VPP: Privacy Preserving Machine Learning via Undervolting

Md Shohidul Islam, Behnam Omidi, Ihsen Alouani, Khaled Khasawneh

V_{PP}: Privacy Preserving Machine Learning via Undervolting

Md Shohidul Islam
CSE Dept., DUET, Bangladesh
George Mason University
Fairfax, VA, USA
mislam20@gmu.edu

Behnam Omidi
ECE Department
George Mason University
Fairfax, VA, USA
bomidi@gmu.edu

Ihsen Alouani
CSIT, Queen's University Belfast, UK
IEMN CNRS-8520
INSA Hauts-de-France
Université Polytechnique Hauts-de-France
i.alouani@qub.ac.uk

Khaled N. Khasawneh
ECE Department
George Mason University
Fairfax, VA, USA
kkhasawn@gmu.edu

***Abstract*—Machine Learning (ML) systems are susceptible to membership inference attacks (MIAs), which leak private information from the training data. Specifically, MIAs are able to infer whether a target sample has been used in the training data of a given model. Such privacy breaching concern motivated several defenses against MIAs. However, most of the state-of-the-art defenses such as Differential Privacy (DP) come at the cost of lower utility (i.e., classification accuracy).**

In this work, we propose Privacy Preserving Volt (V_{PP}), a new lightweight inference-time approach that leverages undervolting for privacy-preserving ML. Unlike related work, V_{PP} maintains protected models' utility without requiring re-training. The key insight of our method is to blur the MIA differential analysis outcome by comprehensively garbling the model features using random noise. Unlike DP, which injects noise within the gradient at training time, V_{PP} injects computational randomness in a set of layers' during inference through carefully designed undervolting. Specifically, we propose a bi-objective optimization approach to identify the noise characteristics that yield privacy-preserving properties while maintaining the protected model's utility. Extensive experimental results demonstrate that V_{PP} yields a significantly more interesting utility/privacy tradeoff compared to prior defenses. For example, with comparable privacy protection on CIFAR-10 benchmark, V_{PP} improves the utility by 32.93% over DP-SGD. Besides, while related noise-based defenses are defeated by label-only attacks, V_{PP} shows high resilience to such adaptive MIA. Moreover, V_{PP} comes with a by-product inference power gain of up to 61%. Finally, for a comprehensive analysis, we propose a new adaptive attacks that operate on the expectation over the stochastic model behavior. We believe that V_{PP} represents a significant step towards practical privacy preserving techniques and considerably improves the state-of-the-art.

I. Introduction

The promising performance of ML models has made them commonplace in myriads of applications: recommendation system [1], medical diagnosis [2], system security [3], [4], malware detection [5]–[7], image/audio/video based application [8]–[10], resource provisioning [11] etc. Models are being trained with increasingly sensitive datasets such as clinical/biomedical records, personal photos, genome data, financial, social, and location traces, etc. Due to its complexity and high computational requirements, training ML models is often performed with crowd-sourced data on cloud providers (*e.g.*, Amazon AWS, Microsoft Azure, Google API), which offer ML-as-a-Service, thereby allowing novice end users as well as professionals to train models that often contain *personally identifiable information* or potentially sensitive personal data [12]. For this reason, ensuring data privacy in these systems, especially protecting training data from any leakage is crucial for building trustworthy ML systems.

One of the first privacy-related ML vulnerabilities is membership inference attack (MIA). MIAs leak sensitive information about the private training data by only accessing the model at inference time. More precisely, MIAs are able to infer whether a target sample has been used in training a targeted model. Due to overfitting to the training data, ML models are generally biased and behave differently on training data (members) versus test data (non-members). This bias can be observed through a statistically higher confidence of models in members classification compared to non-members. Attackers exploit this bias and mount MIAs [13].

In the literature, MIAs are demonstrated using several features, *e.g.*, logit, confidence, loss, entropy, gradients, hidden layer output, etc [12], [14]. While MIAs may seem like a weak and harmless attack model, revealing the membership in settings such as shown in [15] can represent a critical threat. In addition to the privacy threat, MIAs can also be used for privacy auditing purposes as an upper bound method. The importance of MIAs have motivated researchers to investigate various defenses [12], [16]–[19]. In general, defenses against MIAs try to inhibit the model behavioral bias between members and non-members. For instance, Differentially-Private Stochastic Gradient Descent (DP-SGD) [16] is a state-of-the-art defense that leverages differential privacy in training models. Specifically, it adds bounded random noise to gradients in the back-propagation during training to obfuscate the impact of individual samples on the overall loss function. However, this method comes at a cost in terms of utility; it considerably reduces models' baseline accuracy. Private Aggregation of Teacher Ensembles (PATE) [20], [21] is another defense based on ensemble learning that uses differential privacy; while ensuring theoretical privacy guarantee, PATE also comes at a significant cost in terms of models' utility.

In contrast with provable privacy approaches, recent works are based on *empirical membership privacy*, where the eval-

uation of model privacy is empirical using practical MIAs, to preserve model utility. For example, AdvReg [17] enhances privacy by improving model's regularization. However, like other regularization techniques, such as label smoothing [22] or dropping [23], AdvReg achieves a low privacy-utility trade-off (*i.e.*, acceptable privacy guarantee with a substantial loss in utility). MemGuard [18] another defense, which proposed to inject bounded noise to the model's prediction vector to confuse the attack. While MemGuard can maintain high utility, it shows considerably high privacy risk compared to other defenses. More specifically, since the noise is injected only in the output layer, MemGuard has been shown vulnerable to Label-Only MIA [24]. More recently, distillation for membership privacy (DMP) [12] and Self Ensemble Architecture (SELENA) [19] have been proposed, which improve both the utility and privacy significantly. However, both defenses require changes to the training procedure and *add computational overhead to both training and inference* of the defended model.

Recent works show that undervolting can be leveraged to introduce computational noise to the model inference that makes the model more robust against adversarial attacks, i.e., carefully crafted additive noise that undermines model integrity [4], [25]. More interestingly, the particularity of undervolting is that beyond offering effective defense against adversarial attacks, it comes with by-product power savings, without requiring changes to the hardware/software nor to the model, *i.e.*, no retraining is needed.

Inspired by recent works on undervolting as a defense, we propose Privacy Preserving Volt (V_{PP}), a randomness-based approach for privacy-preserving ML that maintains protected models' utility via undervolting. Unlike related noise-based techniques that inject random noise in the gradient (*e.g.*, DP-SGD [16]) or exclusively in the output (*e.g.*, MemGuard [18]), V_{PP} injects noise within the model computations at inference time. The injected noise is a random variable whose parameters are identified through a design-time, bi-objective space exploration *to maintain both privacy and utility*. Interestingly, since noise is injected during inference, V_{PP} does not require retraining the model, thus, can be deployed on off-the-shelf pre-trained models. The intuition behind V_{PP} is that the behavioral bias can be obfuscated **at inference time** by introducing stochasticity within the model's decision boundary to inhibit the information leakage from the confidence distribution of members and non-members. Therefore, instead of limiting the stochasticity to the output, V_{PP} exploits the noise-tolerance characteristic of ML models to explore the noise space in a way that garbles features used for MIA and obfuscates the model's behavioral bias. The challenge of such approach is to use sufficient noise to stop the information leakage, without degrading the model's accuracy. Therefore, we propose to explore the space for optimal noise properties; we formulate the problem as a multi-objective optimization with two objectives: minimizing both privacy leakage and accuracy drop due to the injected noise. To solve this problem, we use a Multi-Objective Genetic Algorithm (MOGA), which explores for the Pareto front, and chooses the solution that maximizes the model privacy. Our results show that V_{PP} achieves the best privacy/utility tradeoffs compared to prior defenses. In particular, V_{PP} incurs only a minor drop (no more than 2.37%) in classification accuracy, while achieving similar privacy protection to the strong DP-SGD defense. In addition, we propose an adaptive attack against V_{PP}, which operate on the expectation over the stochastic model behavior.

