Membership Inference Attacks and Defenses in Neural Network Pruning

Xiaoyong Yuan and Lan Zhang, *Michigan Technological Unviersity*

https://www.usenix.org/conference/usenixsecurity22/presentation/yuan-xiaoyong

This paper is included in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

Membership Inference Attacks and Defenses in Neural Network Pruning

Xiaoyong Yuan, Lan Zhang
Michigan Technological University

Abstract

Neural network pruning has been an essential technique to reduce the computation and memory requirements for using deep neural networks for resource-constrained devices. Most existing research focuses primarily on balancing the sparsity and accuracy of a pruned neural network by strategically removing insignificant parameters and retraining the pruned model. Such efforts on reusing training samples pose serious privacy risks due to increased memorization, which, however, has not been investigated yet.

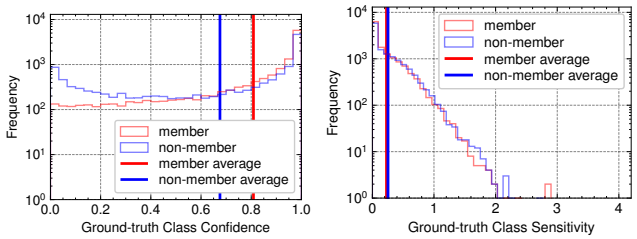
In this paper, we conduct the first analysis of privacy risks in neural network pruning. Specifically, we investigate the impacts of neural network pruning on training data privacy, *i.e.*, membership inference attacks. We first explore the impact of neural network pruning on prediction divergence, where the pruning process disproportionately affects the pruned model's behavior for members and non-members. Meanwhile, the influence of divergence even varies among different classes in a fine-grained manner. Enlightened by such divergence, we proposed a self-attention membership inference attack against the pruned neural networks. Extensive experiments are conducted to rigorously evaluate the privacy impacts of different pruning approaches, sparsity levels, and adversary knowledge. The proposed attack shows the higher attack performance on the pruned models when compared with eight existing membership inference attacks. In addition, we propose a new defense mechanism to protect the pruning process by mitigating the prediction divergence based on KL-divergence distance, whose effectiveness has been experimentally demonstrated to effectively mitigate the privacy risks while maintaining the sparsity and accuracy of the pruned models.

1 Introduction

Much of the progress in artificial intelligence over the past decade has been the result of deep neural networks (DNNs). The powerful DNNs with a large number of parameters consume considerable storage and memory bandwidth, which makes it challenging to deploy the state-of-the-art neural networks on resource-constrained devices. To address this issue, neural network pruning as one of the most popular compression technologies has attracted great attention [1, 2]. By removing insignificant parameters from a DNN, recent research has shown that neural network pruning can substantially reduce the size of a DNN and speedup the inference process without largely compromising prediction accuracy [2–5]. In general, neural network pruning includes three main stages: 1) train an original DNN; 2) remove the insignificant parameters; 3) fine-tune the remaining parameters with the training dataset. Most existing research on neural network pruning has focused on improving the trade-off between accuracy and sparsity by strategically designing the last two stages [2–5]. However, such efforts on reusing training samples pose serious privacy risks of the pruned neural networks due to the potentially increased memorization of training samples.

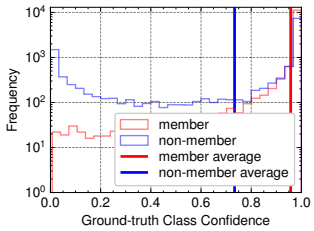
The privacy risks of DNNs have already been pointed out, where a DNN is prone to memorizing sensitive information of the training dataset [6–9]. Taking the membership inference attack (MIA) as an example, an adversary can infer whether a given data sample was used to train a DNN, seriously threatening individual privacy. For instance, an adversary can infer an individual was a confirmed case, if it is known that the individual's record was used to train an infectious disease model. The MIA was first proposed against black-box models in [10], where the adversary only has access to the data sample and predictions of the target model. Later on, more attention has been attracted against various DNN models, such as generative models [7, 8], graph models [11], machine translation [12], text generation [13], genomic analysis [14], and transfer learning [15]. Although extensive analysis has been conducted, none of the existing efforts have been put into analyzing MIAs against pruned neural networks.

In view of this, the paper focuses on one fundamental question: *comparing with original deep neural networks, are the pruned networks more vulnerable to membership inference attacks?* Specifically, most MIAs infer a sample's membership based on the different behaviors of a target model between

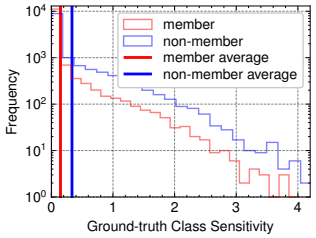


(a) Original model confidence

(b) Original model sensitivity



(c) Pruned model confidence



(d) Pruned model sensitivity

Figure 1: Histograms of the prediction confidences and the prediction sensitivity of the ground-truth label. We remove 70% of the parameters in the original DenseNet121 model using l1 unstructured pruning on the CIFAR10 dataset. The figures show the frequency of prediction confidence (a) and (c) and prediction sensitivity (b) and (d) belonging to the ground-truth class on the training and test data. The vertical lines indicate the average values of training data, *i.e.*, members (black), and test data, *i.e.*, non-members (red), respectively. In both prediction confidence and sensitivity measurements, neural network pruning makes the distances between the two vertical lines in the pruned model larger than that in the original model, which indicates a larger confidence gap and sensitivity gap between members and non-members due to pruning.

members (*i.e.*, training samples) and non-members (*i.e.*, test samples), such as the different prediction confidences [9, 10]. Since most neural network pruning approaches rely on reusing the training dataset to fine-tune the parameters after pruning the insignificant parameters, the additional training at the pruned neural network inevitably increases its memorization of the training samples. Moreover, the pruned neural network enforces a small number of parameters to achieve similar prediction capabilities, which also increases the memorization of training data and makes the pruned model more sensitive to the training data. Hence, such increased memorization can intuitively lead to a larger divergence of the prediction confidences and sensitivities between members and non-members. Figure 1 illustrates the prediction confidence and the prediction sensitivity[1] of members and non-members in the original DNN and the pruned network, respectively. The larger divergence of the confidences and the sensitivities in the pruned model at (c) and (d) confirms our intuition: *neural network pruning can aggravate the privacy issues of the original deep neural network*. Therefore, in the following paper, we conduct a comprehensive analysis to reveal the impacts of neural network pruning on training data privacy, *i.e.*, MIAs. Specifically, we first explore the impact of neural network pruning on prediction divergence: *the pruning process disproportionately affects the pruned model's behavior for members and non-members*. Enlightened by this insight, a new MIA is proposed against the pruned neural networks. In addition, with the proposed new attack, we propose a new defense mechanism to protect the fine-tuning process by mitigating the prediction divergence based on KL-divergence distance. Extensive experiments are conducted to rigorously evaluate our proposals. To the best of our knowledge, this is the first study to investigate the privacy risks of neural network pruning. Our main contributions are summarized below:

- We investigate the privacy risk of neural network pruning and propose a new MIA: self-attention membership inference attack (SAMIA). By exploring the impacts of neural network pruning on prediction divergence, the proposed attack results in high attack accuracy of revealing the membership status from the pruned models. In particular, SAMIA has advantages in identifying the pruned models' prediction divergence by using finer-grained prediction metrics. We recommend SAMIA as a competitive baseline attack model for future privacy risk study of neural network pruning.
- To rigorously evaluate the privacy impacts of different pruning approaches, sparsity levels, and adversary knowledge, we conduct extensive experiments on seven commonly used datasets, four neural network architectures, four pruning approaches, five sparsity levels, and 255 pruned models in total. Experimental results demonstrate the effectiveness of the proposed attacks against pruned neural networks, which further indicates that neural network pruning can aggravate the privacy issues of the original DNN. The adversary can successfully reveal the membership status, even without the knowledge of the pruning approach used in the target model. Furthermore, we evaluate the privacy impacts of different pruning approaches and various sparsity levels.
- To defend the pruned models against MIAs, we propose a new defense mechanism: pair-based posterior balancing (PPB). PPB protects the fine-tuning process of neural network pruning by narrowing down the divergences of posterior predictions and reducing the prediction sensitivities based on their KL-divergence distances. Experimental results demonstrate the effectiveness of the PPB mechanism, which significantly mitigates the privacy risks while maintaining the sparsity and accuracy of the pruned model. Besides, compared with the state-of-the-art defenses, PPB achieves a better trade-off between prediction performance and privacy in most cases.

[1] The definitions of the prediction confidence and the prediction sensitivity are detailed respectively in Section 4.1.

