target models to define the Average Non-Member Exposure Rate (ANMER) of data point (x, y).

From definitions of $AMER(x, y)$ and $ANMER(x, y)$, they average the MIAs' inference accuracies over target models while the data point is in the target models' training and test data. Higher inference accuracy means more data points are correctly inferred. Therefore, we use those two metrics to determine vulnerable data points while in the training and test data separately.

The above definitions consider the perspective of the data point to analyze the vulnerability. We also analyze the correctness of the inference from the perspective of MIA. Let us consider that a specific MIA correctly infers the existence of a single data point in a target model. Can this MIA infer the data point's presence while attacking other target models? What is the inference correctness of this MIA to other data points? Does this MIA have an inference tendency to a part of data points? To understand those questions, we define metrics about MIA's inference rate from the perspective of the MIA to analyze its inference correctness and tendency to data points.

C. MIA's Inference Rate

For the data point (x, y) in the training data of $MT(x, y)$ target models, the i-th MIA infers the membership of this data point in those target models. We define the percentage of correct inferences as the Member Inference Rate (MIR) of i-th MIA to the data point (x, y), formulated in (3).

$$MIR_i(x,y) = \frac{\sum_{j=1}^{MT(x,y)} B(b'_{(j,i)}(x,y) == 1)}{MT(x,y)}. \quad (3)$$

Including the data point (x, y), MN different data points occurred in the training data of k target models. Therefore, we average the $MIR_i(x, y)$ values of those data points as

TABLE IV
AN EXAMPLE OF CALCULATING THE (AVERAGE) MEMBER INFERENCE RATE AND (AVERAGE) MEMBER EXPOSURE RATE

	Member											
	D_1				MIR	*D_2*					MIR	AMIR
M_1	1	0	0	1	0.5	1	1	0	1	0	0.6	0.55
M_2	1	0	1	1	0.75	1	0	1	1	1	0.8	0.775
MER	1.0	0.0	0.5	1.0		1.0	0.5	0.5	1.0	0.5		
AMER	(1.0+0.0+0.5+1.0)/4					(1.0+0.5+0.5+1.0+0.5)/5						

the Average Member Inference Rate (AMIR) of the i-th MIA, formulated in (4).

$$AMIR_i = \frac{\sum_{m=1}^{MN} MIR_i(x_m, y_m)}{MN}. \tag{4}$$

If the data point (x, y) is in the test data of $NMT(x, y)$ target models, we define the percentage of correct inferences from i-th MIA as its Non-Member Inference Rate (NMIR) to the data point (x, y), which modify the judgemental condition to $b'_{(j,i)}(x, y)$ equals 0. Including the data point (x, y), NMN different data points occurred in the test data of k target models. Similarly, we average the $NMIR_i(x, y)$ values of those data points as the Average Non-Member Inference Rate (ANMIR) of the i-th MIA.

With those metrics for MIA's inference rate, we answer questions including MIA's inference correctness to different data points and MIA's inference tendency to a part of data points. We distinguish the functions for defining metrics and the outputs of functions here. The italics represent functions, including MER, $AMER$, $NMER$, $ANMER$, MIR, $AMIR$, $NMIR$, and $ANMIR$. We uniformly use non-italicized letters outside metric definition formulas like MER, AMER, NMER, ANMER, MIR, AMIR, NMIR, and ANMIR to represent functions' outputs or metrics.

D. Explanation of the Exposure and Inference Rates

Considering that definitions of data point's exposure and MIA's inference rates are abstract, we use a simple example to explain those metrics. Due to the similarity between metrics for a data point in the training and test data, we only illustrate the (Average) Member Inference Rate and (Average) Member Exposure Rate in Table IV.

In Table IV, *M_i* represents the i-th MIA and *D_i* represents the i-th data point. The number 0 represents a data point in the training data of one target model but is inferred as a non-member by the MIA. The number 1 means the data point is in the training data of one target model and is predicted as a member by the MIA, which is a correct inference. *D_1* is in the training data of 4 target models, and *D_2* is in the training data of 5 target models. We obtain the number of target models by counting the 0 or 1 inference values for each data point. The number of MIAs is 2 (*M_1* and *M_2*), and the number of data points is 2 (*D_1* and *D_2*). These numbers in the table are placeholders used to illustrate metrics and do not represent actual quantities in our experiments. The table shows that MIR, AMIR, MER, and AMER are calculated based on the inference values.

IV. VARIOUS MEMBERSHIP INFERENCE ATTACKS PLATFORM (VMIAP)

Previous code implementations of membership inference attacks are unsuitable for applying different MIAs to target models. In addition, they do not analyze a single data point's vulnerability under multiple MIAs and target models, which is inconvenient for our research exploration. Thus, we designed and implemented the VMIAP, which is scalable and flexible, to support our research about vulnerable data points under multiple MIAs and target models. In Section IV-A, we explain different types of MIAs. We provide the details of platform implementation in Section IV-B.

A. Different Types of MIAs

The membership inference attack, proposed by Shokri et al. [7], undergoes a series of developments [8], [9], [14], [15], [21]. As mentioned in Section II-E if the adversary obtains extra knowledge and changes how to utilize that knowledge, we regard it as a new type of MIA in our paper. The extra knowledge contains the label y, the target model's output $f_{\theta_t}(x)$, the shadow dataset D_{shadow}, the model structure, and the target model's hyperparameters. The adversary utilizes extra information and alters the utilization strategies to implement multiple MIAs.

The utilization strategies comprise two categories. One is predicting membership by comparing the metric value with a selected threshold. The other is the combination or direct selection of eleven basic features in Table II for attack features in Table III for classifier-based MIAs. For the threshold-based MIA, the metric's selection and source (shadow or relabeled shadow models) determine one MIA. Apart from better imitating the target model, the relabeled shadow model also deals with the label-only [14] situation, where the model only exposes the prediction label. By changing the utilization of extra knowledge and considering the gap MIA [21], which predicts the membership based on the prediction correctness, we implement 77 different MIAs in the VMIAP for analyzing the vulnerability of data points under multiple MIAs and target models, as mentioned in Section II-E.

B. Platform Implementation and Running

The code implementation is divided into multiple parts to make the platform scalable and flexible. Those parts comprise dataset preparation, parameter import, model training, attack feature extraction, classifier-based MIA implementation, threshold-based MIA implementation, and vulnerability analysis. The parameter import part prepares hyperparameters after dividing the dataset with the dataset preparation part. They are vital for training target, shadow, and relabeled shadow models. The attack feature extraction part obtains attack features for multiple MIAs. The implementations of classifier-based and threshold-based MIAs handle attacks. The vulnerability analysis combines the inference results from multiple MIAs and target models. We only need to change and adapt the corresponding parts of our platform to expand datasets, models, and MIAs. The

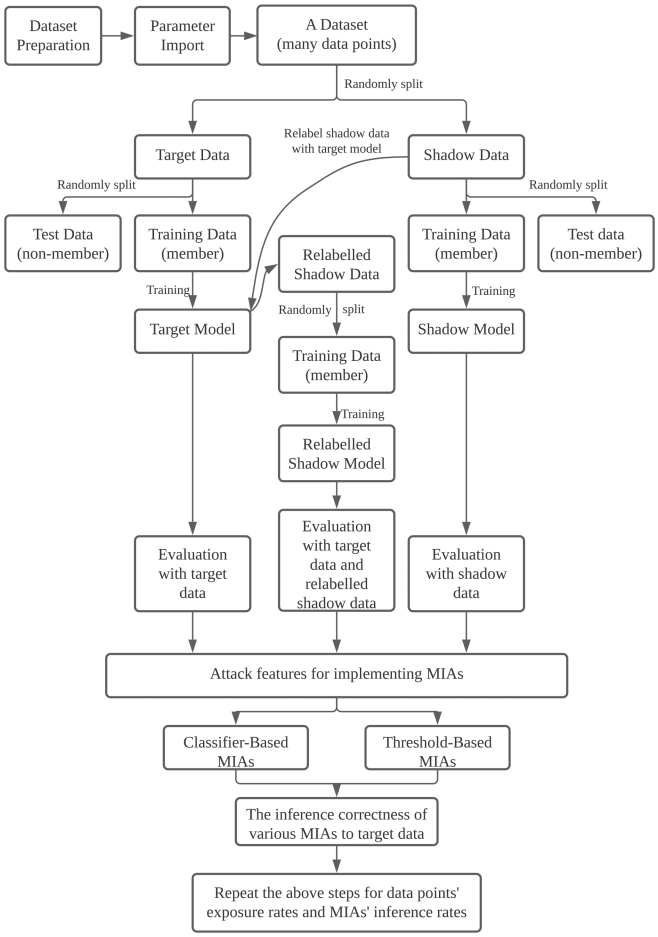


Fig. 1. The process of the VMIAP for implementing multiple MIAs against target models.

framework is implemented in Python, and the main package is Pytorch.[2]