The key contributions of this work are as follows:

- We propose V_{PP}, a new randomness-based defense against MIAs that **protects models privacy with negligible utility loss** and without requiring re-training. Specifically, we inject undervolting-induced noise within the computation of a number of layers at inference time.
- We recorded an average **inference power gain** that ranges from 29% to 61.3% due to V_{PP} for an FPGA set-up.
- To find the optimal noise magnitude and stochasticity depth (*i.e.*, the number of stochastic layers), we propose a multi-objective evolutionary algorithm to explore the space for both privacy and accuracy. The output of the proposed algorithm is a set of non-dominated solutions (noise standard deviation and stochasticity depth combinations).
- We demonstrate a hardware-based implementation of V_{PP} using undervolting that results in a by-product energy gain in addition to the privacy-preserving aspect. We demonstrated this practical implementation on a Xilinx FPGA and verify the noise models with both simulation and real hardware.
- Our extensive evaluation using a range of state-of-the-art MIAs shows that V_{PP} yields significantly higher utility-privacy tradeoff than prior work. For example, with similar privacy to DP-SGD (the strongest privacy defense), our defense has a negligible impact on utility; V_{PP} outperforms DP-SGD by 32.93%, AdvReg by 21.53%, DMP by 20.13%, MemGuard by 8.68%, and SELENA by 3.96% in terms of utility for AlexNet trained on CIFAR-10 dataset.
- We systematically evaluated V_{PP} against a new adaptive attack that attempt to estimate the model expectation over a set of repeated queries. The adaptive attack was significantly more efficient than the state-of-the-art, yet V_{PP} showed $\sim 10\times$ privacy preserving under low false-positive-rates of the attacks.

II. Controlled undervolting analysis on FPGA

In this section, our goal is to characterize the undervolting effect on an FPGA and extract a statistical software framework that simulates the model's behavior under different sub-nominal supply voltage levels.

A. Hardware Setup

The hardware setup consists of an FPGA board, external voltage controller, and a host machine as shown in Figure 1. Specifically, we use a Xilinx Zynq Ultrascale+ ZCU104 FPGA board. The board consists of the XCZU7EV-2FFVC1156

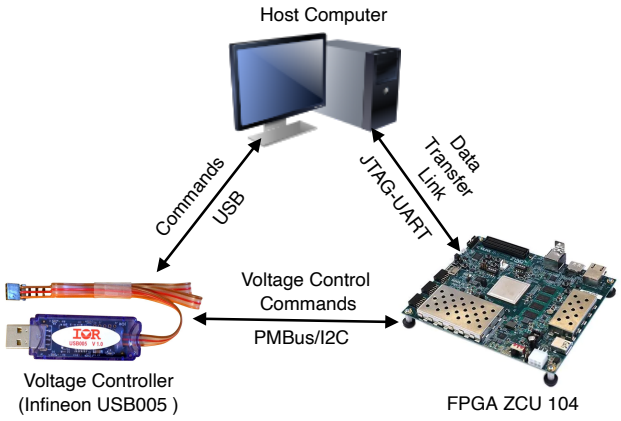


Fig. 1: Hardware Setup Platform.

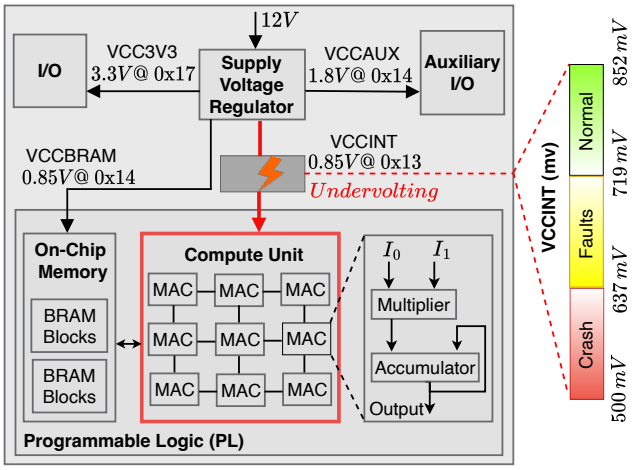


Fig. 2: Undervolting the 'compute unit' of FPGA ZCU 104.

MPSoc. The device's Processing System (PS) includes a quad-core Arm Cortex-A53 applications processor and a dual-core Cortex-R5 real-time processor. We leverage an external voltage controller, the Infineon USB005, to perform undervolting on the FPGA device, which is connected to the board via an I2C wire. We *read* and *write* the different voltage rail supplies to the board using PowerIRCenter GUI.

B. Undervolting Characterization

For any device, the voltage spectrum contains three regions: Safe, Critical and Crash. Normally, devices operate in the Safe region to avoid any faults/errors within the device. In the Crash region, the device would not operate either due to system safeguards or simply due to the intolerable errors. In this paper, we are interested in the Critical region, where the device would experience computational faults/errors but continue to operate.

We used the hardware setup as described in Section II-A to perform undervolting characterization on an FPGA board. We used CHaiDNN [26], an open-source DNN accelerator from Xilinx, to implement VGGNet [27], AlexNet [28], and ResNet18 [29] models and controlled the supply voltage through the external voltage regulator. We focused on VCCINT voltage rail, which can be accessed via PMBus address `0x13`, since it supplies voltage to the internal components in PS, as well as the DSP and LUT units in PL, as shown in Figure 2. The nominal supply voltage of VCCINT voltage rail is 0.852V. Then, we gradually lower the voltage on the VCCINT voltage rail with a step size of 4 mV (voltage change resolution of the voltage controller) to find the three voltage regions within the FPGA voltage spectrum. Our results shows that the regions are: (1) Safe ($0.7V < voltage < 0.85V$): the device functions normally, (2) Critical ($0.637V < voltage < 0.702V$): the device will occasionally have faults, and (3) Crash ($voltage < 0.637V$): the devices cannot function. Moreover, the power consumption of these regions are as follows: Safe ($3.5w < p < 7.750W$), Critical ($3W < p < 5.5W$), and Crash ($2.750W < p < 3.250W$).

C. Software Simulation

CHaiDNN does not support the access to the internal parameters of its models, which interferes with the ability of performing MIAs. Therefore, we conducted an experiment to test if injecting additive layer-wise Gaussian noise can provide the same computational noise distribution of undervolting an ML accelerator. Specifically, in this experiment, our goal is to find the undervolting level that matches the noise distribution of a given noise variance (σ).

We used three models such as VGGNet on ImageNet [30], AlexNet on CIFAR10 [31], and ResNet18 on CIFAR100 [31] dataset, which are denoted as ImageNet+VGGNet, CIFAR10+AlexNet, and CIFAR100+ResNet18 respectively. We run each model using both software implementation (Soft run) and hardware implementation (FPGA run). Following the statistical modeling of undervolting impact in [32], [33], we modeled the undervolting-induced computational error with a bit flip in the output of computational elements. The bit-flip location is a random variable that results in a zero-mean Gaussian distribution added to the output of each layer in the model on the software side. We varied the noise standard deviation (σ) and reported the classification accuracy of the model for each σ. In parallel to this process, we varied the undervolting level at the FPGA side and tracked the classification accuracy of the model. The results are shown in Figure 3. The results show that the models' accuracy behavior, as a function of σ, matches very well with the trend from FPGA undervolting side across all evaluated models. Therefore, with some calibration, the FPGA undervolting noise injection can be accurately replicated in the software model.