2 Background and Related Work

2.1 Neural Network Pruning

The state-of-the-art neural networks are usually deep and resource hungry, requiring large amounts of computation and memory, which becomes a particular challenge on resource-constrained end devices. As one of the most popular network compression approaches, neural network pruning has attracted great attention in recent years [2–5]. In general, most network pruning studies follow the pruning workflow: "train-prune-finetuning." For example, Han *et al*. [2] proposed to remove the individual parameters with the lowest magnitude. Randomly removing individual parameters reduces the model size, but may not be efficient to facilitate hardware optimization and accelerate the neural network computation. Therefore, many methods were proposed to remove parameters in an organized way by removing a group of parameters (*i.e*., structured pruning). For example, Li *et al*. [3] removed the entire filters with the lowest magnitude in the neural network, which leads to significant speedup compared with the unstructured pruning. Liu *et al*. [4] removed the entire channels according to the corresponding scaling factors in the followed batch normalization layers. In this paper, we investigate the privacy risks of both unstructured and structured pruning approaches.

More recently, new pruning approaches have been proposed, which prune parameters by searching the optimal neural architecture [16, 17] or fine-tune the pruned model by rewinding the parameters to the previous states [18, 19]. The privacy risks discussed in this paper might exist in these new pruning approaches. We will investigate their privacy risks in our future work.

On the other hand, recent efforts have been put into neural network pruning from other important perspectives. Paganini [20] investigated the unfairness and systematic biases in the pruned models. Hooker *et al*. [21] demonstrated the biased performance on different groups and classes after pruning. Given the potential of pervasively implementing neural network pruning, this work targets another critical and urgent aspect regarding neural network pruning, *i.e*., training data privacy.

2.2 Membership Inference Attacks (MIAs)

Membership inference attacks have raised serious privacy threats by determining if a record was in the training dataset of a neural network model via querying that model. Given a target neural network model $f : \mathbb{R}^n \to \mathbb{R}$, the process of MIA can be formally defined as:

$$\mathcal{A} : \boldsymbol{x}, f \to \{0, 1\}, \tag{1}$$

where $\mathcal{A}$ denotes the attack model, which is a binary classifier. If the data sample $\boldsymbol{x}$ is used to train the target model f , the attack model $\mathcal{A}$ outputs 1 (*i.e*., member), and 0 otherwise (*i.e*., non-member).

Due to the practical consideration, most MIAs focused on the black-box setting, where an adversary only has access to the target model's outputs. By leveraging the target model's prediction confidences, Shokri *et al*., [22] proposed a black-box MIA. They constructed several shadow models to mimic the behavior of a target model. The well-established shadow models will then be used to generate data to train a neural network-based binary classifier to determine the membership of a record against the target model, *i.e*., whether a record belongs to the target model's training dataset or not. Salem *et al*., [23] further boosted this attack successfully by only using a single shadow model. To further improve the attack accuracy, Nasr *et al*., [24] included more features, such as the class labels of data samples, to train the binary classifier. In addition to the aforementioned neural network-based binary classifier, Leino *et al*., [25], Yeom *et al*., [26], and Song *et al*., [27,28] proposed the metric-based binary classifier, where the membership of a record is directly determined by a predefined threshold based on the metrics, such as the prediction confidences, entropy, or modified entropy of the record. Song and Mittal showed that by setting a class-dependent threshold, the metric-based classifier could achieve comparable or even better accurate inference performance compared with the neural network-based classifier [28]. Despite the extensive research on MIAs, none of them is designed towards pruned models. Therefore, we propose SAMIA to investigate the privacy risks of pruned models.

2.3 Defenses against MIAs

Recent efforts have been made to defend against MIAs. As one of the most popular privacy-preserving techniques, differential privacy (DP) provides provable defense against MIAs by adding noise to the gradient or parameter during model training [29–31]. However, DP usually requires a large magnitude of noises to achieve a meaningful privacy guarantee, which seriously degrades the performance of the protected models [32]. On the other hand, regularization [10], dropout, and model stacking [23] have been used in model training to reduce the privacy risks caused by overfitting. Although these approaches reduced the vulnerability by bridging the generalization gap between member and non-member data samples, in many cases, the privacy risks after applying these approaches are still high. Recent adversarial learning techniques [33, 34] have been introduced in defending against MIAs by adding noises to the prediction confidences for misleading the adversary [24, 35]. In a recent analysis of the defense mechanisms, Song and Mittal showed that the early stopping mechanism achieved comparable performance with most defenses [28]. In this paper, we provide a comprehensive analysis of defenses in neural network pruning, including our proposed PPB defense along with the existing defense mechanisms.

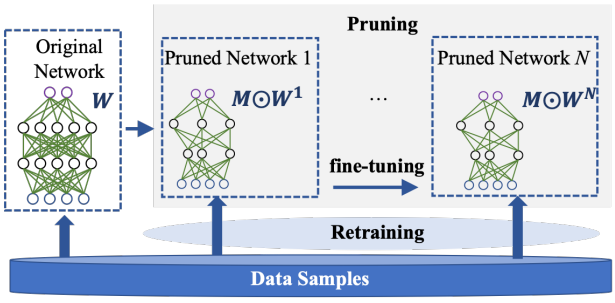


Figure 2: A typical workflow of neural network pruning.

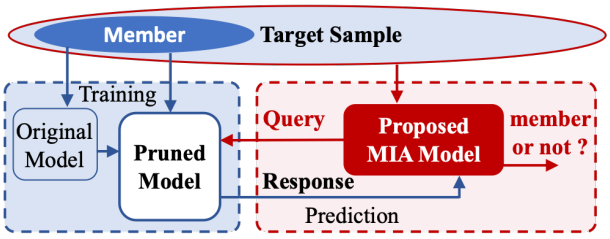


Figure 3: Framework of membership inference attacks (MIA) against neural network pruning.

3 System Overview

3.1 Neural Network Pruning Workflow

This paper is focused on a general neural network pruning process, whose workflow includes three key stages: original network training, coarse pruning, and fine-tuning, as illustrated in Figure 2. Specifically,

1. *Original network training*: A large size original neural network model $f(\boldsymbol{x};\boldsymbol{W})$ (sometime over-parameterized) is first trained at this stage, where $\boldsymbol{x}$ is the training data and $\boldsymbol{W}$ is the model parameters;
2. *Pruning*: Upon the original network, the pruning is conducted by removing insignificant parameters or groups of parameters according to a specific criterion. The pruned network can be given by $f(\boldsymbol{x};\boldsymbol{M}\odot\boldsymbol{W})$, where $\boldsymbol{M}\in 0,1^{|\boldsymbol{W}|}$ denotes the binary mask that can set a parameter to be 0, and $\odot$ denotes the element-wise multiplication;
3. *Fine-tuning*: To recover the performance loss due to pruning, a pruned network can be fine-tuned by reusing the training data. After N-epoch fine-tuning, a pruned network can be given by $f(\boldsymbol{x};\boldsymbol{M}\odot\boldsymbol{W}^N)$.

For the sake of simplicity, we use f to denote the original model $f(\boldsymbol{x};\boldsymbol{W})$ and f_p to denote the pruned model $f(\boldsymbol{x};M\odot\boldsymbol{W}^N)$ in the following paper.

3.2 Adversarial Knowledge

The goal of MIAs is to find the membership of a data sample, *i.e.*, whether the sample is used to train a target model or not. In this paper, we assume the adversary of the MIAs against a pruned neural network has the following knowledge.

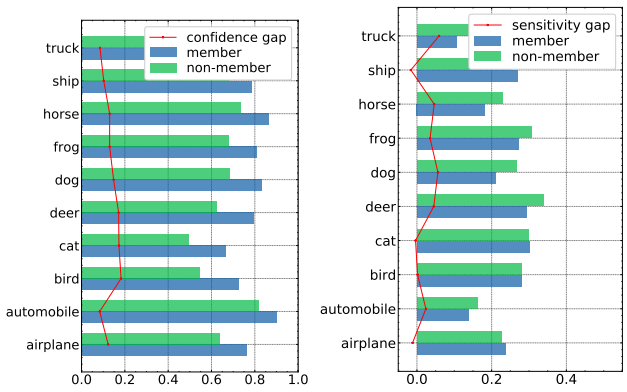
- *Access to query the pruned network*. The pruned model is made available to the public, *i.e.*, queryable. Due to practical considerations, the original model is assumed not published and inaccessible.
- *Access to the prediction confidences*. We consider the practical black-box MIAs [6]. The adversary can only acquire the output, *i.e.*, the prediction confidences, of the pruned network. Any internal information about the pruned model and the original model, such as the network architecture and activation functions, are inaccessible to the adversary.
- *Access to the pruning approach and the sparsity level*. We consider two different types of adversaries with or without knowledge of the pruning approach and the sparsity level.
- *Access to the defense approach*. The arms race between attacks and defenses is one main challenge in machine learning privacy. If the defense mechanisms are designed without considering the adversary's knowledge, their performance might be substantially degraded when adaptive attacks are used against those defensive mechanisms [28, 35]. Hence, we consider both non-adaptive and adaptive attacks to evaluate defense mechanisms: 1) non-adaptive attacks, *i.e.*, the adversary has no access to the defense mechanisms; 2) adaptive attacks, where the adversary has full knowledge of the defense mechanisms and performs the MIAs by taking the defensive mechanisms into account.