Fig. 1 shows the general process of the VMIAP for multiple MIAs and target models. The figure contains the parts we mentioned in the previous paragraph. For the division of the dataset, we utilize settings to control the split rates and the number of selected data points. Given a fixed number of data points from one dataset, we assign *target shadow rate*, *target split rate*, and *shadow split rate* to decide the proportion of the target dataset among all data points, the proportion of training data in the target dataset, and the proportion of training data in the shadow dataset. In most experiments, we set those three rates as 0.5, 0.5, and 0.5 for the shadow model's better imitation and fair evaluation of attack accuracy on training and test data. After dealing with one time of dataset division, we keep dividing, training, extracting, and analyzing multiple times for multiple target models. We define *Split_num* as the division times, which has the same meaning as k defined in Table I. We measure the vulnerability of each data point in the dataset based on the inference results of 77 MIAs on *Split_num* target models with our metrics and previous

[2][Online]. Available: https://pytorch.org/

TABLE V
COMPARISON OF PLATFORMS TO IMPLEMENT MIAS

Platform	Considered features (quantity)	Analysis on a single data point	Parallelism of attacks	Flexibility of expanding attacks
Shokri et al.[3]	2	✘	low	low
Salem et al.[4]	4	✘	low	low
Shokri et al.[5] (ML Privacy Meter)	7	✘	middle	middle
Liu et al.[6] (ML Doctor)	3	✘	middle	middle
Our VMIAP	11	✔	high	high

methods. In experiments, the minimum division time is 20. In each division, one data point has the same probability (0.25) in target and shadow models' training and test data. According to the related knowledge of probability theory [24], most data points will occur in training and test data of the target model five times (20*0.25=5), which is feasible to evaluate the data point's vulnerability. However, the data point's occurrence times in training or test data may vary greatly when evaluating the influence of division rates on the results by setting rates different from 0.5, 0.5, and 0.5. While training models, we train the target and shadow models first. Then, we relabel the shadow dataset with the target model to train the relabeled shadow model. With the relabeled shadow model, we obtain predictions from it with the target and relabeled shadow datasets. We separately input the target and shadow datasets into target and shadow models to get models' predictions for attack feature extraction. The platform will record the inference correctness results of multiple MIAs on each data point in the data of each target model, providing results for analyzing the data point's vulnerability.

To clearly describe the process of attacking multiple target models with different MIAs, we show the process flow of the VMIAP in Algorithm 1. $Split$ is the operation that divides the dataset into target and shadow datasets. $Train$ returns the target, shadow, and relabeled shadow models by training with the dataset and other settings. $Predict$ infers the label of the data point with the model. $Extract$ obtains the model output to the dataset and calculates the attack features for MIAs. Fit trains the attack models or finds the suitable thresholds by fitting the attack features from shadow and relabeled shadow models. $Eval$ evaluates the attack performance on the attack features from the target dataset. $Analyze$ aggregates the inference correctness of data points under multiple MIAs and target models to analyze the vulnerability of each data point.

Table V compares our platform with previous platforms to implement MIAs from four perspectives, including the number of considered features, analysis on a single data point, parallelism of attacks, and flexibility of expanding attacks. We can observe that our VMIAP considers more features when analyzing a single data point. Besides, we extract features for MIAs in parallel instead of separately in other platforms. For example, the process of getting signals in the ML privacy meter [25] extracts confidence and loss within different functions with redundant

[3][Online]. Available: https://github.com/csong27/membership-inference
[4][Online]. Available: https://github.com/AhmedSalem2/ML-Leaks
[5][Online]. Available: https://github.com/privacytrustlab/ml_privacy_meter
[6][Online]. Available: https://github.com/liuyugeng/ML-Doctor

computation. Finally, the modification to the extraction process can easily expand MIAs on our platform by carrying features on two types of returned variables: one contains single-valued features, and the other contains vector-based features. However, other platforms require separate handling of extended features. For example, the ML doctor [26] has separate functions to define attack models for different vector-based features, which could be unified.

V. Experimental Settings

We designed and ran experiments related to three datasets. Our experiments contain two parts. The first part is implementing various MIAs on different target models to observe the data point's vulnerability under multiple MIAs and the inference tendency of the MIA to various data points. The second part compares vulnerable data points found by different methods and reruns each experiment for comparison. In Section V-A, we describe datasets used for experiments. In Section V-B, we detail the content of the experiments.

A. Datasets

We use three datasets: CIFAR-10, MNIST, and PURCHASE-100. We do not use all the data points in the dataset and randomly select 40000 data points from each dataset. There are two reasons for selecting stationary 40000 data points. The first reason is that fixed data points are vital for analyzing data points' vulnerability after the repetitive division of those data points for training models and attacking. The second reason is that Shokri et al. explored the performance of attacking while training target models with 10000 data points. We follow their setting and choose 40000 data points to train and test target and shadow models. More precisely, among 40000 data points, we use non-overlapping 10000 data points for training the target model, testing the target model, training the shadow model, and testing the shadow model, respectively. Among the datasets, the CIFAR-10 and the MNIST are almost balanced. The PURCHASE-100 is unbalanced, which means the number of data points in each class varies significantly. The PURCHASE-100 is from Kaggle's "acquire valued shoppers" challenge dataset containing several thousands of individuals' shopping histories.[7] In the work of Shokri et al. the authors use a converted purchase dataset with 600 binary features, which represents whether a user purchases a specific item or not [7]. The conversion refers to the transformation of features, not the encryption or restriction of data or information access. The competition with the raw dataset predicts which shoppers are most likely to repeat purchases. They transfer this dataset for shopper's purchase style classification. The PURCHASE-100 has 100 categories, which means 100 purchase types among shoppers. We use the same PURCHASE-100 dataset as in the work of Shokri et al.

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

B. Methodology

In this section, we explain the details of the experiments and provide the settings and model architectures. In Section V-B1, we present experimental steps, models' structures, and settings. Then, we discuss finding and comparing vulnerable data points in Section V-B2.

1) Multiple MIAs Against Target Models: For three datasets mentioned in Section V-A, we select LeNet, ResNet18, CNN, shallow MLP, and deeper MLP as the network structures for experiments as they are also selected in previous works [7], [11], [14], [15], [21] and are commonly used for classification tasks. For each combination of the dataset, model structure, and other settings, we follow the process of the VMIAP mentioned in Section IV-B to train *Split_num* target models and attack each target model with 77 MIAs mentioned in Section IV-A. Then, we analyze the vulnerability of each data point under 77 MIAs and *Split_num* target models.

The architecture and hyperparameters are the same for a dataset to get *Split_num* target models. We choose two model architectures for each dataset to explore the influence of model architecture. The Appendix, available online, provides the model structures' details and training settings in Table C.1, available online.

2) Finding and Comparing Vulnerable Data Points: As mentioned in Section III-B, we determine vulnerable data points with the help of AMER and ANMER. Those two metrics are related to the actual inference correctness of data points in multiple target models under multiple MIAs. To compare and test the effectiveness of previous methods, we pick l (40 or 400) data points out of 40000 as vulnerable data points based on newly defined metrics and previous evaluation methods. We select 40 or 400 data points because of the relatively small number of vulnerable points, and it simplifies comparisons when each approach uses a consistent number of them. Besides, selecting vulnerable data points of two quantities provides a new perspective to evaluate the vulnerability of data points. The steps of finding vulnerable data points are identical for various datasets, and the following paragraphs describe how we use different methods to determine vulnerable data points.

Our new metrics: Following the definitions, we compute the AMER of each data point and select l (40 or 400) vulnerable data points based on this value for each model structure and dataset. Similarly, we obtained l (40 or 400) vulnerable data points with ANMER. Besides, we repeat the training of target, shadow, and attack models together or separately to obtain another two sets of vulnerable data points with two new metrics. We try to observe the difference between vulnerable data points while retraining models to explore their impact on the vulnerability. Hence, we obtain several sets of l (40 or 400) vulnerable data points for each dataset and model structure.