III. Threat Model

Similar to the prior MIA defenses [17]–[19], we consider a black-box threat model in this work. In particular, we assume that the adversary can query the target model (F) and see its outputs but not any information from within the model. Specifically, for a target sample, attackers have access to the target model's logits, confidence vector, loss, predicted label,

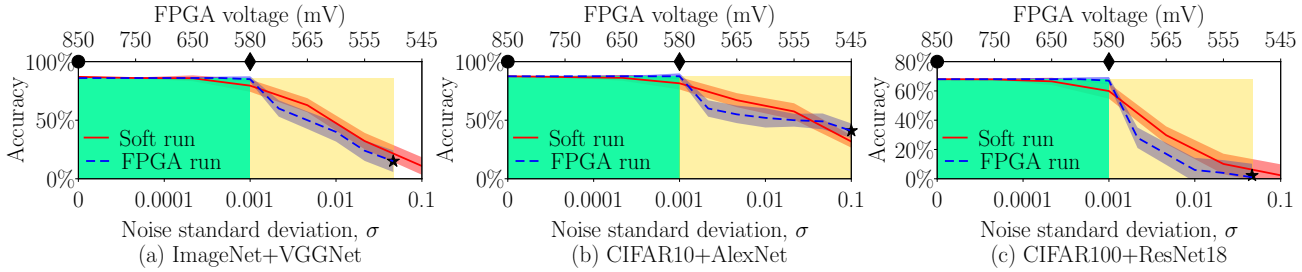


Fig. 3: Effects of stochastic noise and FPGA undervolting on utility or classification accuracy. (V_{nom}: , V_{min}: ♦, V_{crash}: ★). V_{nom}, V_{min}, and V_{crash} are nominal, end of safe region, and start of crash region voltages, respectively.

entropy, etc. Adversaries exploit these black-box features and either train an attack model or perform thresholding on the metrics distribution to infer membership. Furthermore, similar to prior works [17], [19], [34], we assume that the adversary knows some members; they have access to a ratio of the training data samples. In other words, we assume that the adversary has access to part of the members or at least samples from the same underlying distribution as the training data. Thus, the adversary goal is to leak the membership of other samples.

IV. Approach: Privacy Preserving Volt (V_{PP})

In this work, we propose Privacy Preserving Volt (V_{PP}), a new inference time randomness-based approach that relies on undervolting for privacy-preserving ML without sacrificing protected models' utility. The hypothesis behind MIAs is that ML systems retain information about the private dataset at training time, which they are susceptible to leak at inference time through the confidence levels associated with each class in samples classification. The whole MIA approach relies on the confidence bias at the inference of samples from the training dataset. **Our intention is to obfuscate this inference bias using stochastic computations**. Therefore, V_{PP} injects stochastic noise in a number of layers' computations during inference. The intended impact of random noise injection is to garble the metrics of the model's behavior on members and non-members, *i.e.*, to obfuscate MIAs features, to prevent an adversary from inferring private information. For our approach to remain practical, our objective is to additionally maintain the baseline accuracy. Specifically, we exploit ML models' noise tolerance property to explore the space for the highest possible noise under accuracy constraint.

Design objectives – The objectives that guide our design are:

1) *Preserving privacy:* Our goal is to design a practical defense against MIAs that guarantees the lowest information leakage on training samples. Therefore, the proposed defense needs to be rigorously and systematically evaluated to ensure that it can achieve high membership privacy (*i.e.*, low MIA accuracy) across a broad range of MIAs.
2) *Maintaining utility:* For a defense to be practically useful, privacy protection should not come at the cost of the baseline accuracy. Therefore, our goal is to protect the model against MIA without degrading its utility.
3) *Easy deployment:* Our goal is to propose a solution that does not interfere with the baseline training process. Hence, our defense needs to be applied at inference time and can be used to protect pre-trained models.

In this section, we will describe our proposed approach as follows: (i) study the impact of stochastic noise on privacy, (ii) describe our space exploration methodology.

A. Adding noise as a privacy defense

We propose to leverage stochastic computation noise to preserve the privacy of the ML classifiers against MIAs. To keep the noise under control, V_{PP} applies undervolting to a selected set of layers and not necessarily to the whole model. In particular, to maximize privacy, V_{PP} injects noise that is: (i) strong enough to obfuscate the protected model behavior on both members and non-members, and (ii) remains within the noise tolerance of the model to preserve the model's accuracy. Therefore, V_{PP} leverages the noise-tolerance property of ML models to add noise to the model computations to maximize privacy while preserving the model's utility.

Let a model $\mathcal{F}_n$ composed of n layers $\ell_i, i = 1..n$, s.t., $\mathcal{F}_n(.) = \ell_n \circ \ell_{n-1} \circ ... \ell_1(.)$. A conventional neuron $n(\boldsymbol{x})$ within a given layer ℓ outputs $n(x) = \psi(\boldsymbol{w}^\top \boldsymbol{x})$, where ψ is the activation function, $\boldsymbol{w} \in \mathbb{R}^d$ is the weights and $\boldsymbol{x} \in \mathbb{R}^d$ is the input features to the neuron.

Technically, the proposed technique consists of injecting additive layer-wise Gaussian noise within the computations. We note $\tilde{\ell}_i$ a stochastic layer, and a stochastic model $\tilde{\mathcal{F}}_n^d$ is composed of d stochastic layers, and expressed as:

$$\tilde{\mathcal{F}}_n^d(.) = \ell_n \circ \ell_{d+1} \circ \tilde{\ell}_d \circ ... \tilde{\ell}_1(.) \tag{1}$$

More specifically, a stochastic layer is composed of stochastic neurons where the noise is injected between the matrix multiplication and the activation function. The inference-time stochastic neuron $\tilde{n}_i(\boldsymbol{x})$ within a stochastic layer $\tilde{\ell}_i$ would then be expressed by:

$$\tilde{n}_i(\boldsymbol{x}) = \psi(\boldsymbol{w}^\top \boldsymbol{x} + \alpha), \quad \texttt{s.t.} \quad \alpha \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

Where α is a random variable that follows a normal distribution with a 0 mean and a variance σ^2.

From Equations 1 and 2, there are two variables that would shape the noise impact on the model behavior, both in terms of accuracy and privacy: stochasticity depth d and the noise variance σ^2. However, the impact of these stochasticity parameters (d, σ) is also dependent on the model hyperparameters. In fact, different networks consist of different number of layers, types of each layers (*e.g.*, convolutions/fully connected), and the size of each layer. As a result, a given combination (d, σ) has a different impact on each network in terms of utility loss and privacy gain.

To illustrate this, Figure 4 shows an experiment where we varied the number of stochastic layers (d) and noise standard deviation (σ) and observed their effect on both privacy and utility; this result is obtained for ResNet50 trained on ImageNet dataset. The results show that while higher noise magnitude leads to higher privacy, it also leads to lower accuracy. Similarly, while injecting noise in more layers leads to higher privacy, it also leads to lower accuracy. Therefore, a design space exploration is needed to be able to find the optimal solution that achieves the best privacy-utility tradeoffs. Thus, in this paper, we propose a design space exploration framework based on a multi-objective genetic algorithm. An overview of the design space exploration framework is shown in Figure 5, and we describe the details in the next sub-Section.