4 MIA against Neural Network Pruning

Given the workflow of neural network pruning presented in Section 3.1, this section focuses on investigating the privacy risks introduced by the pruning process. A general framework of MIAs against the pruned model is illustrated in Figure 3. Specifically, to extract the membership information from the pruned model, the adversary first derives the predictions of the given input sample by querying the target pruned model. The adversary then feeds the predictions into the trained attack model and provides the binary classification of the membership status. The attack model is derived following the shadow-training technique, which was originally proposed by Shokri *et al.* [22] and is widely used in MIAs [23, 24]: a shadow model is trained and pruned to imitate the behavior of the target pruned model. The adversary trains an attack model based on the pruned shadow model's predictions over shadow training and test data.

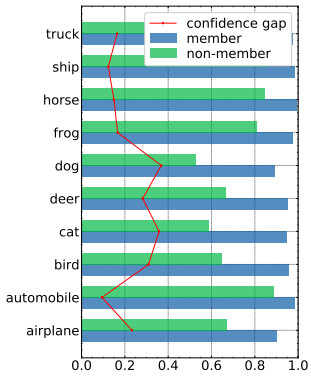
4.1 Divergence of Prediction Behaviors

To investigate the prediction behaviors of pruned neural networks, we first introduce two metrics: prediction confidence

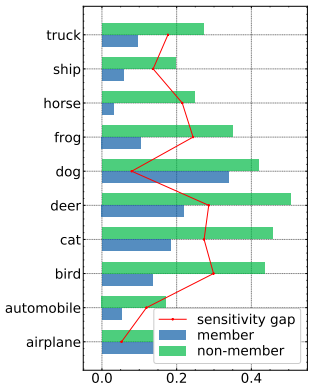


(a) Original Model Confidence

(b) Original Model Sensitivity



(c) Pruned Model Confidence



(d) Pruned Model Sensitivity

Figure 4: Divergence of the pruned model's prediction confidences and prediction sensitivities over different classes, respectively. We prune DenseNet121 models with 70% sparsity on the CIFAR10 datasets. The blue bar indicates the average prediction confidence/sensitivity of members in different classes. The green bar indicates the average prediction confidence/sensitivity of non-members. The divergence of confidence (a) and sensitivity (b) between members and non-members is increased after pruning as shown in (c) and (d), respectively. Such divergence also differs among classes.

and prediction sensitivity. Specifically, given an input sample $\boldsymbol{x}$ and a pruned model f_p, the prediction confidence is defined as $\text{PC} = f_p(\boldsymbol{x})$. In addition, to further measure the prediction behavior changes in terms of slight input change, we introduce prediction sensitivity, which is defined as

$$\text{PS} = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_p(\boldsymbol{x}+\epsilon\boldsymbol{\delta_i}) - f_p(\boldsymbol{x})|}{\epsilon}, \tag{2}$$

where $\boldsymbol{\delta_i} \sim \mathcal{N}(0,1)$ is a random Gaussian noise vector added to the input data $\boldsymbol{x}$, and ϵ controls the magnitude of input changes. A similar idea has been used in the gradient estimation for black-box adversarial attacks [36]. It has been shown that a small number of noise vectors can achieve a good estimation of prediction changes, so that we set a small query budget in the evaluation ($n = 10$) [36, 37]. Accordingly, we use the confidence and sensitivity to measure the divergence between members and non-members. We define the confidence gap as

$$\frac{1}{|\mathcal{D}_{train}|}\sum_{(\boldsymbol{x_i},y_i)\in\mathcal{D}_{train}} f_p^{y_i}(\boldsymbol{x_i}) - \frac{1}{|\mathcal{D}_{test}|}\sum_{(\boldsymbol{x_i},y_i)\in\mathcal{D}_{test}} f_p^{y_i}(\boldsymbol{x_i}), \tag{3}$$

where $f_p^{y_i}$ denotes the prediction confidence of ground-truth class y_i. Confidence gap calculates the difference of average confidence between members and non-members in the ground-truth class. Similarly, we define the sensitivity gap as

$$\frac{1}{|\mathcal{D}_{train}|}\sum_{(\boldsymbol{x_i},y_i)\in\mathcal{D}_{train}} \text{PS}^{y_i}(\boldsymbol{x_i}) - \frac{1}{|\mathcal{D}_{test}|}\sum_{(\boldsymbol{x_i},y_i)\in\mathcal{D}_{test}} \text{PS}^{y_i}(\boldsymbol{x_i}), \tag{4}$$

where PS^{y_i} denotes the prediction sensitivity (Eq. 2) of ground-truth class y_i. The sensitivity gap calculates the difference of average sensitivity between members and non-members in the ground-truth class.

As illustrated in Figure 1, the divergence of prediction confidences and prediction sensitivities is increased due to neural network pruning, which introduces the new attack vectors for MIAs and thus makes the pruned models more vulnerable. Moreover, the divergences of prediction confidences and sensitivities from the pruned model vary widely among the different classes of training and test data. Figure 4 shows that the divergences of the pruned models' prediction behavior (confidence and sensitivity) over members and non-members are significantly different among classes. Similar observations of prediction confidences on different classes after pruning have been made in other fields such as model fairness and transparency [20, 21, 38].

4.2 SAMIA: Self-Attention MIA

Upon the above observations, we propose one hypothesis: *the divergences among classes, i.e., confidence gap and sensitivity gap, can provide fine-grained "evidence" for MIAs, leading to serious privacy leakage*. In addition, most existing MIA research only considers the confidence gap and a single threshold of the ground-truth class, which may underestimate the privacy risks of MIAs in neural network pruning. Hence, we propose SAMIA, a self-attention MIA, to fully utilize the increased divergence information along with the class information to conduct a finer-grained analysis. Specifically, self-attention is a neural network module to capture global dependencies among inputs and allows the inputs to interact with each other. Despite the recent success of self-attention mechanism in many areas, such as natural language processing [39, 40] and computer vision [41–43], it has not been well exploited in the research of privacy attacks yet.

In SAMIA, we leverage the self-attention mechanism to automatically extract the finer-grained "thresholds" from different classes by capturing the dependency between predicted information (confidence and sensitivity) and class information

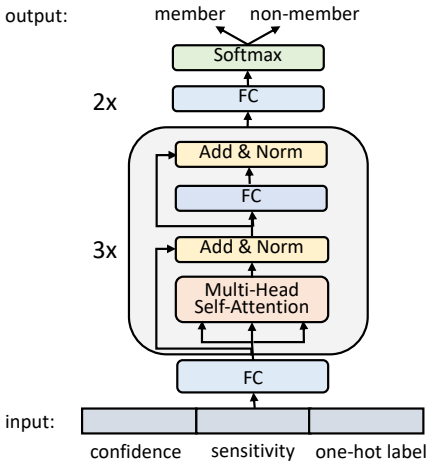


Figure 5: Attack model architecture in SAMIA.

and allowing them to interact with each other. Specifically, SAMIA takes the pruned model's prediction confidence and sensitivity and ground-truth labels as inputs. Given a specific class, the self-attention mechanism finds out the specific confidence information and sensitivity information that the attack "threshold" should pay more "attention" to.

Figure 5 illustrates the network architecture of the attack model used in SAMIA, enlightened by the idea of Transformer [39], *i.e.*, one of the most widely used self-attention architectures. We first convert the ground-truth label into a one-hot vector and then feed both pruned model's prediction confidence, sensitivity, and the one-hot vector into the attack model as the input features. The input features are encoded into a vector using a Fully Connected (FC) layer, which is then fed into the multi-head self-attention modules. In each module, we encode the features as query Q, key K, and value V vectors using a linear function following the self-attention strategy. The attention module calculates the attention scores of the subgroups in a scaled dot-product way: $\text{Attention}(Q,K,V) = \text{softmax}(QK^T)V$, where softmax() denotes the softmax function to make the attention scores sum up to 1. The output of the attention module is the weighted sum of the value vector, where the weight assigned to each value is derived by the attention scores $\text{softmax}(QK^T)$. In addition, we calculate four attention scores (*i.e.*, multi-head attention) to capture the different attention strategies. Followed by the attention module, we add the result to the input features and apply the layer normalization [44] to stabilize the attack model training. The result will be fed into another FC layer with layer normalization. We consider these operations as a block and repeat the block three times, followed by two fully connected layers. A non-linear activation function, ReLU is applied to the output of the first few FC layers. A softmax function is applied to the last FC layer to provide the binary prediction on the membership status.