Other methods: For the neighbors-based [8] method, SUOD [12], privacy risk score [9], and Shapley value [10], we calculate the corresponding metrics to measure the vulnerability and select l (40 or 400) out of 40000 data points based on the metrics for each dataset division. After $Split\_num$ divisions of the dataset, we obtain $Split\_num$ sets of l (40 or 400) data

TABLE VI
MIAs' ATTACK PERFORMANCE ON TARGET MODELS WITH DIFFERENT CONFIGURATIONS AND ACCURACY

	Configurations				Metrics								
Row	Dataset	Model Structure	Split_num	Rates (target shadow rate, target split rate, shadow split rate)	Avg train acc	Avg test acc	Avg dif acc	Avg MIA acc	MIA acc var	MIA acc med	MIA acc max	MIA acc min	Difference MIA acc (max - min)
1	CIFAR-10	LeNet	20	(0.5, 0.5, 0.5)	0.676	0.450	**0.2260**	**0.573**	0.00227	0.585	0.636	0.500	0.136
2	CIFAR-10	LeNet	40	(0.5, 0.5, 0.5)	0.676	0.449	**0.2266**	**0.572**	0.00219	0.580	0.632	0.500	0.132
3	CIFAR-10	LeNet	40	(0.5, 0.8, 0.8)	0.653	0.482	**0.1712**	**0.761**	0.00490	0.794	0.810	0.452	0.358
4	CIFAR-10	LeNet	40	(0.8, 0.5, 0.5)	0.654	0.479	**0.1751**	**0.557**	0.00132	0.568	0.602	0.500	0.102
5	CIFAR-10	ResNet18	20	(0.5, 0.5, 0.5)	0.999	0.719	**0.2805**	**0.679**	0.00718	0.713	0.754	0.494	0.260
6	MNIST	CNN	20	(0.5, 0.5, 0.5)	0.964	0.950	**0.0135**	**0.506**	0.000019	0.505	0.513	0.499	0.014
7	MNIST	CNN	40	(0.5, 0.5, 0.5)	0.968	0.954	**0.0140**	**0.505**	0.000017	0.505	0.513	0.499	0.014
8	MNIST	CNN	40	(0.5, 0.8, 0.8)	0.961	0.956	**0.0053**	**0.750**	0.007250	0.782	0.800	0.506	0.294
9	MNIST	CNN	40	(0.8, 0.5, 0.5)	0.963	0.955	**0.0079**	**0.504**	0.000006	0.504	0.507	0.400	0.007
10	MNIST	ResNet18	20	(0.5, 0.5, 0.5)	0.948	0.942	**0.0049**	**0.503**	0.000004	0.502	0.507	0.498	0.009
11	PURCHASE-100	Shallow MLP	20	(0.5, 0.5, 0.5)	0.995	0.586	**0.408**	**0.779**	0.01872	0.850	0.880	0.485	0.395
12	PURCHASE-100	Shallow MLP	40	(0.5, 0.5, 0.5)	0.991	0.584	**0.407**	**0.779**	0.01865	0.849	0.879	0.496	0.383
13	PURCHASE-100	Shallow MLP	40	(0.5, 0.8, 0.8)	0.984	0.647	**0.338**	**0.865**	0.00430	0.893	0.913	0.630	0.283
14	PURCHASE-100	Shallow MLP	40	(0.8, 0.5, 0.5)	0.979	0.639	**0.340**	**0.728**	0.01371	0.770	0.847	0.496	0.351
15	PURCHASE-100	Deeper MLP	20	(0.5, 0.5, 0.5)	0.779	0.615	**0.164**	**0.548**	0.00062	0.549	0.601	0.499	0.101

points. Then, we count the number of occurrences of each data point in those $Split\_num$ sets and select l (40 or 400) frequent ones as vulnerable data points. Finally, we repeat this process to get two sets of l (40 or 400) vulnerable data points for each method. We refer to original papers for more detail apart from Section II-B.

VI. RESULTS AND DISCUSSION

In this section, we present our experimental results and discuss the findings. In particular, we include parts about the accuracy of target models, attack performance of MIAs (Section VI-A), exposure rate of data point (Section VI-B), inference rate of MIA (Section VI-C), exploratory experiments (Section VI-D), and vulnerable data points comparison (Section VI-E).

A. Target Models' Accuracy and MIAs' Attack Performance

Table VI shows statistical metrics about target models' settings, accuracies, and MIAs' attack performance. Each row represents the result for a given configuration (*Dataset*, *Structure*, *Split_num*, and *Rates*). We train *Split_num* target models by creating resampled data subsets through multiple divisions of the same dataset. They have identical hyperparameters under one configuration. The *Rates* column comprises three elements: *target shadow rate*, *target split rate*, and *shadow split rate*. We use those three rates to partition the dataset into four subsets for training and testing target and shadow models, as elaborated in Section IV-B. Besides, we calculate the average training accuracy (*Avg train acc*), the average testing accuracy (*Avg test acc*), and the average accuracy difference between training and testing accuracy (*Avg dif acc*) of *Split_num* target models.

Table VI shows the average accuracy of *Split_num* target models. First, the larger the gap between the training and testing accuracy of the target model, the better MIAs' attack performance is in general. The accuracy gap of MNIST (0∼0.01) is smaller than CIFAR-10 (0.17∼0.28) and PURCHASE-100 (0.16∼0.40). The attack performance of CIFAR-10 and PURCHASE-100 is relatively higher than MNIST. We attribute it to the fact that a more significant accuracy gap means a higher gap of overfitting, which is the commonly accepted reason for MIA [21]. Second, increasing the *Split_num* does not change the average training and testing accuracy. For instance, the first and second rows pertain to the CIFAR-10 results, with the only difference being the *Split_num*. Altering *Rates* does not change the average training and testing accuracy much. For example, the twelfth and thirteenth rows correspond to the PURCHASE-100 results, where the sole distinction lies in the *Rates*, with an accuracy discrepancy of 0.01 (*Avg train acc*) and 0.06 (*Avg test acc*). This observation can be attributed to the stability of the optimization algorithm used during the training of machine learning models. Third, changing *Split_num* does not impact the attack performance as it causes very slight alteration to the average overfitting gap (*Avg dif acc*) like the first and second rows for CIFAR-10. Meanwhile, the *Rates* parameter exerts an interpretable influence on the attack performance. The attack accuracy increases while applying the *Rates* of (0.5, 0.8, 0.8). Since the percentage of training data points increases, the attack model obtains high accuracy if it infers most data points as members while training the attack model and using it for attacking. This indicates that altering the *Rates* parameter does not influence the attack capability of MIA and can be considered an external perturbation.

Regarding the attack accuracy, we compute the average attack accuracy over $Split\_num$ target models to represent a specific MIA's attack performance for those target models. In Fig. 3, each point in the picture depicts the average attack accuracy of a configuration under one MIA. Due to 77 different MIAs, we obtain a list of average attack accuracy values: points with the same color and shape. Then, we calculate the mean (*Avg MIA acc*), variance (*MIA acc var*), median (*MIA acc med*), maximum (*MIA acc max*), and minimum (*MIA acc min*) values of each list of average attack accuracy values in Table VI. We provide Fig. 2 to show the relation between the average overfitting gap (*Avg dif acc*) and the double average attack accuracy (*Avg MIA acc*). We refer to the mean of the average attack accuracy values as the "double average attack accuracy" (*Avg MIA acc*) because it averages attack accuracy over both $Split\_num$ target models and 77 MIAs. The figure shows that a more significant average overfitting gap usually leads to a higher double average attack accuracy. However, the double average attack accuracy could be high even if the average overfitting gap is low. As far as we know, there is no clear mathematical formula to formulate this

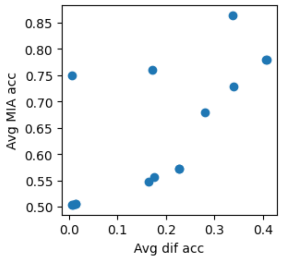
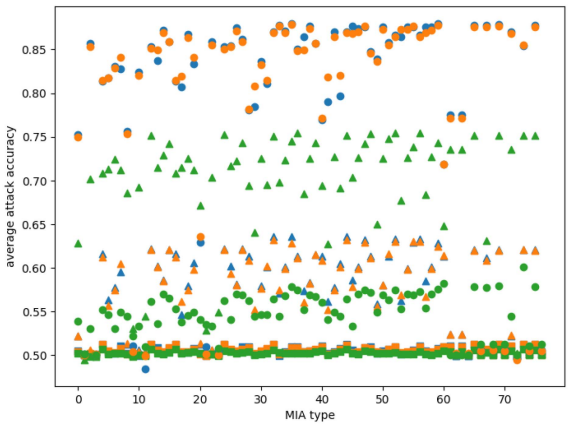


Fig. 2. The relationship between the average overfitting gap (x-axis) and the double average attack accuracy (y-axis).

(a) Legend



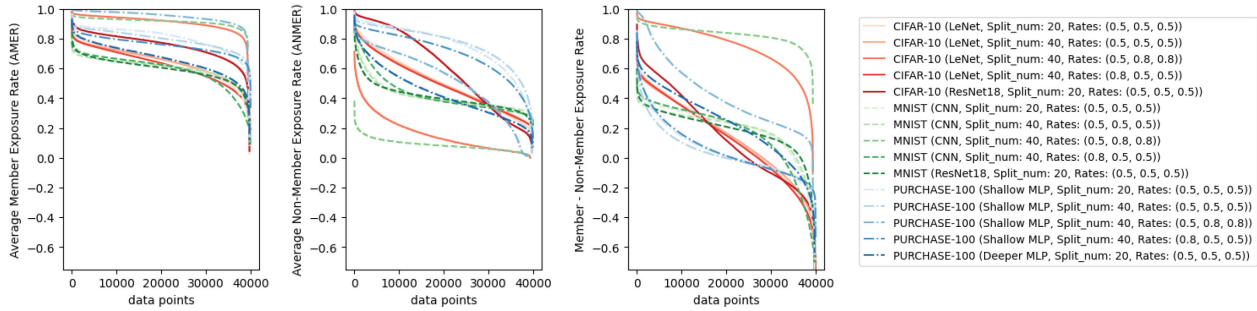
(b) Average Attack Accuracy

Fig. 3. The average attack accuracy of different MIAs. The x-axis represents different MIAs. The y-axis indicates the average attack accuracy over *Split_num* target models.

relationship. Therefore, we display this relationship in Fig. 2 as in the previous work [7].