B. Design space exploration

Our problem can be formulated as a multi-objective optimization with two objectives: minimizing the vulnerability to MIA and minimizing the accuracy drop due to the injected noise. Let h be a model parameterized by the noise standard deviation σ and the stochasticity depth d (the number of stochastic layers). Notice that for $(\sigma, d) = (0, 0)$, we have a baseline model, *i.e.*, no stochasticity. We note $\alpha_{MIA}(h)$ the accuracy of the inference attack, and we define $\mathcal{L} = 1 - \mathcal{A}_{\sigma,d}(.)$ as the model loss of accuracy. Our problem could be formulated as the following multi-objective optimization problem:

$$\begin{aligned} &\textit{Minimize} \quad \left(\mathcal{V}_{\sigma,d}(h), \mathcal{L}_{\sigma,d}(h)\right), \\ &\text{s.t.} \quad \sigma \leq \sigma_{max}, \text{and} \quad d \leq d_{max} \end{aligned} \tag{3}$$

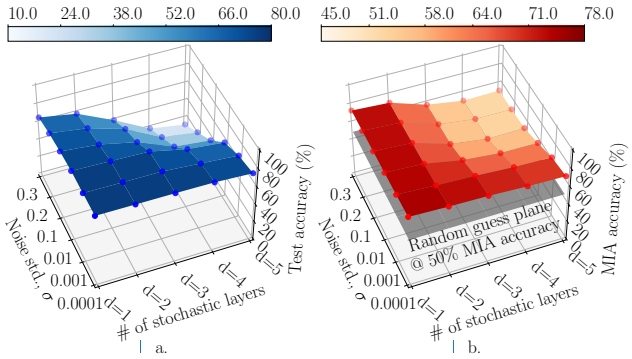


Fig. 4: Effect of σ and d on (a) Test accuracy (*i.e.*, utility) and (b) MIA accuracy (*i.e.*, privacy risk) for ImageNet+ResNet50.

Where $\mathcal{V}_{\sigma,d}(.) = |\alpha_{MIA}(h) - 0.5|$ is the model vulnerability to MIA. To solve this multi objective problem, we use a Multi-Objective Genetic Algorithm (MOGA) detailed in Algorithm 1, which explores for the Pareto front but also proposes a solution within the Pareto front that maximizes the model privacy; optimal solutions of σ and d for different datasets and models are shown in Table I.

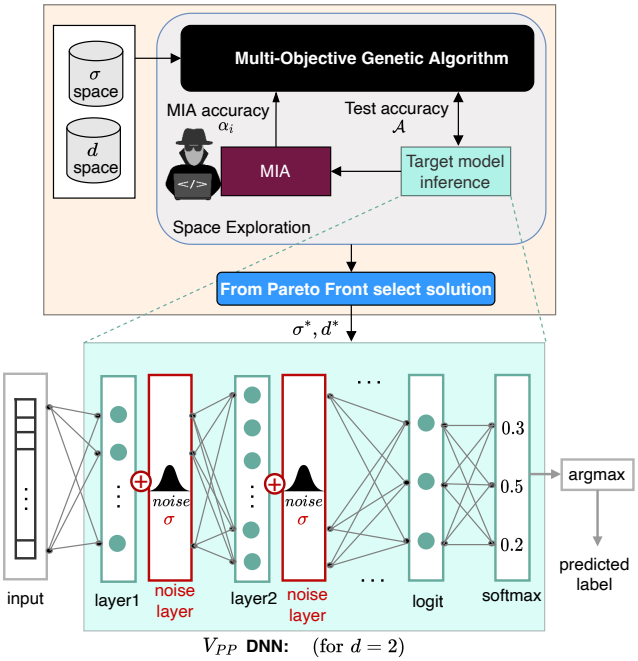


Fig. 5: Overview of V_{PP} approach: Gaussian noise injection to a typical DNN model. The figure shows a sample V_{PP} for d = 2.

Algorithm 1: Multi-Objective Genetic Algorithm

Input: Initial Population $\mathcal{P}$, A Model h;
Output: A set of non-dominated solutions and a proposed solution $(\mathcal{S}, S^*)$;

$\mathcal{S} \leftarrow \phi$;
for *all* $X_i = (\sigma_i, d_i) \in \mathcal{P}$ *s. t.* $\sigma_i \in [0, \sigma_{max}]$ *and* $d_i \in [0, d_{max}]$ **do**
  $k_i = Rank(X_i)$; *// # of dominated solutions by X_i + 1*
  / Selection of non dominated solutions*/*
  if $k_i == Sizeof\ (\mathcal{P})$ **then**
    Append $(\mathcal{S}, X_i)$;
  end
end
G= Crossover($\mathcal{S}$);
G= Mutation(G);
$\mathcal{P} = G \cup \mathcal{S}$;
/ Not yet hitting maximum iterations or solutions still improving */*
if *Stopping condition not met* **then**
  goto step 3;
end
$S^* = \arg\min_X(\mathcal{V}_X(h)) \quad s.t. \quad X = (\sigma, d) \in \mathcal{S}$;
return $(\mathcal{S}, S^*)$;

TABLE I: Optimal solution of σ and d for different datasets and models. FC: stands for Fully Connected DNN model.

Dataset + Model	Optimal solution	
	σ	d
Purchase100 + FC	0.01	$d = 3$
Texas100 + FC	0.1	$d = 3$
CIFAR10 + AlexNet	0.01	$d = 4$
CIFAR100 + ResNet18	0.01	$d = 4$

V. Experimental Setup

This section details the datasets/models, the MIAs, and the evaluation framework that have been used.

A. Datasets and Models

We follow prior works on MIAs [13], [14] and defenses [12], [17], [19] to select the representative datasets and models. We briefly summarize them below.

Purchase100 [35]: It is a dataset comprising 197,324 binary feature vectors. Each vector has 600 features corresponding to different *products*; the binary value for each feature indicates whether the product is purchased or not for a given customer sample, and the label indicates the *purchase habit* of a customer.

Texas100 [36]: It is a dataset comprising 67,300 binary feature vectors. Each vector has 6,170 features corresponding to different *symptoms*; the binary value for each feature indicates whether the symptom is present or not in a patient, and the label indicates the *treatment* given to a patient.

CIFAR-10 and CIFAR-100 [31] : CIFAR-10 is an image classification dataset containing a total of 60,000 color images representing 10 different classes; it has 5000 training images and 1000 testing images for each class. Again, CIFAR-100 is a collection of 60,000 color images representing 100 classes.

We split the dataset for the target and attack models, as shown in Table II. We denote the training data and test data of target model as D_{tr} and D_{te} respectively; we used $|D_{tr}|=|D_{te}|$ to build an unbiased target model. Again, $D^{'}_{tr}$ and $D^{'}_{te}$ denote the training data and test data of the attack model, respectively. As mentioned in the threat model, we assume that the attacker has partial access to the target model dataset. Following all prior works, we also assume an equal number of members and nonmembers in $D^{'}_{tr}$ and $D^{'}_{te}$.

TABLE II: Dataset split

Dataset	Target model		Attack model	
	Train $\lvert D_{tr}\rvert$	Test $\lvert D_{te}\rvert$	Train $\lvert D^{'}_{tr}\rvert$	Test $\lvert D^{'}_{te}\rvert$
Purchase100	10000	10000	5000	5000
Texas100	10000	10000	5000	5000
CIFAR10	25000	25000	12500	12500
CIFAR100	25000	25000	12500	12500

Models: We train target models on Purchase and Texas datasets using fully connected (FC) deep neural networks, having six-layer architecture such as [features, 1024, 512, 256, 128, classes] and Tanh() activation after each layer. We denote their corresponding models by Purchase100+FC and Texas100+FC, respectively. We used AlexNet [28] for CIFAR10 dataset and ResNet18 [29] for CIFAR100 dataset; as such, their corresponding models are denoted by CIFAR10+AlexNet and CIFAR100+ResNet18 respectively.

B. Implemented Attacks

Here, we summarize four membership inference attacks implemented in our paper. Specifically, we selected three powerful score-based attacks and a decision-based (*e.g.*, label-only) attack to evaluate the effectiveness of our defense. Score-based MIAs are mounted using various output scores of the target model that are available before obtaining the predicted label. On the contrary, decision-based MIAs are performed using only the decision of the target model, which means the hard label or predicted label. We outline the implemented attacks more precisely as follows:

(i) Bounded loss (BL) attack (I_{bl}): Yeom et al. [37] used this attack approach, where they inferred members by applying a threshold (τ) on the loss of the target model on the target sample. They used 0-1 loss, meaning that the target model's gap between train and test accuracy indicates attack accuracy. This is a score-based attack, and we denote this attack by I_{bl}.