Compared with existing MIAs that learn a single threshold of prediction confidence to determine the membership, the proposed SAMIA captures the information of confidences and sensitivities and intuitively better learns the diverse thresholds to multiple classes. Our evaluation results demonstrate that SAMIA leads to higher attack accuracy compared with the state-of-the-art attacks.

5 Attack Evaluation

This section conducts comprehensive experiments[2] to thoroughly investigate the privacy risks of the proposed MIAs against neural network pruning. In the following, we first introduce the experimental setup, and then evaluate the privacy risks of the pruned models by comparing them with those of original models. Next, we investigate the impact of the confidence gap, sensitivity gap, and generalization gap, respectively. Finally, we evaluate the privacy risks without the knowledge of pruning approaches and sparsity levels.

5.1 Evaluation Setup

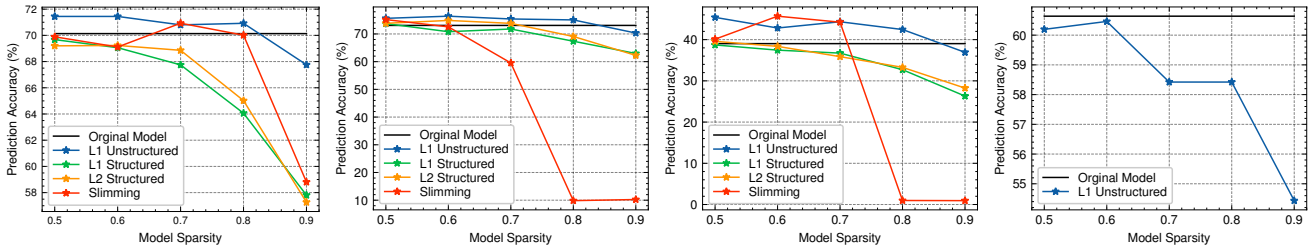
In the evaluation, we consider the most widely used datasets, neural network architectures, and optimization approaches following recent research of MIAs [10, 23, 28, 45].

5.1.1 Datasets

We consider seven popular datasets in the experiments: CIFAR10, CIFAR100, CHMNIST, SVHN, Texas, Location, and Purchase.

- *CIFAR10* and *CIFAR100* [46]. These are two benchmark datasets for image classification. CIFAR10 dataset contains 60,000 32×32 color images in 10 classes, with 6,000 images per class. CIFAR100 dataset contains 60,000 color images in 100 classes, with 600 images per class.
- *CHMNIST* [47]. This dataset consists of 5,000 histological images of human colorectal cancer containing 10 classes of tissues. We resize all images to 32×32, the same dimension as CIFAR10 and CIFAR100.
- *SVHN* [48]. This dataset consists of 99,289 32×32 color images from house numbers in the Google Street View dataset, containing 10 classes from 0 to 9.
- *Location* [49, 50]. This dataset contains location "check-in" records of mobile users in the Foursquare social network, restricted to the Bangkok area. The dataset is used to predict users' geosocial type based on the geographical history record features: whether the user visited a certain region or location type. We use the preprocessed purchase dataset provided by Shokri *et al.* [10], which contains 5,010 data samples, 446 binary features, and 30 classes.

[2] Due to the space limit, we only present the major results in this paper. More details can be found in the extended version https://arxiv.org/abs/2202.03335.



(a) CIFAR10, ResNet18 (b) CIFAR10, DenseNet121 (c) CIFAR100, DenseNet121 (d) Location, FC

Figure 6: Prediction accuracy (test accuracy) of the pruned models using different pruning approaches and sparsity levels. Each point indicates the prediction accuracy achieved by the pruned model with a certain pruning approach and sparsity level. The black line indicates the prediction accuracy of the original models.

- *Texas* [51]. This dataset is presented in the Hospital Discharge Data Public Use Data File provided by the Texas Department of State Health Services. The dataset is used to predict the types of patient's main procedure based on a wide range of features, such as external causes of injury, diagnosis of the patient, procedures the patient underwent, and other generic information. We use the preprocessed purchase dataset provided by [10], which contains 67,330 data samples, 6,169 binary features, and 100 classes.
- *Purchase* [52]. This dataset is presented in Acquire Valued Shoppers Challenge to predict which shoppers will become repeat buyers based on the purchase history. We use the preprocessed purchase dataset provided by Shokri *et al.* [10], which contains 197,324 data samples, 600 binary features, and 100 classes.

Each above dataset is first randomly and equally split into two parts: one for target model, one for shadow model. In each part, we split the data into three datasets: training (45%), validation (10%), and test (45%). We use the validation dataset to determine if the model needs to stop training or fine-tuning for early stopping. Therefore, the membership inference via random guessing results in 50% attack accuracy. Due to the space limit, we only show the results of the CIFAR10 and Purchase datasets. The rest results are presented in the Appendix.

5.1.2 Neural Network Architectures

For the four image datasets, *i.e.*, CIFAR10, CIFAR100, CHMNIST, and SVHN, we consider three representative neural network architectures: ResNet18, VGG16, and DenseNet121[3]. For the other three datasets, *i.e.*, Texas, Purchase, and Location, we implement fully connected (FC) neural networks with two layers, and the numbers of neurons for each layer are 256 and 128, respectively. All the FC layers except the last one are followed by ReLU activation functions. In addition, Adam optimizer [53] is implemented with a learning rate of 0.001 and the batch size of 128 to train all the original models and fine-tune all the models after pruning.

[3]All neural networks are trained using https://github.com/huyvnphan/PyTorch_CIFAR10

5.1.3 Neural Network Pruning Approaches

Four representative neural network pruning approaches are considered, including L1 unstructured pruning, L1 structured pruning, L2 structured pruning, and Network slimming.

- *L1 unstructured pruning* [2] (L1 unstructured), which removes the weights with the lowest absolute values individually. This pruning approach can produce a sparse neural network with a small size, but may not improve efficiency given the existing hardware and software optimization.
- *L1 structured pruning* [3] (L1 structured), which removes the entire filters with the lowest absolute values from the convolution layers. By removing the entire filters, this method leads to significant speedup compared with the unstructured pruning since optimization for dense matrix can be applied for efficient computation.
- *L2 structured pruning* (L2 structured), which removes the entire filters with the lowest L2 norm values from the convolution layers, similar to L1 structured pruning.
- *Network slimming* [4] (Slimming), which associates scaling factors used in the batch normalization layer with each channel and removes the entire channels with the lowest scaling factors. This method automatically identifies the insignificant channels and finds the target architectures.

We apply the L1 unstructured pruning to all models. Since structured pruning approaches, *i.e.*, L1 structured and L2 structured pruning and Slimming, can only be applied to pruning convolution layers, we evaluate the structured pruning approaches on the ResNet18, VGG16, and DenseNet121 models trained on CIFAR10, CIFAR100, SVHN datasets. In addition, five sparsity levels $\gamma = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ are investigated for all pruning approaches, which denote the portions of the

removed parameters[4]. We follow typical pruning procedures: train the original model, prune the model using the above approaches, and finally fine-tune the pruned model.

Figure 6 shows the prediction accuracy of the original model and the pruned models with different pruning approaches and sparsity levels. We observe that the pruned models achieve close performance compared to the original model if the sparsity level is not high. The accuracy of pruned models is reduced with the increase of the pruning sparsity. Sometimes, pruned models can achieve higher accuracy than the original models, which has been shown in recent studies of neural network pruning [5]. Unstructured pruning usually performs better than structured pruning in the evaluation, since structured pruning forces the removed parameter in a restricted way, which limits the performance of the pruned model but increases the speed of model inference.

5.1.4 State-of-the-art MIAs

To thoroughly evaluate the proposed SAMIA, we investigate eight state-of-the-art MIAs along with SAMIA. [5].