In Fig. 3 and Table VI, we see that the average attack accuracy of MIAs ranges from 0.5 to 0.9 among the three datasets. However, within the same configuration (one row in Table VI), the average attack accuracy variance is relatively small (0.000004∼0.02). This phenomenon shows the configuration itself will significantly affect the attack performance. According to the previous analysis, the *Rates* does not essentially affect the attack performance. The *Split_num* has no effort on the attack performance. In Table VI, modifying model structures while keeping other settings (e.g., rows 11 and 15) impacts the average attack accuracy of MIAs (0.779 in row 11 and 0.548 in row 15). Similarly, the attack performance varies (0.679 in row 5 and 0.503 in row 10) while changing only the dataset (rows 5 and 10). Besides, a higher average overfitting gap could lead to a higher average attack accuracy in Fig. 2. According to those analyses, we conclude that overfitting gap, dataset, and model structure are the main factors affecting attack performance. This finding corresponds with the result of previous work [27]. We regard the maximal and minimal average attack accuracy values of 77 MIAs as two discrete variables and utilize the *t-test* to test the significance of the difference between those two variables. There are two hypotheses. The first one is the null hypothesis H_0: there is no significant difference between two variables; the other is the alternative hypothesis H_a : there is a significant difference between two variables. We use a standard significance level of 0.05 for comparison. After calculation, we get a p-value of 0.000055. Therefore, we reject the null hypothesis and conclude that there is a significant difference between the maximal and minimal average attack accuracy values of 77 MIAs. For example, row 11 for the result of PURCHASE-100 has a difference of 0.395. At the same time, the minimal average attack accuracy (*MIA acc min*) is close to 0.5in all experiments. This result indicates that not all 77 MIAs are efficient at attacking *Split_num* target models of each configuration. The MIA that achieves a higher attack performance than other MIAs under one configuration may not obtain competitive attack performance in other configurations compared to other MIAs.

From the previous analysis, suitable MIAs of high attack performance might vary within different datasets and model structure combinations. To provide more detail, we select three configurations to show the inference correctness of 77 MIAs on each 20 (*Split_num*) target models from three datasets in Fig. D.1 of the appendix, available online. Here, we only calculate the inference accuracy of the training data of the target model according to the goal of the MIA. In the figure, the x-axis depicts the types of MIAs, and the y-axis depicts target models. Each number on the y-axis represents three target models with the same mark from three datasets. There are two values (”0” and ”1”) in the figure. Setting ”1” indicates that the specific MIA attains an inference accuracy greater than or equal to 0.7 on each training set of three target models. Otherwise, we fill the cell with ”0”. We tally the ”1” in the column for each MIA, revealing that MIAs 4, 12, 16, 24, 32, 44, 65, 67, 69, 73, and 75 are particularly adept at inferring training data from three datasets. The explanation of those 11 MIAs is in Table D.1 of the appendix, available online. Additionally, most MIAs excel in inferring target models labeled 0, 1, 2, 3, 11, 12, and 15 by similarly counting the ”1” in each row. The analysis suggests that specific MIAs perform superior inference, while several target models exhibit weaknesses. The overfitting gaps of those target models range from 0.0125 to 0.4165. This suggests that we can accurately infer the training data of the target model with a low overfitting gap using multiple high-performance MIAs. While a more substantial overfitting gap typically correlates with higher attack performance, it is important to note that partial MIAs can still perform better in inferring models with a low overfitting gap. To conclude, the average training and test accuracies of *Split_num* target models do not rely on the *Split_num* value and are slightly related to the *Rates* value. Generally, a larger average overfitting gap leads to a higher double average attack accuracy. Dataset and model structure are also the main factors of MIA's attack performance. By analyzing the inference correctness of



(a) AMER (b) ANMER (c) Difference (d) Legend

Fig. 4. The sorted AMER values (2), ANMER values, and difference of those two metrics under varying configurations. The x-axis indicates the symbol of data points, and the y-axis means the metric value. As there is no direct relation between lines, the same x symbol usually means different data points in varying lines because of the value sorting.

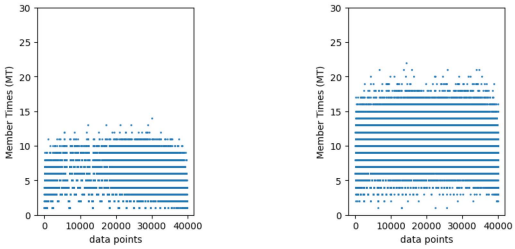
77 MIAs on the training data of each of the 20 target models from three datasets, we find that eleven MIAs achieve better attack performance, and seven target models in each dataset are weaker.

B. Data Point's Exposure Rate

Following Section III, the data point's AMER and ANMER represent its vulnerability under multiple MIAs and target models. As per the explanation in Section V-B2 for finding vulnerable data points, we use those metrics to select vulnerable data points. This section analyzes those two metrics and their difference in Fig. 4.

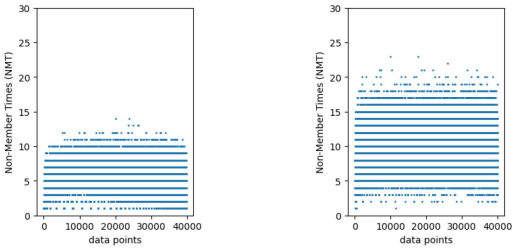
Fig. 4 shows the sorted AMER values, ANMER values, and the difference between those two metrics under varying configurations. First, the AMER and ANMER decrease from a value close to 1 to near 0. The ANMER has a faster and more significant drop than the AMER (from the shape and termination of curves), which leads to a large proportion of the AMER value being higher than the ANMER value. This observation is understandable because MIA is biased toward inferring the training data rather than the test data. Therefore, training data points' vulnerability is more substantial than test data points under multiple MIAs. It corresponds with the result that the AMER value is higher than the ANMER value. From another perspective, this phenomenon reflects the appropriateness of our newly defined metrics. Second, there is a proportion of data points with an AMER or ANMER value close to 1, meaning 77 MIAs almost correctly infer the membership of these data points in $Split\_num$ target models. In other words, these data points are susceptible to inference by 77 MIAs while being in $Split\_num$ target models. Therefore, a high AMER or ANMER value reflects the data point vulnerability under multiple MIAs and target models. Specifically, around 45 percent of data points have an AMER value larger than 0.6in all experiments in Fig. 4. Meanwhile, some data points have an ANMER value larger than 0.6. Those observations indicate partial but not all data points are vulnerable under multiple MIAs and target models. Third, the AMER and ANMER values change with the alteration of split rates. In particular, the AMER values are high when the split rates are 0.5, 0.8, and 0.8. The ANMER values are relatively high when the split rates are 0.5, 0.5, and 0.5. The possible reason for this phenomenon is that more training data points lead to the class imbalance of the dataset for training the attack model. The attack model tends to predict all data points to members. Therefore, the AMER values are higher. Fourth, increasing the value of *Split_num* from 20 to 40 does not introduce substantial changes to the curves of the AMER and ANMER values. It reflects that those two metrics are not sensitive to the *Split_num*. Increasing the *Split_num* means increasing the number of target models whose training data includes the specific data point while calculating its AMER value. The sum of MER values also increases because the AMER value is the average value of MER among target models whose training data includes this specific data point (Section III-B). Hence, we conclude that the *Split_num* has a limited impact on the values of AMER and ANMER. From Table VI, we observe that experiments related to PURCHASE-100 have higher average overfitting gaps (*Avg dif acc* from 0.164 to 0.408) than CIFAR-10 (0.1712 to 0.2805). Meanwhile, experiments of CIFAR-10 are more overfitted than MNIST (0.0049 to 0.0140). In Fig. 4(a) and (b), the curves for PURCHASE-100 (purple) and CIFAR-10 (red) are relatively higher than the curves of MNIST (green). It indicates that a higher average overfitting gap generally leads to high values of AMER and ANMER, which means high vulnerability under multiple MIAs and target models. As previously analyzed in Fig. 2 within Section VI-A, a larger average overfitting gap leads to an increased double average attack accuracy, implying improved membership inference correctness. This improved membership inference correctness is directly linked to higher values of AMER and ANMER, as stipulated by their respective definitions.

Due to the limited random splitting of the dataset for experiments, the MT and NMT values of different data points vary, which is inevitable because it is impossible to iterate all the subsets of the dataset with partial data points. Fig. 5 shows the MT of data points following the decrease of AMER values. Fig. 6 shows the NMT of data points following the decrease of ANMER values. Each figure's blue point represents the MT or NMT of one data point. In Fig. 5, there is a slightly less densely populated region at the bottom of each picture. Still, the overall distribution of MT over all data points is even. Similarly, the



(a) 20 Splits (b) 40 Splits

Fig. 5. The MT of data points following the decrease of AMER values with the *Split_num* of 20 and 40.