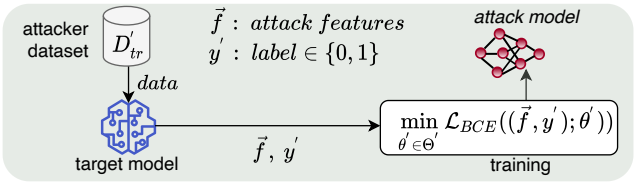
(ii) NSH attack (I_{bb}): Nasr, Shokri, and Houmansadr (NSH) [14] proposed this attack approach, where they used black box features of the target model on target samples. More specifically, they combined the target model's class probability (or the confidence) with the loss to train a binary attack model for inferring membership. This is also a score-based attack, and we denote this attack as I_{bb}.

(iii) NN attack (I_{nn}): Salem et al. [38] introduced this attack approach, where they trained a neural network using the target model's prediction (logits) as a feature to infer the membership. This is yet another score-based attack, and we denote this attack by I_{nn}.

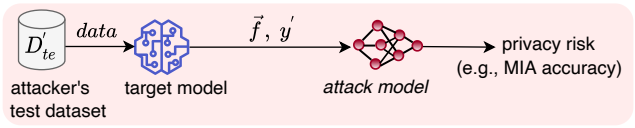
(iv) Label-only attack: Unlike I_{bl}, I_{bb}, and I_{nn} attacks, where different output scores of the target model are used to mount an attack, label-only attacks [24], [39] fall in the category of decision-based attack, which assume the availability of target model's predicted labels only. In many real-world ML-as-a-service applications, ML model does not necessarily publish the confidence scores but the predicted label only [24], [39]. After obtaining labels, attackers iteratively insert the necessary amount of adversarial noise or perturbation to the input samples in order to change their predicted labels. Then the distortion between the original inputs and their corresponding perturbed inputs are measured and exploited to perform attacks, *i.e.*, differentiate between members and non-members. In fact, label-only attacks exploit the fact that member samples are farther from the decision boundary than non-member samples, implying that member samples need greater distortion than non-members to change their predicted labels. Consequently, simple thresholding on the adversarial distortion can differentiate between members and non-members.

C. Experimental Methodology

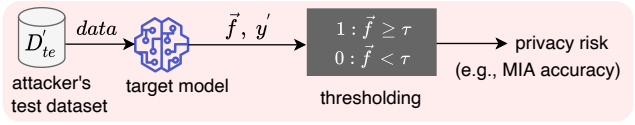
Here, we present our experimental methodology: training the attack models and mounting the membership inference attacks.



(a) Training an attack model



(b) I_{bb} and I_{nn} attack



(c) I_{bl} attack

Fig. 6: Experimental methodology— (a) shows training an attack model; (b) and (c) show different MIA methods.

As shown in Figure 6 (a), we train a target model (F) on its training dataset (D_{tr}) using cross-entropy loss function; at this stage, F is an unprotected target model. Then we extract *attack features* ($\vec{f}$) of target model by making an inference using attacker training dataset (D'_{tr}). Table III summarizes the attack features for each attack. Once the attack features are extracted, we train the attack model (or inference model) (I) using binary cross-entropy loss. Additionally, Figure 6 (b) shows the I_{bb} and I_{nn} membership inference attacks (MIAs) method; here attack features are collected for attacker test data D'_{te} and subsequently used by the attack model to measure privacy risk. Furthermore, Figure 6 (c) shows the I_{bl} attack method that does not use an attack model; it rather uses threshold (τ) on the attack feature to determine membership and privacy risk.

TABLE III: Attack features of different attack variants.

Attacks	Attack features $\vec{f}$	Attack type
I_{bb} [14]	Confidence & loss	Score based
I_{nn} [38]	Logit	Score based
I_{bl} [37]	Loss	Score based
Label-only [24]	predicted label	Decision based

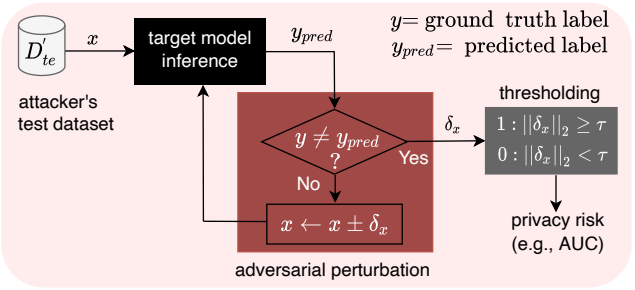
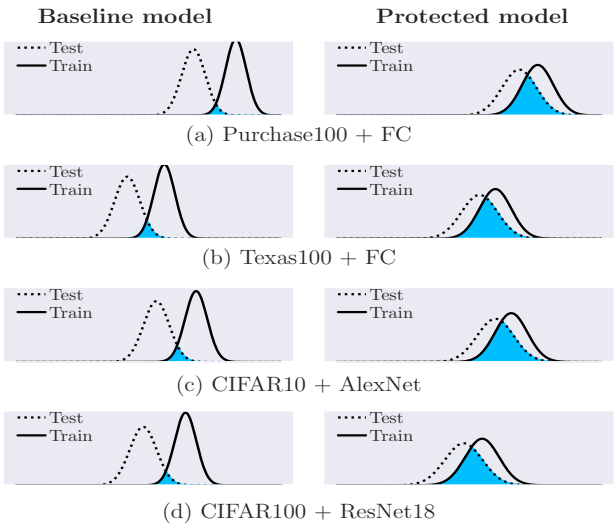


Fig. 7: Label-only attack.

Figure 7 explains the label-only attack, where attackers iteratively inserts perturbation (*i.e.*, $\pm\delta x$) to original input samples (x) to change their labels. Then the distortion between the clean samples and the corresponding perturbed samples is measured, which is essentially the distance of input samples from the decision boundary. Finally, thresholding over the measured distance is used to determine the membership.

VI. Privacy Analysis

Score-based Attacks – In this section, we present the privacy risk of unprotected and protected target models. We also compare the privacy risk of our defense with that of the existing defenses. Following prior works, we measure the privacy risk as the MIA accuracy. Since attackers predict as either member or non-member, privacy risk becomes the least when attackers are most uncertain, which happens when the prediction is a random guess or the MIA accuracy is 50%. Alternatively, MIA accuracy higher than 50% indicates a higher privacy risk.



(a) Purchase100 + FC

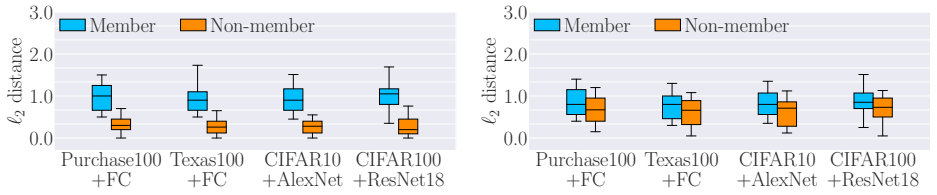
(b) Texas100 + FC

(c) CIFAR10 + AlexNet

(d) CIFAR100 + ResNet18

Fig. 8: Impact of stochastic noise on confidence distribution of train and test samples. Figures on the left side are for undefended model and figures on the right side are for V_{PP} protected model.