- *Ground-truth class confidence-based threshold attack (Conf).* Yeom *et al.* used the prediction confidence of ground-truth class to identify membership status [26]. The adversary learns a threshold to determine the membership of a data sample based on the confidence of ground-truth class. Given an input sample $\boldsymbol{x}$, its class y, and the pruned model f_p, the attack function is defined as $I_{\text{conf}}(f_p,(\boldsymbol{x},y)) = \mathbb{1}\{f_p^{(y)}(\boldsymbol{x}) \geq \zeta_y\}$, where $f_p^{(y)}$ is the prediction confidence of class y and ζ_y is the threshold of class y derived from the shadow pruned model.
- *Cross-Entropy-based threshold attack (Xent).* The entropy loss can be used to derive the threshold from the shadow pruned model [26]. The attack function is defined as $I_{\text{xent}}(f_p,(\boldsymbol{x},y)) = \mathbb{1}\{\text{xent}(f_p^{(y)}(\boldsymbol{x})) \geq \zeta_y\}$, where xent denotes the cross entropy loss.
- *Modified-entropy-based threshold attack (Mentr).* Song and Mittal proposed modified entropy by including the information about the ground-truth class, which achieved better performance than using prediction confidence [28]. The attack function is defined as $I_{\text{mentr}}(f_p,(\boldsymbol{x},y)) = \mathbb{1}\{\text{mentr}(f_p^{(y)}(\boldsymbol{x})) \geq \zeta_y\}$, where $\text{mentr}(f_p(\boldsymbol{x}),y) = -(1-f_p^{(y)}(\boldsymbol{x}))\log(f_p^{(y)}(\boldsymbol{x})) - \sum_{t\neq y} f_p^{(t)}(\boldsymbol{x})\log(1-f_p^{(t)}(\boldsymbol{x}))$.
- *Top1 Confidence-based threshold attack (Top1-Conf).* Salem *et al.* proposed to derive the threshold from the highest prediction confidence [23]. The attack function is defined as $I_{\text{top1}}(f_p,(\boldsymbol{x})) = \mathbb{1}\{\text{top1}(f_p(\boldsymbol{x})) \geq \zeta_y\}$, where top1 calculates the highest value from the prediction confidence.
- *Confidence-based Neural Network attack (NN).* Shokri *et al.* proposed to use prediction confidence as features to train a neural network from the shadow model [10], which is used to distinguish member and non-member data.
- *Top-3 Confidence-based Neural Network attack (Top3-NN).* Salem *et al.* proposed to use the top-3 prediction confidences as features [23] to train a neural network classifier.
- *Confidence-based Neural Network attack with ground-truth class (NNCls).* Nasr *et al.* combined one-hot encoded class labels with the prediction confidence as features to train a neural network classifier [24].
- *Blind Membership Inference Attack (BlindMI).* Hui *et al.* proposed to determine the membership of a data sample by moving it to a non-member set and check if the moving operation increases the distance between member and non-member sets [45]. BlindMI considers the data sample as a non-member if the distance is increased. We use the default BlindMI attack provided in [45].

In the main paper, we present the results of five attacks, that achieve the highest attack accuracies in most experiments, *i.e.*, Conf, Mentr, NNCls, BlindMI, and SAMIA. The results of the rest of the attacks are reported in the Appendix.

Besides, it should be mentioned that to provide a practical analysis of privacy risks, we adopt early stopping and l2 regularization as a baseline defense mechanism and apply it to all the following experiments of membership inference attacks. Other defense mechanisms will be discussed in Section 6.

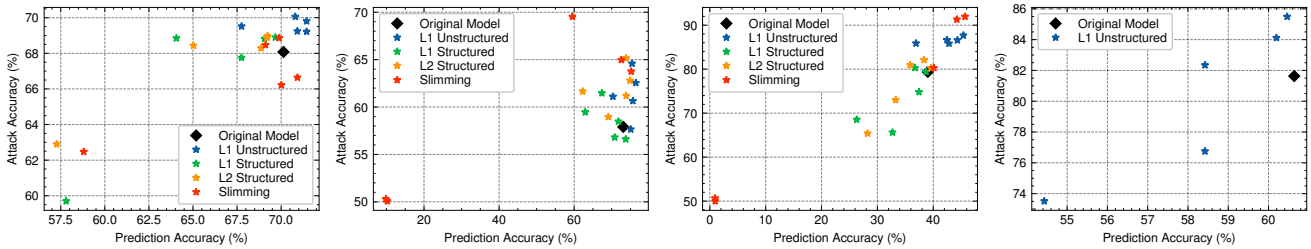
5.1.5 SAMIA Settings

Following the experimental setting in [10], we first train five shadow models and their pruned models. The predictions of the shadow models on shadow training and shadow test datasets are used to train an attack model. In the attack model, we use four attention heads, 64 neural units, and GeLU activation function [54] in each self-attention module, with a 20% dropout rate. We use SGD optimizer [55] to train the attack models for 100 epochs with batch size 128. The learning rate of the SGD optimizer is set as 0.01 and reduced to 0.001 and 0.0001 at the 1/2 and 3/4 of the training process (*i.e.*, the 50th and 75th epoch). Due to the large number of settings evaluated in the attacks and defenses and the high computational cost in each setting, we conduct all the experiments only once. Thus experimental variation may be observed due to the randomness in neural network pruning and membership inference attacks (*e.g.*, parameter initialization, dataset shuffling).

5.2 Privacy Risk Discussions

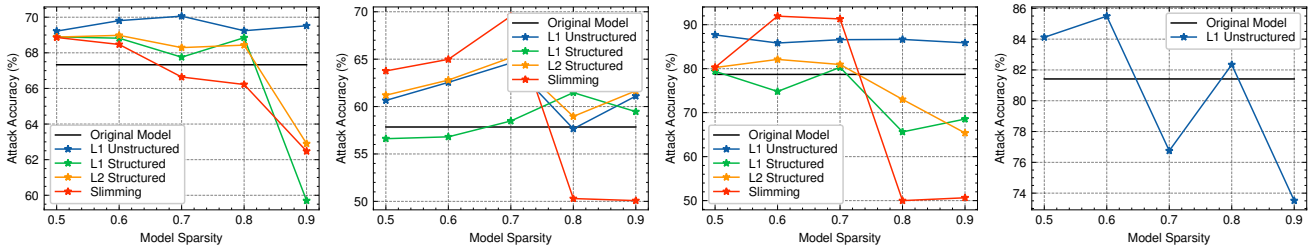
[4] Since structured pruning only removes the parameters in the convolution layers, the sparsity levels for structured pruning only count the removed parameters in the convolution layers instead of the entire neural network.

[5] We implement Conf, Xent, Mentr, and Top1-Conf attacks based on https://github.com/inspire-group/membership-inference-evaluation and BlindMI attack based on https://github.com/hyhmia/BlindMI/blob/master/BlindMI_Diff_W.py.



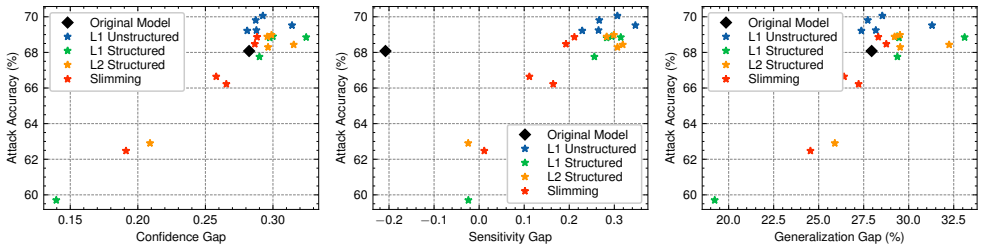
(a) CIFAR10, ResNet18 (b) CIFAR10, DenseNet121 (c) CIFAR100, DenseNet121 (d) Location, FC

Figure 7: Privacy Risks of Neural Network Pruning (w.r.t. prediction accuracy). Most pruning approaches result in a higher attack accuracy when considering a similar prediction accuracy, compared with the original models. We present the attack accuracy of SAMIA for pruned models and the attack accuracy of Conf attack for the original models.



(a) CIFAR10, ResNet18 (b) CIFAR10, DenseNet121 (c) CIFAR100, DenseNet121 (d) Location, FC

Figure 8: Privacy Risks of Neural Network Pruning (w.r.t. model sparsity). We present the attack accuracy of SAMIA for pruned models and the attack accuracy of Conf attack for the original models.