(a) 20 Splits (b) 40 Splits

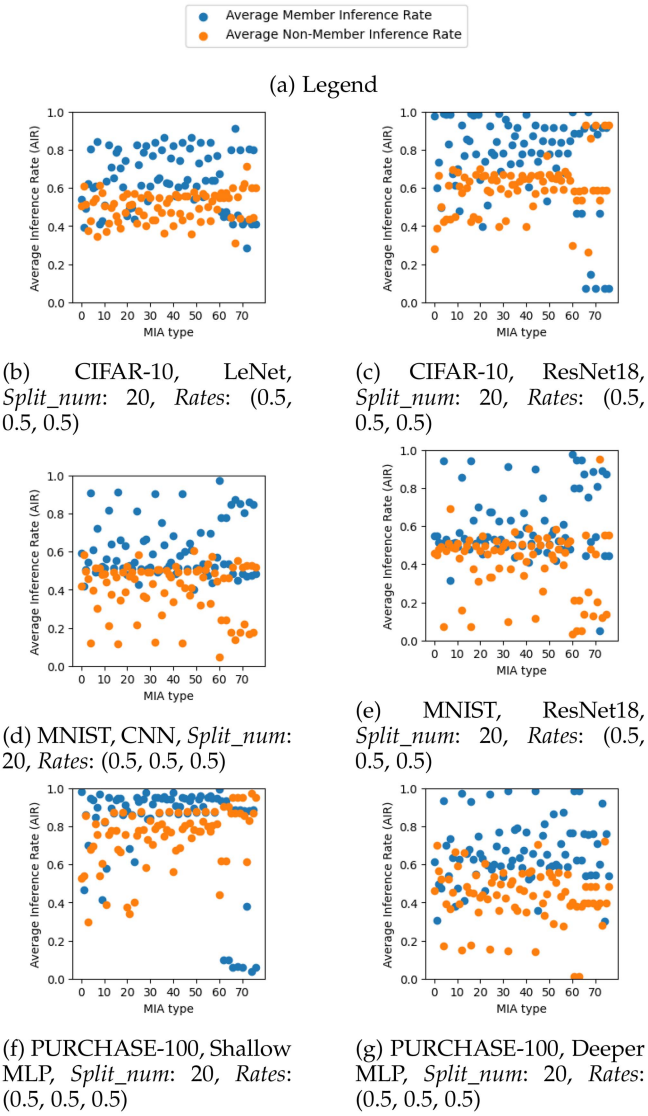
Fig. 6. The NMT of data points following the decrease of ANMER values with the *Split_num* of 20 and 40.

general distribution of NMT in Fig. 6 is balanced over all data points. According to (2), the MT and NMT values influence the AMER and ANMER metrics calculation. The distributions of MT and NMT are even in the mentioned figures. Nonetheless, each data point's AMER and ANMER values exhibit significant variation across the curves depicted in Fig. 4. This means that the values of MT and NMT among data points are not reasons for data points' fluctuating AMER and ANMER values. We have obtained similar figures and observations from other experiments with varying configurations, although we have not displayed all due to space constraints.

From the high values of AMER and ANMER, partial data points are vulnerable under multiple MIAs and target models. The vulnerability of the training data is more significant than the test data. Changing the *Split_num* does not change much to the curves of AMER and ANMER. The higher average overfitting gap generally causes larger values of AMER and ANMER. The values of MT and NMT do not account for data points' fluctuating AMER and ANMER values.

C. MIA's Inference Rate

Apart from the data point's exposure rate, we explore MIA's inference rate, which represents the inference correctness from the perspective of the MIA. From (3) and (4), the MIR and NMIR indicate the MIA's inference to a specific data point. The AMIR and ANMIR separately show the MIA's inference to all training and test data points. We draw the AMIR and ANMIR values under varying configurations in Fig. 7. The high values (even close to 1) of AMIR and ANMIR indicate that the MIA



(a) Legend

(b) CIFAR-10, LeNet, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

(c) CIFAR-10, ResNet18, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

(d) MNIST, CNN, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

(e) MNIST, ResNet18, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

(f) PURCHASE-100, Shallow MLP, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

(g) PURCHASE-100, Deeper MLP, *Split_num*: 20, *Rates*: (0.5, 0.5, 0.5)

Fig. 7. The AMIR (4) and ANMIR values of MIAs under different configurations. The x-axis means the type of MIAs. The y-axis represents the specific value of metrics for each MIA.

correctly infers most data points in target models. Besides, the figure shows that the AMIR value is generally more significant than the ANMIR value. The blue points in the figure are higher than the red points. For example, in the second sub-figure for CIFAR-10 and two sub-figures for PURCHASE-100, the AMIR value is close to 1. This observation is similar to the previous result: the AMER value is higher than the ANMER value. The reason is that training data points are more vulnerable than test data points, even under one MIA.

Table VII provides a small number of data points with high MIR or NMIR values while the corresponding MIAs have low AMIR or ANMIR values. In the table, the fourth column shows data points' top ten MIR (NMIR) values. Those ten data points' MT (NMT) values are given in the fifth column. The top ten MIR and NMIR values of data points are all 1, with high MT and NMT values from 8 to 12. Elevated MT and NMT values suggest that the MIR and NMIR metrics are not calculated sporadically.

TABLE VII
TOP 10 DATA POINTS WITH HIGH MIR OR NMIR VALUES WHILE THE AMIR OR ANMIR VALUES ARE RELATIVELY LOW

Configuration	MIA (type)	AMIR	Top 10 MIR of data points	MT for those 10 data points
CIFAR-10 (LeNet, Split_num: 20, Rates: (0.5, 0.5, 0.5))	57	0.64	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[11, 10, 10, 10, 10, 10, 10, 10, 10, 9]
MNIST (CNN, Split_num: 20, Rates: (0.5, 0.5, 0.5))	39	0.54	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[10, 9, 8, 8, 8, 8, 8, 8, 8, 8]
PURCHASE-100 (Shallow MLP, Split_num: 20, Rates: (0.5, 0.5, 0.5))	49	0.9	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[12, 12, 12, 12, 11, 11, 11, 11, 11, 11]

Configuration	MIA (type)	ANMIR	Top 10 NMIR of data points	NMT for those 10 data points
CIFAR-10 (LeNet, Split_num: 20, Rates: (0.5, 0.5, 0.5))	22	0.51	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[9, 9, 8, 8, 8, 8, 8, 8, 8, 8]
MNIST (CNN, Split_num: 20, Rates: (0.5, 0.5, 0.5))	62	0.46	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[10, 10, 9, 9, 9, 9, 9, 9, 9, 8]
PURCHASE-100 (Shallow MLP, Split_num: 20, Rates: (0.5, 0.5, 0.5))	37	0.78	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	[12, 12, 12, 12, 11, 11, 11, 11, 11, 11]

This observation suggests that MIAs with low AMIR or ANMIR values (close to 0.5) tend to infer specific data points precisely. The low AMIR or ANMIR values indicate that these MIAs perform poorly when evaluating all training or test data points together. High MIR and NMIR values indicate that MIA makes accurate inferences for specific data points rather than the entire training or test data. Consequently, we observe low AMIR or ANMIR values and high MIR and NMIR values coexist. This highlights some MIAs have an inference tendency to a piece of data points even though the overall inference performance is relatively low.

The MIAs obtain higher AMIR values than ANMIR values, which means that MIAs perform better on the training data. Even though the AMIR or ANMIR values are close to 0.5, the MIAs still infer some data points precisely.

D. Exploratory Experiments

In previous experiments, the number of selected data points is 40000. The shadow dataset is from the same data distribution as the target dataset. The hyperparameters (including the number of epochs, batch size, learning rate, weight decay, and optimizer) of *Split_num* target models are the same. Therefore, we add exploratory experiments to understand better the impact of data volume, the distribution of the shadow dataset, and the target models' hyperparameters.

Regarding data volume, we select 50000 data points from CIFAR-10, 60000 data points from MNIST, and 60000 data points from PURCHASE-100 in separate experiments. Besides, we changed the distribution of the shadow dataset to another two datasets to explore the effect of data distribution. Finally, we alter the hyperparameters while training *Split_num* target models. *Split_num* and *Rates* are the same (40 and (0.5, 0.5, 0.5)) for those three explorations.