Figure 8 shows the confidence distribution of train (member) and test (non-member) samples. Baseline models (figures on the left side) have small overlaps between member and non-member scores, showing a little confusion and greater separability. On the other hand, V_{PP} protected models (figures on the right side) demonstrate greater overlaps, meaning greater confusion and little separability. Table IV summarizes the MIA accuracy (i.e, privacy risk) of unprotected models and protected models for various defenses. From the table, while unprotected models suffer from high privacy risks, V_{PP} protected models significantly reduce the privacy risks. This is because of the garbling in the attack feature space shown in Figure 8. Comparing with the most recent state-of-the-art defense SELENA, our defense (*i.e.*, V_{PP}) offers 2.01%, 0.63%, and 1.36% lower privacy risks on CIFAR100 dataset for I_{bb}, I_{bl}, and I_{nn} attacks respectively. For Purchase100, Texas100, and CIFAR10 datasets, V_{PP} also shows comparable or lower privacy risks as contrasting other defenses.



(a) Unprotected models (b) V_{PP} protected models

Fig. 9: Distortion, measured in ℓ_2 distance, for members and non-members to change their labels.

TABLE IV: Privacy risk for different defenses. Minimum privacy risk in each attack category (I_{bb}, I_{bl}, and I_{nn}) is marked in bold face.

Dataset + model	Defense	MIA accuracy (privacy risk)		
		I_{bb}	I_{bl}	I_{nn}
Purchase100 + FC	None	77.09%	63.6%	62.2%
	AdvReg	55.4%	54.9%	**50.1%**
	DMP	55.1%	55.2%	50.2%
	SELENA	53.3%	**53.2%**	53.3%
	V_{PP} **(Ours)**	**53.27%**	**53.2%**	50.16%
Texas100 + FC	None	76.23%	76.2%	72.0%
	AdvReg	57.9%	54.1%	50.8%
	DMP	55.4%	53.6%	**50.0%**
	SELENA	55.1%	54.8%	52.2%
	V_{PP} **(Ours)**	**53.6%**	**52.4%**	50.07%
CIFAR10 + AlexNet	None	78.1%	66.6%	66.5%
	AdvReg	51.2%	52.1%	53.14
	DMP	**50.6%**	**51.6%**	51.65
	SELENA	54.1%	53.5%	51.7%
	V_{PP} **(Ours)**	51.94%	51.85%	**50.83%**
CIFAR100 + ResNet18	None	77.49%	68.3%	67.1%
	AdvReg	53.4%	53.6%	53.48
	DMP	54.4%	53.7%	51.92
	SELENA	55.1%	54.0%	52.0%
	V_{PP} **(Ours)**	**53.38%**	**53.37%**	**50.64%**

Decision-based Attack: Label-only Attack – Score-based attacks shown earlier have a drawback. Specifically, they can be easily mitigated/averted if the target model only exposes its final decision (*i.e.*, top-1 predicted label) but not any confidence score. It is, in fact, the case in many real-world ML-as-a-service applications, where ML models provide only the labels rather than their scores [24], [39]. Additionally, label-only attacks have the potential to be robust against some perturbation-based defenses (*e.g.*, [17], [18], [40]). Thus, it is important to evaluate our defense against label-only attacks.

Figure 9 shows the ℓ_2 measure of the distortion (or distance) of members and non-members across different datasets. Figure 9 (a) represents the case of unprotected models, which clearly shows a huge gap between their distances and is thus distinguishable simply by thresholding. On the other hand, Figure 9 (b) shows the case of V_{PP} protected models, where members and non-members have significant overlaps in their distortion distribution, lowering the efficacy of the distance feature in differentiating members from non-members. Such overlaps in the distortion space result from the defensive distortion introduced by V_{PP} as stochastic noise.

Figure 10 shows the privacy risk of different defenses against label-only attacks. Following existing label-only attacks [24], [39], we utilize the receiver operating characteristic (ROC) curve and measure the privacy risk using the area under the curve (AUC). As expected, baseline or unprotected models suffer from high privacy risks across all datasets considered. Furthermore, it shows that MemGuard fails to defend against label-only attacks and has the same privacy risk as undefended models. It is because MemGuard adds noise to the confidence score under the constraint of not changing the predicted label. In contrast, our defense (V_{PP}) offers 0.6% and 0.4% lower privacy risk than the latest defense SELENA for Texas100 and CIFAR10 dataset, respectively. Moreover, compared to adversarial regularization (AdvReg), V_{PP} offers 3%, 2.1%, 7.4%, and 4.2% lower privacy risk on Purchase100, Texas100, CIFAR10, and CIFAR100 datasets, respectively.

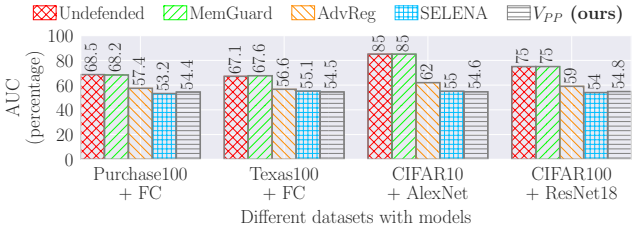


Fig. 10: Effectiveness of defenses against label-only attack.

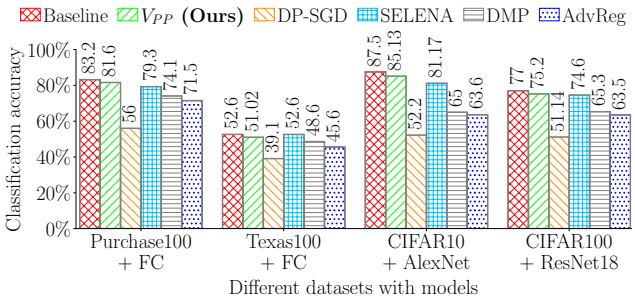


Fig. 11: Defenses effect on classification accuracy (*i.e.*, utility).

VII. Impact on Baseline Accuracy

Baseline accuracy is the classification accuracy without applying a defense; accuracy of unprotected model. Following the most recent defenses against MIAs, we use the same datasets and models in our evaluation. Target model's classification accuracy is also called the utility, which is expected to be maintained as high as possible. However, privacy attacks are resisted by perturbing either the learning outcome, model gradients, or input itself, which in turn reduces the utility. Our defense (*i.e.*, V_{PP}) is also perturbation based, but contrasting

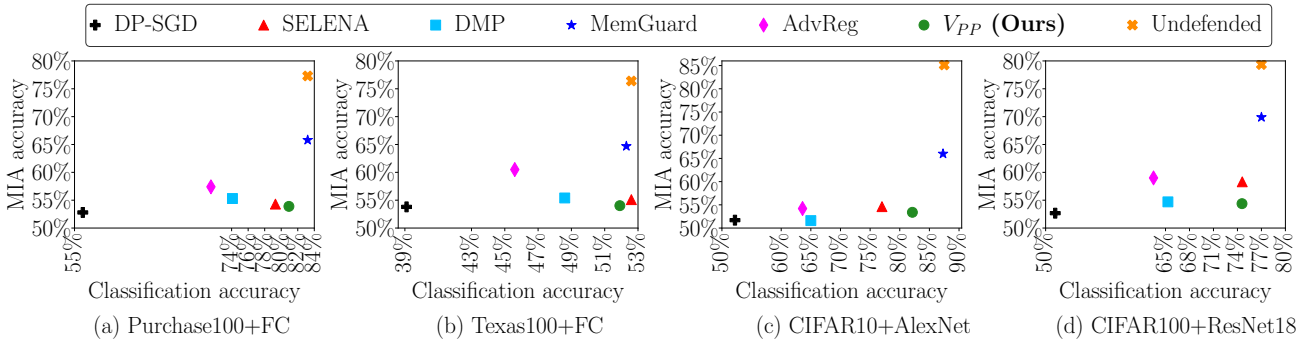


Fig. 12: Comparison of defenses in terms of utility and privacy tradeoff.