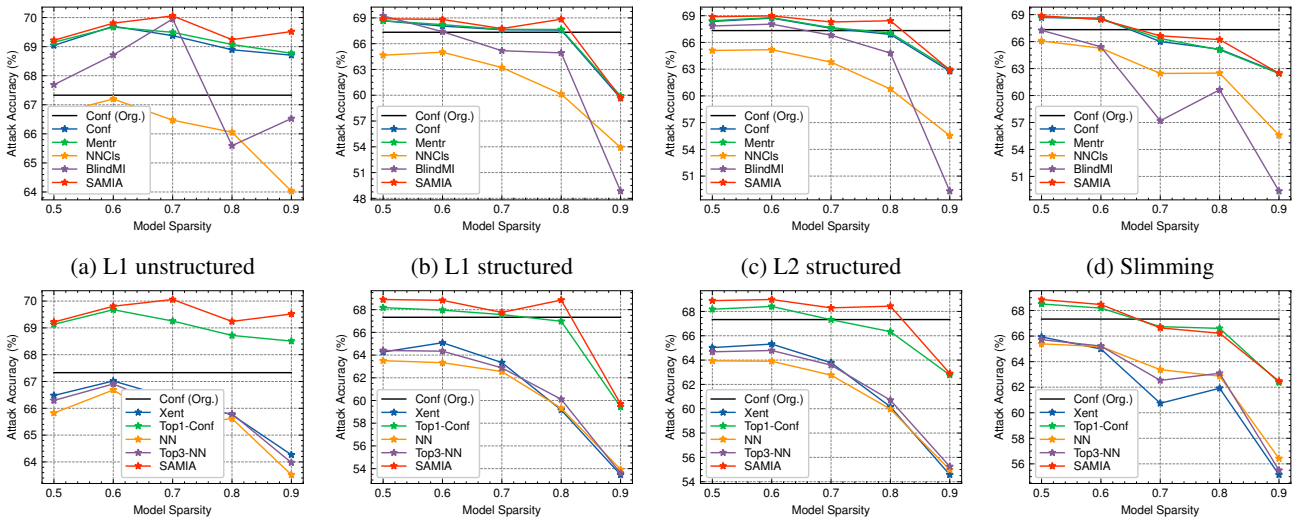
(a) Confidence gap (b) Sensitivity gap (c) Generalization gap

Figure 9: Impact of confidence gap, sensitivity gap, and generalization gap (CIFAR10, ResNet18). We present the relationship between the gap and the attack accuracy of SAMIA.

In this section, we evaluate the privacy risks of the pruned models and compare them with the original models and then investigate several key factors on privacy risks of neural network pruning. Additionally, we investigate the privacy risks of different pruning approaches and discuss the effectiveness of the proposed SAMIA and the impact of unknown sparsity levels and pruning approaches.

5.2.1 Privacy Risks of Neural Network Pruning

Since different pruning approaches and sparsity levels may achieve distinct prediction accuracy, to make a fair comparison, we evaluate the privacy risks of pruning by taking the prediction accuracy into consideration. Figure 7 shows the relationship between prediction accuracy and (SAMIA) attack accuracy when we apply different pruning approaches and sparsity levels on the CIFAR10, CIFAR100, and Location datasets. We observe that *when the pruned model achieves a comparable prediction accuracy with the original model, most pruning approaches result in an increased attack accuracy (i.e., privacy risk).* The attack accuracy may be decreased with the loss of prediction accuracy, as the pruned model becomes less effective for both prediction and attack. However, we still observe that in most cases, when the pruned model performs worse than the original model, the pruned model's attack accuracy remains higher than the original one's. Therefore,



(a) L1 unstructured (b) L1 structured (c) L2 structured (d) Slimming

(e) L1 unstructured (f) L1 structured (g) L2 structured (h) Slimming

Figure 10: Attack performance comparison of MIAs (CIFAR10, ResNet18). We present the attack accuracy of state-of-the-art membership inference attacks and compare them with the proposed SAMIA. Four pruning approaches are used on CIFAR10 ResNet18 models. The black line presents the attack accuracy of original models using Conf attack, *i.e.*, Conf (Org.).

the pruned models become more vulnerable to membership inference attacks than the original models.

When a low sparsity level is used, we always observe the increased privacy risk of the pruned model (Figure 8). Since with a low sparsity level, the pruned model is more likely to achieve a comparable or even higher prediction accuracy compared with the original model, which increases the accuracy of prediction confidence used in MIAs and further increases the privacy risk.

5.2.2 Impact of Confidence, Sensitivity, and Generalization Gap.

As aforementioned, we hypothesize that neural network pruning leads to the increased *confidence gap* and *sensitivity gap* (in ground-truth class) of pruned models, thus increasing their membership inference risks. Meanwhile, overtraining is considered as one of the key causes of membership leakage in previous research [26, 28], leading to our evaluation on *generalization gap*, *i.e.*, the difference between training accuracy and testing accuracy. From Figure 9, we observe that neural network pruning increases the gaps between members and non-members, *i.e.*, confidence gap, sensitivity gap, and generalization gap, in most settings. Further, with the increase of gaps, we observe the increase of attack accuracy, which indicates *the strong correlation between the gaps, i.e., confidence gap, sensitivity gap, and generalization gap, and the increased privacy risk.*

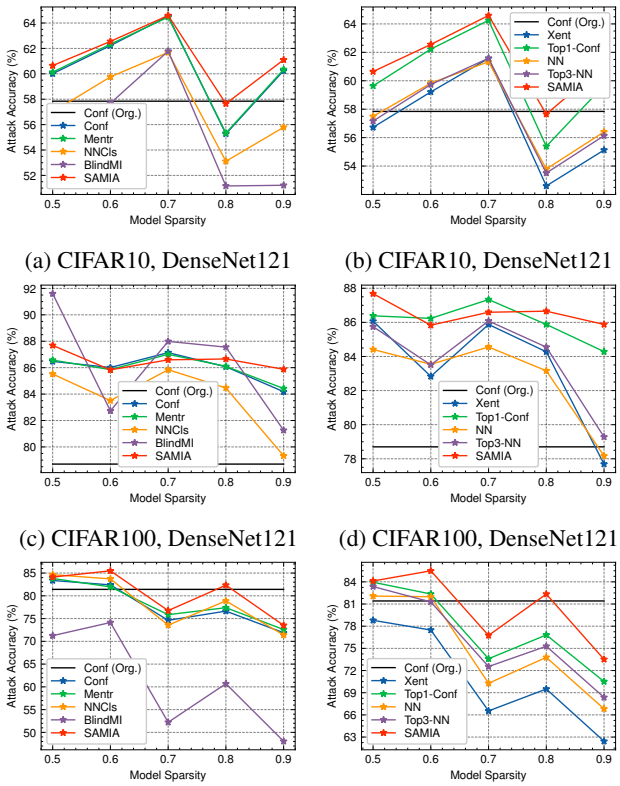
The strong correlation of confidence gap and sensitivity gap validates our intuition that these gaps can be leveraged by the adversary to infer the membership status, introducing a new attack surface in neural network pruning. By investigating the attack results, we find that the confidence gap plays the most important role in the privacy risk. L1 unstructured and slimming pruning usually lead to an increased confidence gap, which will be leveraged by the adversary and result in a higher attack accuracy. Additionally, sensitivity gap can also leak the membership information. For example, the confidence gap of L1 unstructured pruning on a CIFAR10 ResNet18 model (Figure 9a) is close to the original model, but due to the increased sensitivity gap (Figure 9b), the pruning still results in an increased attack accuracy.

5.2.3 Privacy Risks of Pruning Approaches.

Following the same settings above, we investigate the privacy risks of different pruning approaches by comparing the attack accuracy under the similar prediction accuracy. As shown in Figure 7 and 8, given the similar prediction accuracy of the pruned models, L1 unstructured and slimming pruning result in the highest attack accuracy. Besides, L1 structured pruning achieves the lowest attack accuracy among all pruning approaches, but still in some cases, the attack accuracy is higher than the original model, even with the similar or lower prediction accuracy. The structured constraint used in L1 structured pruning regularizes the model in the fine-tuning and thereby reduces the privacy risk.

5.2.4 Effectiveness of SAMIA

To investigate the effectiveness of the proposed SAMIA, we compare SAMIA with the state-of-the-art MIAs in terms of



(a) CIFAR10, DenseNet121 (b) CIFAR10, DenseNet121

(c) CIFAR100, DenseNet121 (d) CIFAR100, DenseNet121

(e) Location, FC (f) Location, FC

Figure 11: Attack performance comparison of MIAs on different datasets (L1 Unstructured). We present the attack accuracy of state-of-the-art membership inference attacks and compared them with the proposed SAMIA. We present the attack accuracy of three models (CIFAR10 DenseNet121, CIFAR100 DenseNet121, Location FC) pruned by L1 unstructured pruning. The black line presents the attack accuracy of original models using Conf attack, *i.e.*, Conf (Org.).