Table B.1, available online, shows the dataset, model structure, total number of data points, target models' accuracies, and MIAs' attack performance. Compared with Table VI, more data points could slightly reduce the average overfitting gap (*Avg dif acc*) and decrease the double average attack accuracy (*Avg MIA acc*). For example, PURCHASE-100 obtains the average overfitting gap of 0.348 and the double average attack accuracy of 0.733 with 60000 data points. The average overfitting gap is 0.407, and the double average attack accuracy is 0.779 with 40000 data points (row 12 in Table VI). This is because more data points could improve the generalization of the trained model, reducing the average overfitting gap and the double average attack accuracy. While sampling the shadow dataset from another distribution, the double average attack accuracy is close to 0.5, even though the target models' average overfitting gap is relatively large. Unsurprisingly, the attack features from the shadow model are different from the target model due to the distribution difference between the shadow and target datasets. This leads to the low performance of MIAs. If we set different hyperparameters while training *Split_num* target models, the average overfitting gap and double average attack accuracy slightly increase for CIFAR-10 and MNIST and decrease for PURCHASE-100 by comparing the last three rows of Table B.1, available online, and three rows (2, 7, and 12) in Table VI. The alterations in average overfitting gaps are as follows: 0.0474 (increase) for CIFAR-10, 0.014 (increase) for MNIST, and 0.094 (decrease) for PURCHASE-100. Regarding the double average attack accuracy, the adjustment levels are 0.016 (increase) for CIFAR-10, 0.008 (increase) for MNIST, and 0.023 (decrease) for PURCHASE-100. Changing the hyperparameters leads to a change in the average overfitting gap, which affects the double average attack accuracy. A heightened average overfitting gap is associated with a more pronounced double average attack accuracy. The fluctuating pattern of the average overfitting gap exhibits inconsistency across the three datasets when training $Split\_num$ target models with varying hyperparameters. We attribute this phenomenon to the interaction between hyperparameters within training.

Fig. B.1, available online, displays the AMER, ANMER, and the difference between these two metrics in the context of three exploration experiments. While increasing the number of data points, the start and end points of AMER and ANMER curves are similar compared with 40000 data points. However, the middle of the curve is smoother with more data points. It indicates that more data points obtain a middle value of the AMER and ANMER range. The situations differ among the three datasets for experiments with shadow datasets from other distributions. For PURCHASE-100, the AMER significantly drops, and the ANMER significantly increases. For MNIST, the AMER slightly increases, and the ANMER slightly decreases. In the case of CIFAR-10, the AMER experiences a marginal increase, while the ANMER undergoes a slight decrease. We observe this trend because these experiments' double average attack accuracies hover around 0.5. The reduction (increase) of inference accuracy on the member data needs the increase (reduction) of inference accuracy on the non-member data to get a final random guessing accuracy. The direction and magnitude

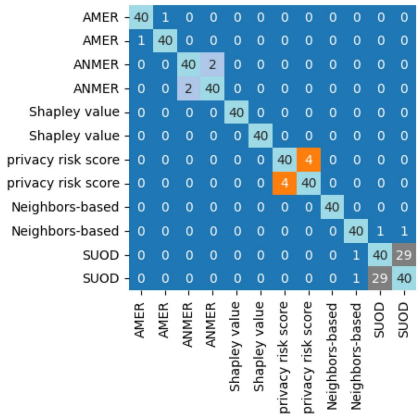
of change observed in either of these metrics are attributed to variations in the dataset. While changing the hyperparameters, the AMER and ANMER curves are relatively higher (lower) if the double average attack accuracy increases (decreases) by comparing lines with the only difference of hyperparameters in Figs. B.1 and 4. We attribute this observed behavior to the double average attack accuracy affecting vulnerability, which is also mentioned in Section VI-B.

In conclusion, increasing the number of data points has a modest effect of reducing the average overfitting gap, lowering the double average attack accuracy, and promoting a more uniform decline in the curves of AMER and ANMER. Sampling the shadow dataset from another distribution could break the attack performance and make the curves of AMER and ANMER change in different ways within different datasets. Training *Split_num* target models with different hyperparameters leads to the inconsistent change of the average overfitting gap among three datasets, which causes the change of the AMER and ANMER curves.

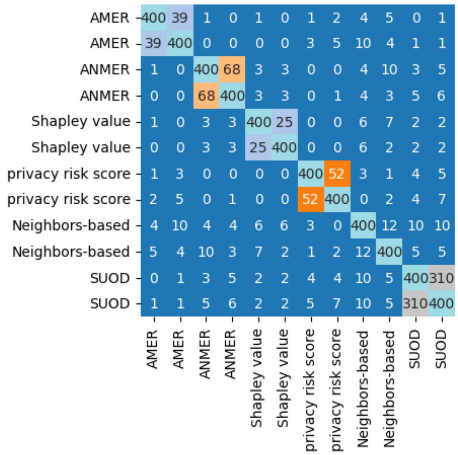
E. Vulnerable Data Points Comparison

We determine vulnerable data points under multiple MIAs and target models based on our new metrics (AMER and ANMER), neighbors-based method, privacy risk score, and Shapley value. Fig. 8 displays the overlapping data points between varying vulnerable sets of different methods and repetitions of experiments for CIFAR. We show the overlapping data points of the other two datasets in Figs. E.1 and E.2, available online. For comparison, we select 40 and 400 vulnerable data points to show the overlapping vulnerable data points. Besides, we use the outlier detection method, SUOD, to compare with previous methods. The previous work does not prove the data points found by the SUOD are genuinely vulnerable. The figure shows that SUOD identifies different data points compared to previous methods. While selecting 400 vulnerable data points, SUOD only identifies a few overlapping data points with those identified by ANMER. We use SUOD for a comparison purpose and do not assert the vulnerability of the data points it identifies, leaving this as future work.

We reveal the vulnerable data points identified by various methods and make comparisons in Fig. 8 (similar in Figs. E.1 and E.2, available online). From the analysis in Section VI-B and definitions in Section III, 77 MIAs correctly infer the vulnerable data points found with AMER and ANMER with a high probability. However, there is very limited overlap between the vulnerable data points identified by our metrics and those from previous methods. For 40 vulnerable data points, there is virtually no overlap. Among 400 vulnerable data points, the maximum overlap with our two new metrics is just 10 (CIFAR), which is less than 2.5%. This suggests that data points detected by previous methods cannot be accurately inferred by 77 MIAs while in $Split\_num$ target models. Previous methods are effective only in single MIA and single target model scenarios, rendering them ineffective in our case with multiple MIAs and target models.



(a) CIFAR-10 (40)



(b) CIFAR-10 (400)

Fig. 8. In the context of CIFAR-10, this figure illustrates the overlap of 40 and 400 vulnerable data points across different methods. The x-axis and y-axis represent the various approaches employed. When the same tag is present, it signifies a repeated experiment with no parameter changes except for retraining the target and attack models. The numerical values within each cell of the figure denote the count of overlapping data points.

From Fig. 8 (similar in Figs. E.1 and E.2, available online), we observe that detected vulnerable data points with our AMER (ANMER) are mostly various when we retrain target, shadow, and attack models (classifier-based MIAs), indicated by the low overlapping data points between two rows named ”AMER” (”ANMER”). Retraining a model in this work means that we keep the data, model structure, and hyperparameters of the previous run, initialize parameters with a different random seed and shuffle the data during training. Although we maintain the data, model structure, and hyperparameters, vulnerable data points found still vary a lot. Hence, we conclude that the randomness that comes from the retraining of target, shadow, and attack models has an impact on the vulnerability of data points under multiple MIAs and target models.

To further explore the impact of retraining target, shadow, and attack models on the detection of vulnerable data points under multiple MIAs and target models, we only retrain one of them

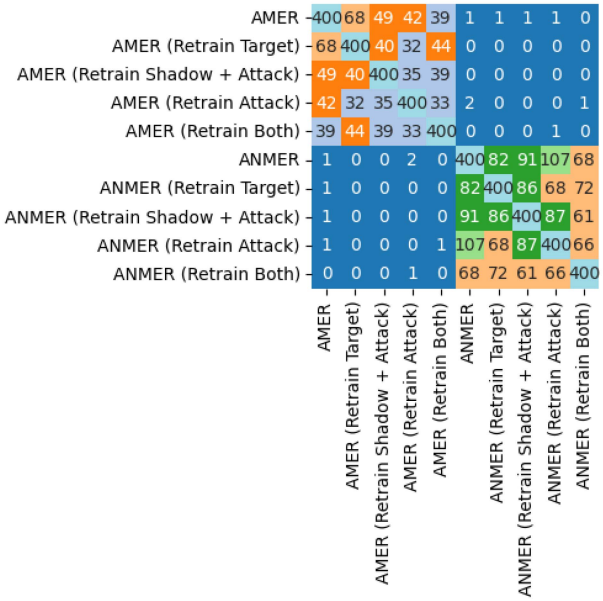


Fig. 9. The number of overlapping vulnerable data points detected by AMER and ANMER if we repeat training target, shadow, and attack models separately (CIFAR-10).