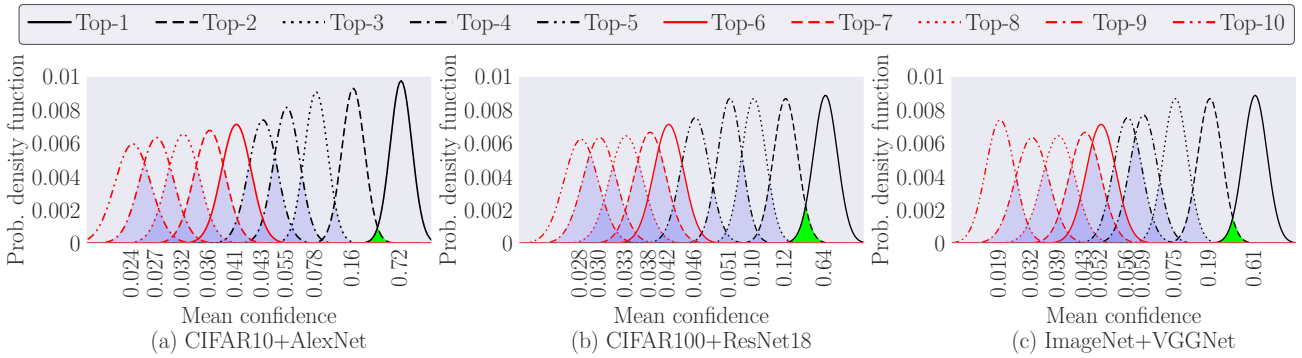


Fig. 13: Prediction confidence distribution of V_{PP} protected models. Green overlapping leads to classification accuracy (*i.e.*, utility) loss and other overlappings indicate confusion that drops MIA accuracy (*i.e.*, privacy risk).

existing methods, we perturb model computation by injecting additive Gaussian noise to models' computations. Thus, V_{PP} is also expected to reduce the utility. We show the utility of V_{PP} based protected model in green color in Figure 11. Compared with baseline models, our proposed V_{PP} based defense loses the accuracy by 1.6% for Purchase100+FC, 1.58% for Texas100+FC, 2.37% for CIFAR10+AlexNet, and 1.8% for CIFAR100+ResNet18. In Figure 11, we also compare the model accuracy under our defense (*i.e.*, V_{PP}) with that of other existing defenses such as DP-SGD, SELENA, DMP, and AdvReg. We will compare our V_{PP} defense with the best performing existing defense, which is SELENA for all datasets. SELENA is also the most recent among all existing defenses. Thus, compared with SELENA, Figure 11 shows that V_{PP} yields 2.3% higher utility on Purchase100 dataset, 1.58% lower utility on Texas100 dataset, 3.96% higher utility on CIFAR10 and 0.6% higher utility on CIFAR100 dataset.

Utility/Privacy Tradeoffs – Defense techniques aim to reduce MIA accuracy (*i.e.*, privacy risk), which also reduce the target model's classification accuracy (*i.e.*, utility). However, for a considerable defense solution, we expect higher utility but lower privacy risk. Therefore, we compare the utility-privacy tradeoff of our defense, *i.e.*, V_{PP} , with other defenses in Figure 12. We report the privacy score of the best attack (*i.e.*, the highest MIA accuracy of the three attacks).

Differential privacy based stochastic gradient descent (DP-SGD) is the earliest defense, which offers strong membership privacy guarantee by adding Gaussian noise to model gradients. While preserving privacy, DP-SGD (for $\varepsilon = 4$) incurs huge utility loss compared to baseline model (*e.g.*, losing 27.2% on Purchase100 , 13.5% on Texas100, 35.3% on CIFAR10, and 25.8% on CIFAR100). On the stark contrast, MemGuard resists attackers from gaining reliable access to model behavior by adding noise to model output while preserving correct prediction. Thus, MemGuard ideally does not lose utility; however, it cannot guarantee promising privacy, as evident by its highest MIA accuracy among all the defenses.

However, Figure 12 shows that other defenses offer comparatively better tradeoff than DP-SGD and MemGuard. Interestingly, our proposed V_{PP} outperforms on Purchase100 and CIFAR10 datasets and shows comparable tradeoff with the state-of-the-art defense SELENA on CIFAR100 and Texas100.

VIII. Why does stochastic noise help?

This section illustrates why stochastic noise helps protect privacy while preserving utility. Injecting stochastic noise in model computation obfuscates model's exact behavior, which resists attackers from having reliable access to model output. Such nondeterministic behavior confuses the attack models (I) which in turn reduces the MIA accuracy. To better understand the stochastic noise effect, we take a closer observation in the output confidence vector. We do so because attackers exploit model's output to mount membership inference attacks. As such, we plot the Top-10 confidence distribution of V_{PP} protected models in Figure 13. For all models, we see overlapping between different confidence distributions; these overlapping

areas are the result of the nondeterministic behavior, which is controlled by the variance of the injected noise during computation.

Effect on utility: Utility, *i.e.*, classification accuracy, is determined by Top-1 confidence. Additive Gaussian noise causes overlapping in Top-1 confidence distribution, which is shown in green color in Figure 13. These green overlapping zones indicate the percentage of misclassification, which leads to baseline accuracy loss. All V_{PP} protected models in Figure 13 show small overlapping in Top-1 confidence, which suggests a slight utility drop.

Effect on privacy: Attackers exploits output confidence vector to mount MIA. Confidence distribution other than Top-1 shows multiple levels of overlapping, which indicates the degree of unreliable access that the attackers might have. Such nondeterministic attack features confuse the attack model (I), which drops the MIA accuracy or the privacy risk.

IX. Adaptive Attack

Adversaries naturally attempt to gain insight from the defense to break it. Thus, when proposing a new defense against MIAs, it is necessary to systematically evaluate new defenses against adaptive attack to make sure they can't be easily bypassed. Therefore, in this section: we propose a new adaptive attack against V_{PP} based on the expectation over several queries to compensate the noise impact. Then, we evaluate V_{PP} against the new adaptive attack. Our results demonstrate the adaptive attack was not able to amplify leakage, *i.e.*, get significantly higher attack accuracy compared to prior attacks.

Proposed adaptive attack: V_{PP} depends on injecting additive layer-wise Gaussian noise into the network/model. In Section VI, we showed that V_{PP} can indeed obfuscate the protected model behaviour on both members and non-members via the additive layer-wise Gaussian noise and thus protect the model's privacy. Thus, from an attacker perspective, an adaptive method should be able to take into account the stochastic component of the system, *i.e.*, the Gaussian noise, to bypass it. Therefore, our proposed adaptive attack tries to estimate the model expectation over a set of repeated queries. Specifically, we assume an adaptive attacker is able to repeatedly query the target model for each sample to collect a set of attack features per sample. The collected set of attack features per sample represents a wider spectrum of attack features variations. Accordingly, the attacker needs to average the collected set of attack features per sample to get the expectation of the model behavior under the assumption of a zero-mean Gaussian noise model. Please notice that this needs to be done per sample regardless if the attacker is using it to generate data to be used in training the attack model or to infer whether a specific sample is a member or non-member. As such, we develop adaptive attack as described in Algorithm 2, which outputs adaptive attack models based on I_{bb} attack. Line 3, 4, and 5 shows the averaging of different attack features for n runs.