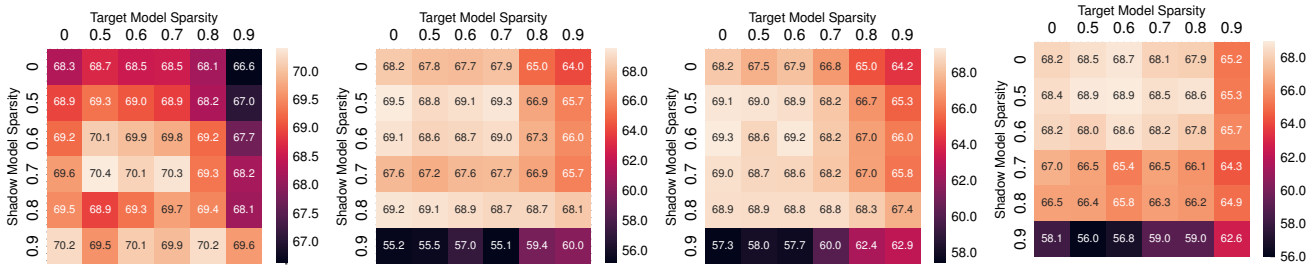
attack accuracy. As shown in Figure 10, we observe that *our proposed SAMIA achieves the highest attack accuracy in most cases compared with baseline attacks*, which is mainly due to the fact that SAMIA best leverages both confidence gap and sensitivity gap (in ground-truth class) introduced in pruning. Besides, Top1-conf and Mentr attacks are also effective, as both attacks take advantage of the confidence gap, the most important factor for model privacy. We also observe that when the pruning introduces a high generalization gap , all attacks can achieve a high attack accuracy (*e.g.*, CIFAR100 DenseNet121 in Figure 11c, 11d and Appendix Figure 36c)), which has been discussed in the previous MIA research.

5.2.5 Unknown Sparsity Level and Pruning Approach

In the evaluation above, we assume the adversary has the knowledge of sparsity levels and pruning approaches used in network pruning. In this section, we explore the privacy risks of a more realistic scenario, *i.e.*, when the adversary has no prior knowledge of the sparsity levels and the pruning approaches.

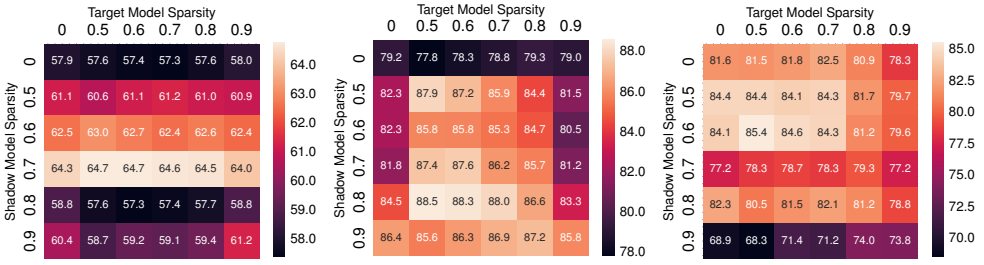
Unknown sparsity level. We assume the adversary only knows the pruning approach but not the sparsity level that is the major factor of model efficiency. We evaluate the attack accuracy of SAMIA when the adversary prunes target models and shadow models using different sparsity levels. We also consider the case when the target model is not pruned, *i.e.*, sparsity level = 0. As shown in Figure 12 and 13, the attack accuracy is not affected too much due to the different sparsity levels between target models and shadow models. In some cases, using a different sparsity level in pruning shadow models can even increase the attack accuracy. *The attack accuracy mainly depends on the performance of the shadow model*, and thus the adversary can attack victim models with higher attack accuracy by selecting a good pruned shadow model. For instance, the adversary can use each shadow model to attack other shadow models with different sparsity levels and select the one with the highest attack accuracy.

Unknown sparsity level and pruning approach. Since we assume the adversary has no prior knowledge of the sparsity level and pruning approach, the adversary may randomly pick a sparsity level and a pruning approach to prune a shadow model for attacks. To evaluate the attack accuracy, we conduct 20 experiments for the aforementioned four image datasets and the corresponding neural networks. In each experiment, we randomly select the sparsity levels and pruning approaches for target models and shadow models, respectively. For example, the target model uses L1 Structured pruning with 0.5 sparsity level while the shadow model uses Slimming pruning with 0.8 sparsity level. The sparsity levels are selected from the set of $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ and the pruning approaches are selected from the four pruning approaches. To measure the privacy risks, we define the attack accuracy loss as $(acc_{known} - acc_{unknown})/acc_{known}$, where acc_{known} denotes the attack accuracy when the adversary knows all the pruning information and $acc_{unknown}$ denotes the attack accuracy without knowing any pruning information. Table 1 shows the average attack accuracy loss over 20 experiments for each dataset and model. We observe that *without knowing the sparsity levels and pruning approaches, the attack is still effective in most cases except the CIFAR10 VGG16 and CIFAR100 ResNet18 models*. The poor attack performance in these two models is due to the ineffectiveness of shadow models using specific sparsity levels and pruning approaches for attacks. For example, we observe a significant drop of attack accuracy (from 90% to 50%) when applying L1 structured and slimming pruning with sparsity levels 0.7 to 0.9, on the CIFAR10 VGG16 model (Figure 20a in Appendix). The large gap makes the shadow models pruned using these settings ineffective in attacking the unknown victim model.



(a) L1 Unstructured (b) L1 Structured (c) L2 Structured (d) Slimming

Figure 12: Attack accuracy with unknown sparsity levels (CIFAR10, ResNet18).



(a) CIFAR10, DenseNet121 (b) CIFAR100, DenseNet121 (c) Location, FC

Figure 13: Attack accuracy with unknown sparsity levels (L1 Unstructured).

Table 1: Attack accuracy loss with unknown sparsity levels and pruning approaches.

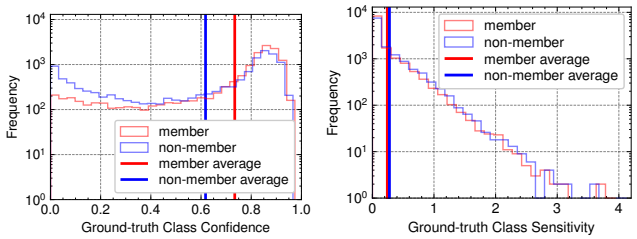
Dataset	Model	Attack Acc Loss
CIFAR10	ResNet18	4.77%
	DenseNet121	1.63%
	VGG16	26.83%
CIFAR100	ResNet18	12.41%
	DenseNet121	6.90%
	VGG16	2.43%
SVHN	ResNet18	0.60%
	DenseNet121	0.12%
	VGG16	0.05%
CHMNIST	ResNet18	0.78%
	DenseNet121	0.52%
	VGG16	-0.58%

6 Defenses against MIAs

Given the privacy risks of pruned neural networks, this section focuses on defenses against the proposed SAMIA. We first present the design principle of defenses for pruned neural networks, then describe the proposed defensive design, and lastly compare the performance of the proposed defense with the state-of-the-art defenses. In addition, to rigorously evaluate the defense performance, we consider the defenses against both the non-adaptive attacks and adaptive attacks, where the adversary of adaptive attacks is put into the last step of the arms race between privacy attacks and defenses (*i.e.*, the adversary knows all the details of defense mechanism and performs adaptive attacks against the defended models) [28, 56]. Extensive experiments are conducted to evaluate our defensive proposals.

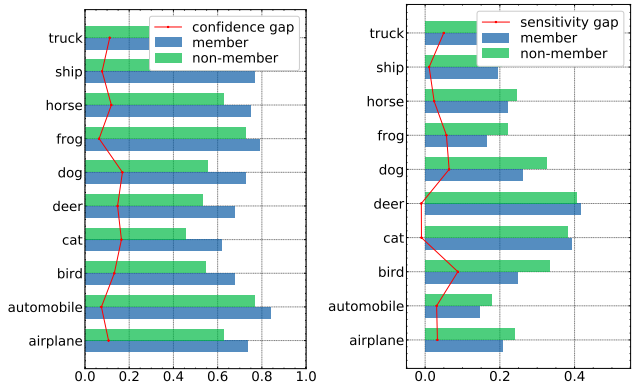
6.1 Design Principles of Defenses

Two major design principles are considered for the defenses of pruned neural networks. On the one hand, effective defenses should be able to reduce the behavior discrepancy introduced by pruning. The above attack evaluation has demonstrated that the privacy risks introduced by MIAs in the pruned models are due to the increased divergence of prediction confidences and sensitivities. Hence, it is essential to reduce such divergence between members and non-members of the pruned neural networks for defense. On the other hand, the defenses need to take into consideration the resource constraints imposed by low-end devices. Neural network pruning aims to reduce the computational cost during inference. Such cost cannot be increased by the defenses. Therefore, the defenses should be designed to mitigate the privacy risks of pruned models before deploying them on devices, thus without introducing additional defense costs in the inference phase.



(a) Confidence gap (b) Sensitivity gap

Figure 14: Divergence of the pruned model's prediction confidences and sensitivities using PPB defense (CIFAR10, DenseNet121).



(a) Confidence gap (b) Sensitivity gap

Figure 15: Divergence of the pruned model's prediction confidences and sensitivities over different classes with PPB defense (CIFAR10, DenseNet121).