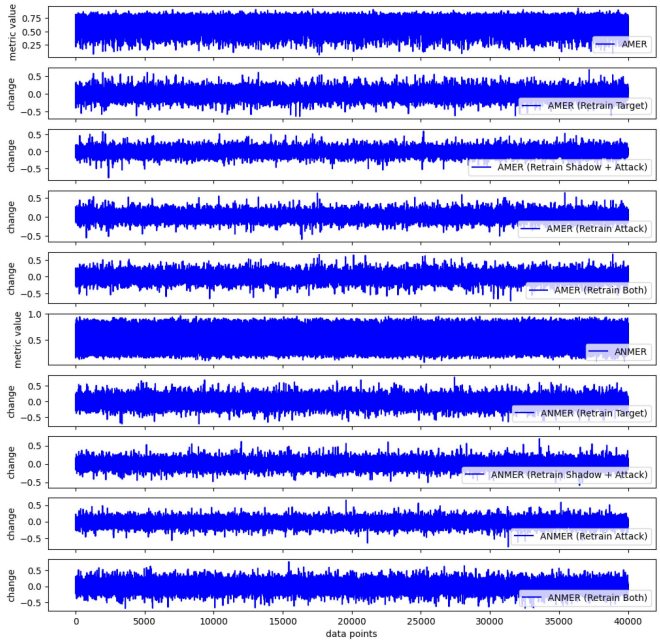


Fig. 10. The modification of AMER and ANMER during the repetition experiments (CIFAR-10).

and keep the other two unchanged. Fig. 9 (together with Figs. E.3 and E.4, available online) shows the number of overlapping vulnerable data points found with AMER and ANMER if we retrain models. Under PURCHASE-100, we find that retraining target models could still detect 236 identical vulnerable data points when measuring vulnerability with AMER. However, the number of maintained vulnerable data points is 68 for CIFAR-10 and 50 for MNIST. This is reasonable since the overlapping level of models trained with PURCHASE-100 is higher than the other two datasets. The high overlapping level makes it easier to distinguish members and non-members for MIAs. Hence, MIAs could detect more overlapping vulnerable data points under PURCHASE-100.

We compare the number of overlapping vulnerable data points measured with AMER when retraining target, shadow, or attack models. In PURCHASE-100 and CIFAR-10, retraining the target model will maintain more vulnerable data points (236 in PURCHASE-100 and 68 in CIFAR-10) than retraining shadow and attack models. Retraining shadow and attack models will keep more vulnerable data points than only retraining attack models, which indicates the impact of retraining attack models on determining the vulnerability of data points is larger. We speculate the reason for this phenomenon is that the attack model is closer to the final prediction of MIAs. Retraining the attack model will directly rebuild the strategy to distinguish members and non-members, which leads to less overlapping vulnerable data points. For MNIST, retraining the attack model has less impact on vulnerable data points than retraining target and shadow models, which is contrary to PURCHASE-100 and CIFAR-10. We attribute this difference to the attack features of data points from MNIST that are not distinguishable enough for MIAs due to a lower overfitting level than PURCHASE-100 and CIFAR-10. Hence, attack models tend to maximize the attack performance in a similar way in MNIST, which leads to more overlapping vulnerable data points if only retraining attack models of MIAs. From the above analysis, we conclude that retraining the target model has less impact than retraining shadow and attack models on vulnerable data points measured with AMER if the dataset tends to be trained in an overfitting way. If the model trained on the dataset is hard to overfit, retraining target models has more impact than retraining shadow and attack models on the number of overfitting vulnerable data points.

To verify our previous findings, we analyze the modification of AMER and ANMER values of data points while retraining target, shadow, and attack models separately. We draw the modification of AMER and ANMER in Fig. 10 (together with Fig. E.5, available online) with the x-axis representing the index of the data point and the y-axis indicating the metric value (AMER or ANMER) or change. Among ten rows in the figure, rows named "AMER" and "ANMER" are actual metric values, and the other eight rows are modified metric values. From Fig. E.5, available online, we can observe that retraining the target model brings a larger modification range (-0.5 to 0.5) than retraining shadow (-0.25 to 0.25) and attack (-0.1 to 0.1) models. More modification leads to less overlapping vulnerable data points in Fig. E.3, available online. This is reasonable since larger modification in metric value makes the vulnerable data points with the highest metric value inconsistent. For CIFAR-10, the modification of AMER (ANMER) is within a high range (-0.5 to 0.5), which leads to a low number of overlapping vulnerable data points (from 42 to 68 for AMER). Hence, the modification of metric value reflects on the number of overlapping vulnerable

data points. A high metric modification leads to less overlapping vulnerable data points.

The vulnerable data points determined by the AMER and ANMER values are different except for a few data points. This observation comes from the small number of overlapping vulnerable data points determined by the AMER and ANMER values. This occurs because the vulnerability of a data point differs when it is part of the training data compared to when it is included in the test data. We also show the vulnerability difference between the training and test data in Section VI-B. Consequently, it is challenging to identify similar vulnerable data points based on their AMER and ANMER values. Besides, the impact of retraining target, shadow, or attack models on vulnerable data points measured with ANMER is different from that measured with AMER. For example, the number of overlapping vulnerable data points measured with AMER decreases from 68 (retrain the target model) to 42 (retrain the attack model) in Fig. 9. When it comes to ANMER, the number of overlapping vulnerable data points increases. The reason for this difference is also related to the vulnerability difference between the training and test data.

The previous methods are ineffective in identifying vulnerable data points when subjected to multiple MIAs and target models, as evidenced by the limited overlap between the data points identified by previous methods and our metrics. The impact of retraining target, shadow, and attack models on the vulnerability of data points varies on the metric type and dataset. In terms of AMER, a high overfitting level of the target model on the dataset eliminates the impact of retraining the target model, which makes retraining the target model less influential to the vulnerability of data points compared with retraining shadow and attack models. Larger modification of metric values leads to less number of overlapping vulnerable data points, which reflects the impact of retraining models on detecting vulnerable data points. The vulnerable data points found by AMER and ANMER vary due to the different vulnerabilities of the training and test data.

VII. Related Work

MIA has attracted significant attention after the seminal work of Shokri et al. [7]. We divide current research works into attacking methods, defense strategies, and a deeper understanding of MIA.

Attacking methods: Shokri et al. put forward the application of the shadow model to mimic the behavior of the target model and obtain the dataset for training the attack model [7]. Salem et al. relaxed some assumptions in the work of Shokri et al. They proposed data-transferring and threshold-based attacks using the highest posterior, standard deviation, and entropy [15]. Yeom et al. formulated three adversaries of MIAs with additional information, including loss, empirical error, leave-one-out validation error, average training loss, and even the training dataset [21]. Sablayrolles et al. used Bayes' formula, optimal steps, and approximations to implement optimal MIA only with loss [28]. Long et al. carefully selected a few vulnerable data points and attacked them with direct and indirect inference [8]. Li et al. [14], and Choquette-choo et al. [13] investigated attacking target model under label-only condition. They proposed three methods: applying adversarial examples, data augmentation, and relabeling shadow data with the target model. Carlini et al. presented the Likelihood Ratio Attack (LiRA), which formulates the MIA as hypothesis testing and considers the data point in or out of the training data separately [11]. The concept of separation manipulation serves as an inspiration for our work. Our empirical result shows that the data point's vulnerability is different in the training data compared with the test data. Ye et al. put forward model-dependent and sample-dependent MIA via distillation, which means the determination threshold is related to the target model and data point [29].

Defense strategies: Strategies for reducing the overfitting are proposed for eliminating the MIA, including dropout [30], L2-norm standard regularization, and model stacking. Some strategies manipulate the target model's output, including classes of output limitation, prediction vector modification, and prediction entropy increase [7], [15]. Nasr et al. combined the training process of the target model with a misleading attack classifier, decreasing the performance of classifier-based MIAs [31]. Jia et al. added slight perturbation to the prediction vector, leading to the misclassification of classifier-based MIAs [32]. Shejwalkar et al. leveraged knowledge distillation to train ML models with membership privacy [33]. Li et al. utilized the mix-up data augmentation and Maximum Mean Discrepancy regularization to mitigate the gap between training and validation accuracy, ultimately leading to a decrease in attack performance [34]. Tang et al. proposed a novel ensemble architecture and a self-distillation framework to defeat MIAs [35]. Jarin et al. decreased MIAs' performance by excluding the sub-model prediction whose training data include the target data point [36]. Finally, DP-SGD is frequently mentioned to defend against MIAs with high utility costs [37].

Deeper understanding: The factors contributing to the success of the MIA attracted frequent discussion. Overfitting, the choice of target model and dataset, the selection of part data points, and the complexity of the training dataset are recognized as influence factors of MIAs [7], [8], [15], [21], [26], [27], [38]. Besides, some works are devoted to analyzing the privacy risk of the target model with the help of the MIA [25], [26]. Furthermore, the data points with high privacy risk, susceptibility, or vulnerability are detected with different methods [8], [9], [10]. The focus of this paper, vulnerable data points under multiple MIAs and target models, differs from previous settings.

VIII. Limitations

There are three main limitations to our work. First, we only consider classification tasks with the image and number features. The MIAs for other tasks and data formats are not included, e.g., generative models [39], [40], [41], graph data [42], [43], [44], [45], and federated leaning [46], [47].

Second, the type of MIAs implemented in current research is limited. MIAs in this work generally use the shadow dataset from the same distribution to determine the threshold or train the attack model. Some strategies exist to overcome the condition that the shadow dataset comes from the same distribution [14], [15].