Algorithm 2: /*Adaptive I_{bb} attack */

Input: $D^{'}_{tr}$: Attacker data (member+nonmember);
$\tilde{F}$: V_{PP} protected target model;
n: # of queries per input sample;
Output: I^{bb}_{adap}: adaptive attack model;

1 $n \leftarrow 10000$; *// Assign a large number*
2 **for** *all* $(\boldsymbol{x}, y) \in D^{'}_{tr}$ **do**
3 $\quad \mathbb{E}(\tilde{F}(\boldsymbol{x})) \leftarrow \frac{1}{n}\sum_{i=0}^{n-1} \tilde{F}(\boldsymbol{x})$; *// $\tilde{F}(.)$: logit*
4 $\quad \mathbb{E}(f(\tilde{F}(\boldsymbol{x})_i)) \leftarrow \frac{1}{n}\sum_{i=0}^{n-1} \frac{e^{\tilde{F}(\boldsymbol{x})_i}}{\sum_{j=1}^{K} e^{\tilde{F}(\boldsymbol{x})_j}}$;
5 $\quad \mathbb{E}(\mathcal{L}) \leftarrow -\frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{k-1} y_j \log \tilde{F}(\boldsymbol{x})_j$; *// $\mathcal{L}$: CE loss*
end
/ construct black-box adaptive attack features, $\vec{f}_{bb}$ */*
6 $\vec{f}_{bb} \leftarrow \mathbb{E}(\tilde{F}(\boldsymbol{x}))||\mathbb{E}(f(\tilde{F}(\boldsymbol{x})))||\mathbb{E}(\mathcal{L})$
/ train adaptive attack classifier using binary cross entropy loss */*
7 $I^{bb}_{adap} \leftarrow \min_{\theta' \in \Theta'} \mathcal{L}_{BCE}((\vec{f}_{bb}, y^{'}); \theta^{'})$;
8 **return** I^{bb}_{adap};

Adaptive attack evaluation: We varied n to be 20, 40, 60, 80, 100, 500, 1000, 5000, and 10000. Then we measure the MIA accuracy (*i.e.*, privacy risk) of adaptive attack. The result is shown if Figure 14. The result shows that increasing the number of queries per input sample helps attackers increase the privacy leakage. In other words, attackers might see higher MIA accuracy when applying more queries per sample. However, as shown in Figure 14, the MIA accuracy gain is not significant and saturates for most models after $n = 10000$.

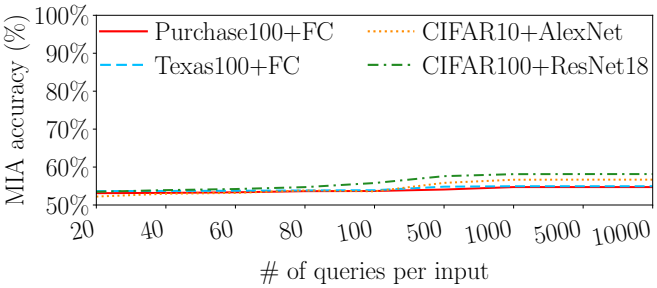


Fig. 14: MIA accuracy (*i.e.*, privacy risk) of adaptive attack.

X. Conclusion

We propose a lightweight, effective defense called Privacy Preserving Volt (V_{PP}) for preserving ML privacy while maintaining utility. V_{PP} injects computational noise to a set of layers of the protected model during inference time through undervolting. Interestingly, since noise is injected during inference, V_{PP} does not require retraining the model and can be deployed on off-the-shelf pre-trained models.

Acknowledgment

This work has been supported in part by EdgeAI KDT-JU European project (101097300) and the National Science Foundation grant CCF-2212427.

REFERENCES

[1] N. Yanes, A. M. Mostafa, M. Ezz, and S. N. Almuayqil, "A machine learning-based recommender system for improving students learning experiences," *IEEE Access*, vol. 8, pp. 201 218–201 235, 2020.

[2] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature communications*, vol. 11, no. 1, p. 3923, 2020.

[3] M. S. Islam, K. N. Khasawneh, N. Abu-Ghazaleh, D. Ponomarev, and L. Yu, "Efficient hardware malware detectors that are resilient to adversarial evasion," *IEEE Transactions on Computers*, 2021.

[4] S. Islam, I. Alouani, and K. N. Khasawneh, "Lower voltage for higher security: Using voltage overscaling to secure deep neural networks," in *ICCAD*, 2021.

[5] M. S. Islam, B. Omidi, and K. N. Khasawneh, "Monotonic-hmds: Exploiting monotonic features to defend against evasive malware," in *ISQED*, 2021.

[6] M. S. Islam, I. Alouani, and K. N. Khasawneh, "Enhancing hardware malware detectors' security through voltage over-scaling," in *2021 5th ACM SIGARCH Workshop on Cognitive Architectures*.

[7] M. S. Islam, A. P. Kuruvila, K. Basu, and K. N. Khasawneh, "Nd-hmds: Non-differentiable hardware malware detectors against evasive transient execution attacks," in *ICCD*, 2020.

[8] F. Behnia, A. Mirzaeian, M. Sabokrou, S. Manoj, T. Mohsenin, K. N. Khasawneh, L. Zhao, H. Homayoun, and A. Sasan, "Code-bridged classifier (cbc): A low or negative overhead defense for making a cnn classifier robust against adversarial attacks," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2020, pp. 27–32.

[9] K. N. Khasawneh, N. B. Abu-Ghazaleh, D. Ponomarev, and L. Yu, "Adversarial evasion-resilient hardware malware detectors," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–6.

[10] A. Guesmi, K. N. Khasawneh, N. Abu-Ghazaleh, and I. Alouani, "Room: Adversarial machine learning attacks under real-time constraints," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–10.

[11] H. M. Makrani, H. Sayadi, N. Nazari, K. N. Khasawneh, A. Sasan, S. Rafatirad, and H. Homayoun, "Cloak & co-locate: Adversarial railroading of resource sharing-based attacks on the cloud," in *2021 International Symposium on Secure and Private Execution Environment Design (SEED)*. IEEE, 2021, pp. 1–13.

[12] A. Houmansadr and V. Shejwalkar, "Membership privacy for machine learning models through knowledge transfer," in *35th AAAI Conference on Artificial Intelligence, 2021*.

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

[14] 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 *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[15] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays," *PLoS genetics*, vol. 4, no. 8, p. e1000167, 2008.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[17] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 634–646.

[18] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 259–274.

[19] X. Tang, S. Mahloujifar, L. Song, V. Shejwalkar, M. Nasr, A. Houmansadr, and P. Mittal, "Mitigating membership inference attacks by self-distillation through a novel ensemble architecture," *arXiv preprint arXiv:2110.08324*, 2021.

[20] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *International Conference on Learning and Representation (ICLR)*, 2017.

[21] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with pate," in *International Conference on Learning and Representation (ICLR)*, 2018.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 880–895.

[25] S. Majumdar, M. H. Samavatian, K. Barber, and R. Teodorescu, "Using undervolting as an on-device defense against adversarial machine learning attacks," in *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2021, pp. 158–169.

[26] Xilinx, "Chaidnn-v2: Hls based deep neural network accelerator library for xilinx ultrascale+ mpsocs," https://github.com/Xilinx/CHaiDNN, [Online; accessed 20-October-2022].

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *ACM Neural information processing systems (NeurIPS)*, vol. 60, no. 6, pp. 84–90, 2017.

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

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

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

[32] D. Jeon, M. Seok, Z. Zhang, D. Blaauw, and D. Sylvester, "Design methodology for voltage-overscaled ultra-low-power systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 12, pp. 952–956, 2012.

[33] J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, ser. CASES '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 158–168. [Online]. Available: https://doi.org/10.1145/1176760.1176781

[34] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2615–2632.

[35] P. 2017, "Acquire valued shoppers challenge," https://www.kaggle.com/c/acquire-valued-shoppers- challenge/data, [Online; accessed 11-September-2019].

[36] T. 2017, "Texas hospital stays dataset," https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm, [Online; accessed 10-February-2020].

[37] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 268–282.

[38] 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 *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*, 2019.

[39] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1964–1974.

[40] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," *arXiv preprint arXiv:2005.03915*, 2020.