Other recently proposed MIAs are not implemented in this work. For example, the label-only MIAs proposed by Li et al. [14] and Choquette-choo et al. [13] with the help of adversarial examples or data augmentation, the LiRA put forward by Carlini et al. [11], and the strategy of calculating the threshold for each category proposed by Song et al. [9].

Third, the data of *Split_num* target models are from the same dataset. It would be interesting to investigate the cases where the *Split_num* target models are trained from multiple datasets with different but similar distributions. This would be more close to real-world scenarios for multiple ML target models. While our work has those limitations (which we leave for future work), we still manage to provide results unrelated to specific MIAs and datasets.

IX. Conclusion

This paper explores a single data point's vulnerability and tries to find vulnerable data points under multiple MIAs and target models. To formally analyze the data point's vulnerability, we define metrics about the data point's exposure rate and MIA's inference rate. All experiments are completed with the help of our newly developed platform, VMIAP, which is scalable and flexible for attacking target models with varying MIAs.

Our main takewaway messages are

1) The overfitting gap, the dataset, and the model structure are the main factors for the MIAs' attack accuracies.
2) Our new metrics, AMER and ANMER, reflect the actual situation of the data point's vulnerability and capture vulnerable data points under multiple MIAs and target models. Changing the factors that influence MIAs' attack accuracies will also impact the vulnerability of the data point.
3) MIA could still infer some data points precisely despite its relatively low overall inference performance of 0.5.
4) From definitions and a few overlapping data points between previous methods and our metrics, we empirically verify that methods previously used to find vulnerable data points are inappropriate for the case under multiple MIAs and target models.
5) The impact of retraining target, shadow, and attack models on detecting vulnerable data points is different and related to the metric type (AMER or ANMER) and the overfitting level of the model trained on the dataset. From the perspective of the adversary (AMER), retraining target models has less impact on the vulnerability of data points than retraining shadow and attack models if the target model trained on the dataset is more overfitted.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, pp. 115:1–115:35, 2021.

[2] H. Chang and R. Shokri, "On the privacy risks of algorithmic fairness," in *Proc. 6th IEEE Eur. Symp. Secur. Privacy*, 2021, pp. 292–303.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[4] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[5] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. 25th USENIX Secur. Symp.*, Austin, TX, USA, 2016, pp. 601–618.

[6] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. 39th IEEE Symp. Secur. Privacy*, 2018, pp. 36–52.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. 38th IEEE Symp. Secur. Privacy*, 2017, pp. 3–18.

[8] Y. Long et al., "A pragmatic approach to membership inferences on machine learning models," in *Proc. 5th IEEE Eur. Symp. Secur. Privacy*, 2020, pp. 521–534.

[9] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 2615–2632.

[10] V. Duddu, S. Szyller, and N. Asokan, "SHAPr: An efficient and versatile membership privacy risk metric for machine learning," 2021, *arXiv:2112.02230*.

[11] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership inference attacks from first principles," in *Proc. 43rd IEEE Symp. Secur. Privacy*, 2022, pp. 1897–1914.

[12] Y. Zhao, X. Ding, J. Yang, and H. Bai, "SUOD: Toward scalable unsupervised outlier detection," 2020, *arXiv: 2002.03222*.

[13] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 1964–1974.

[14] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proc. 2021 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 880–895.

[15] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. 26th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[16] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, no. 1, pp. 169–198, 1999.

[17] W. S. Noble, "What is a support vector machine?," *Nature Biotechnol.*, vol. 24, pp. 1565–1567, 2006.

[18] K. S, V. S, and R. R, "A comparative analysis on linear regression and support vector regression," in *Proc. Online Int. Conf. Green Eng. Technol.*, 2016, pp. 1–5.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[20] A. Zimek, R. J. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: Challenges and research questions a position paper," *SIGKDD Explorations*, vol. 15, no. 1, pp. 11–22, 2014.

[21] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. 31st IEEE Comput. Secur. Found. Symp.*, 2018, pp. 268–282.

[22] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, "Revisiting membership inference under realistic assumptions," in *Proc. Privacy Enhancing Technol.*, vol. 2021, pp. 348–368, 2021.

[23] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interactive Multimedia Artif. Intell.*, vol. 4, pp. 26–30, 2016.

[24] E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[25] S. K. Murakonda and R. Shokri, "ML privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning," 2020, *arXiv:2007.09339*.

[26] Y. Liu et al., "ML-Doctor: Holistic risk assessment of inference attacks against machine learning models," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 4525–4542.

[27] S. Truex, L. Liu, M. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Trans. Serv. Comput.*, vol. 14, no. 6, pp. 2073–2089, Nov./Dec. 2021.

[28] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5558–5567.

[29] J. Ye, A. Maddi, S. K. Murakonda, and R. Shokri, "Enhanced membership inference attacks against machine learning models," 2021, *arXiv:2111.09679*.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[31] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proc. 25th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 634–646.
[32] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "MemGuard: Defending against black-box membership inference attacks via adversarial examples," in *Proc. 26th ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 259–274.
[33] V. Shejwalkar and A. Houmansadr, "Membership privacy for machine learning models through knowledge transfer," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 9549–9557.
[34] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in classification models," in *Proc. 11th ACM Conf. Data Appl. Secur. Privacy*, 2021, pp. 5–16.
[35] X. Tang et al., "A novel self-distillation architecture to defeat membership inference attacks," in *Proc. NeurIPS Workshop Privacy Mach. Learn.*, 2021.
[36] I. Jarin and B. Eshete, "MIAShield: Defending membership inference attacks via preemptive exclusion of members," 2022, *arXiv:2203.00915*.
[37] M. Abadi et al., "Deep learning with differential privacy," in *Proc. 23rd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.
[38] B. Kulynych, M. Yaghini, G. Cherubin, M. Veale, and C. Troncoso, "Disparate vulnerability to membership inference attacks," in *Proc. Privacy Enhancing Technol.*, 2022, pp. 460–480.
[39] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "GAN-leaks: A taxonomy of membership inference attacks against generative models," in *Proc. 2020 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 343–362.
[40] H. Hu and J. Pang, "Membership inference attacks against GANs by leveraging over-representation regions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 2387–2389.
[41] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "LOGAN: Membership inference attacks against generative models," in *Proc. Privacy Enhancing Technol.*, vol. 2019, pp. 133–152, 2019.
[42] B. Wu, X. Yang, S. Pan, and X. Yuan, "Adapting membership inference attacks to GNN for graph classification: Approaches and implications," in *Proc. 21st IEEE Int. Conf. Data Mining*, 2021, pp. 1421–1426.
[43] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," 2021, *arXiv:2102.05429*.
[44] I. E. Olatunji, W. Nejdl, and M. Khosla, "Membership inference attack on graph neural networks," in *Proc. 3rd IEEE Int. Conf. Trust Privacy Secur. Intell. Syst. Appl.*, 2021, pp. 11–20.
[45] Z. Zhang, M. Chen, M. Backes, Y. Shen, and Y. Zhang, "Inference attacks against graph neural networks," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 4543–4560.
[46] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. 40th IEEE Symp. Secur. Privacy*, 2019, pp. 691–706.
[47] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. 40th IEEE Symp. Secur. Privacy*, 2019, pp. 739–753.

Mauro Conti (Fellow, IEEE) received the PhD degree from the Sapienza University of Rome, Italy, in 2009. He is a full professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. After obtaining his PhD, he was a post-doc researcher with Vrije Universiteit Amsterdam, The Netherlands. In 2011, he joined as an assistant professor with the University of Padua, where he became associate professor in 2015 and full professor in 2018. He has been a visiting researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded a Marie Curie Fellowship (2012) by the European Commission and a Fellowship by the German DAAD (2013). Companies, including Cisco, Intel, and Huawei also fund his research. His main research interest include the area of security and privacy. In this area, he published more than 450 papers in the topmost international peer-reviewed journals and conferences. He is editor-in-chief of *IEEE Transactions on Information Forensics and Security*, area editor-in-chief of the *IEEE Communications Surveys & Tutorials*, and has been associate editor for several journals, including the *IEEE Communications Surveys & Tutorials*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Forensics and Security*, and *IEEE Transactions on Network and Service Management*. He was program chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, CANS 2021, and general chair for SecureComm 2012, SACMAT 2013, NSS 2021, and ACNS 2022. He is a senior member of the ACM, and a Young Academy of Europe fellow.

Jiaxin Li received the bachelor's and master's degrees from the Harbin Institute of Technology, China. He is currently working toward the PhD degree with the University of Padua, Italy. He focuses on machine learning privacy research, including membership inference attacks and differential privacy.

Stjepan Picek (Senior Member, IEEE) received the PhD degree in cryptology and evolutionary computation techniques, in 2015. He is an associate professor with Radboud University, The Netherlands. His research interests include security/cryptography, machine learning, and evolutionary computation. Prior to the associate professor position, he was an assistant professor with TU Delft, and a postdoctoral researcher with MIT, USA and KU Leuven, Belgium. Up to now, he has given more than 40 invited talks and published more than 150 refereed papers. He is a program committee member and reviewer for a number of conferences and journals, and a member of several professional societies.