6.2 Proposed Defense: PPB

Following the two design principles, we propose a countermeasure approach named by pair-based posterior balancing (PPB). The main idea of PPB defense is to mitigate the new prediction behaviors on prediction confidence and sensitivity by aligning the posterior predictions of different input samples. In this way, PPB can reduce the divergence of prediction confidence between members and non-members as well as the degree of sensitivities. Specifically, given any pair of two input samples, we try to make the distributions of their ranked posterior predictions as close as possible. The difference between ranked posteriors' distributions is measured by the Kullback–Leibler divergence (KL divergence) [57]. Give two posterior predictions P and Q, the KL divergence is defined as:

$$\mathcal{L}_{\text{KL}}(P,Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}. \tag{5}$$

KL-divergence is considered as a regularization term in neural network pruning. The loss function includes both the prediction loss and KL divergence loss, which can be given by:

$$\mathcal{L}(f_p(\boldsymbol{x}), \boldsymbol{y}) = \sum_i \mathcal{L}_{\text{predict}}(f_p(\boldsymbol{x}_i), y_i) + \lambda \sum_{j,k(j\neq k)} \mathcal{L}_{\text{KL}}(R(f_p(\boldsymbol{x}_j)), R(f_p(\boldsymbol{x}_k))), \tag{6}$$

where $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{predict}}$ denote the KL-divergence loss and the prediction loss (*e.g.*, cross-entropy loss for the classification tasks), respectively. $R(\cdot)$ sorts the posteriors provided by the pruned model f_p in decreasing order and λ is a hyper-parameter to balance the two losses. It is computationally costly to calculate the KL loss for all possible pairs of data samples in the training dataset. To address this issue, we sample training pairs in each mini-batch during fine-tuning by randomly selecting two data samples as a pair without replacement. Hence, in each mini-batch with batch size B, KL loss consists of $B/2$ pairs of training samples.

In addition, the PPB defense is only applied in fine-tuning of neural network pruning by using KL-divergence as a regularization term. Thus, the defense does not include the additional computational costs in the inference phase.

After applying PPB defense, we observe the divergence between the member and non-member data is significantly reduced by comparing Figure 14 with Figure 1. Such decreased divergence can be observed in different classes by comparing Figure 4 with Figure 15. Both changes on the distributions of the pruned model's posterior predictions indicate that the PPB defense makes the attack model fail to learn the binary classification thresholds from the prediction confidence and sensitivity. Moreover, the PPB defense is designed to change the distribution of predictions instead of their orders. In other words, the PPB defense will not change the predicted classes of the pruned models during fine-tuning, which largely preserves the prediction accuracy of pruned models.

As shown in Figure 15, such decreased divergence can also be preserved in different classes. After applying PPB defense, the divergence of the pruned model is close to that of the original model (comparing Figure 4 with Figure 15), which indicates the effectiveness of the PPB defense.

6.3 Defense Evaluation

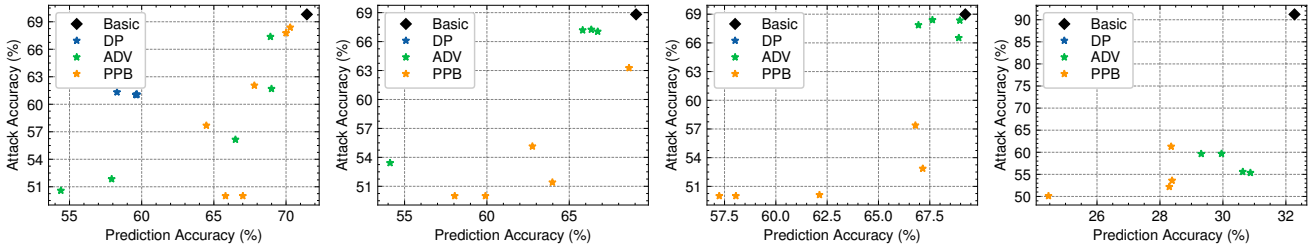
This section evaluates the effectiveness of PPB by comparing the performance of PPB with that of state-of-the-art defenses[6].

6.3.1 State-of-the-art Defenses

We investigate three state-of-the-art defenses against MIA attacks in neural network pruning.

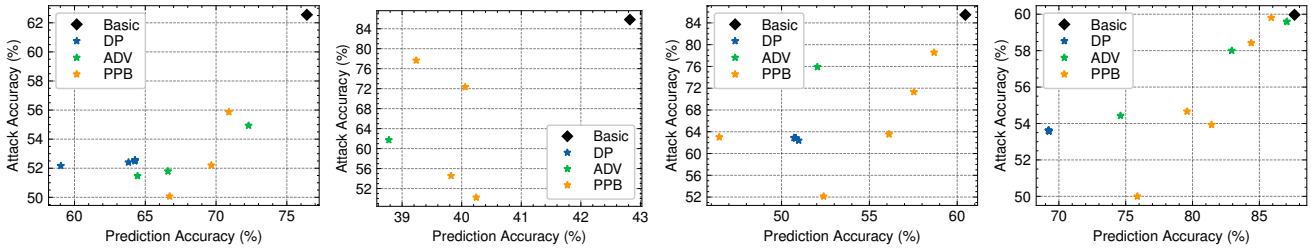
Early Stopping and L2 Regularization (Basic). Early stopping and l2 regularization have been used to successfully

[6]The defenses are evaluated by the empirical experiments. We will investigate the strict privacy guarantee in future work.



(a) L1 Unstructured (b) L1 Structured (c) L2 Structured (d) Slimming

Figure 16: Performance of defenses for different pruning approaches (CIFAR10, ResNet18, Sparsity 0.6).



(a) CIFAR10, DenseNet121 (b) CIFAR100, DenseNet121 (c) Location, FC (d) Purchase, FC

Figure 17: Performance of defenses for different datasets (L1 Unstructured, Sparsity 0.6).

defend membership inference attacks with competitive performance [10, 23, 28]. As discussed in Section 4.1, an adversary infers the membership of a sample based on the divergence of the prediction confidences between members and non-members. Such divergence becomes more severe as the number of training epochs increases, due to the increased memorization. Hence, the early stopping mechanism with fewer training epochs and l2 regularization for penalizing the over-training can tradeoff a slight reduction in model accuracy with lower privacy risk. In the evaluation, we stop the training and fine-tuning when the validation loss is not decreased for five epochs using early stopping mechanism. In l2 regularization, we set the regularization factor as 0.0005. *Note that we use early stopping and l2 regularization in all the other defenses to improve the defense performance.*

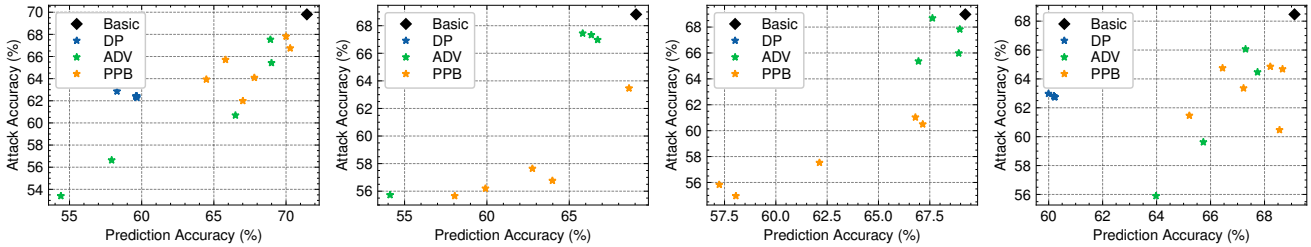
Differential Privacy (DP). Differential privacy is a strategy to bound the individual information exposure when running an algorithm f and has been widely investigated for preventing privacy leakage against membership inference attacks [14, 58, 59]. We implement differentially private SGD (DPSGD) [31, 59], one of the most widely-used defense techniques, to train neural networks with DP guarantees. Following DPSGD, we first clip the gradient, then add noise to the gradient, and use the generated noisy gradient to update the model's parameters. The noise is sampled from a Gaussian distribution $\mathcal{N}(0,\sigma)$. To achieve (ϵ,δ)-DP, the standard deviation of the Gaussian distribution, i.e., δ, should be in the order of $\Omega(q\sqrt{T\log(1/\delta)/\epsilon}$, where q denotes the sampling ratio and T denotes the total number of iterations. Accordingly, the privacy guarantee of the DP defense can be derived from δ, which plays an important role in balancing utility and privacy. Therefore, in the defense evaluation, we evaluate the effectiveness of DP defense and explore the impact of different privacy budgets (*i.e.*, different values of δ).

Adversarial Regularization (ADV). Nasr *et al.* proposed to consider the membership inference adversary in the training process [24]. The defender first trains a surrogate attack model to distinguish between members and non-members and then trains the target model to minimize the prediction loss while maximizing the classification loss of the surrogate attack model. A parameter α is used to balance the prediction performance and privacy risk. ADV is applied in the fine-tuning process of pruning to protect the privacy of pruned models.

6.3.2 Experimental Results of Defenses

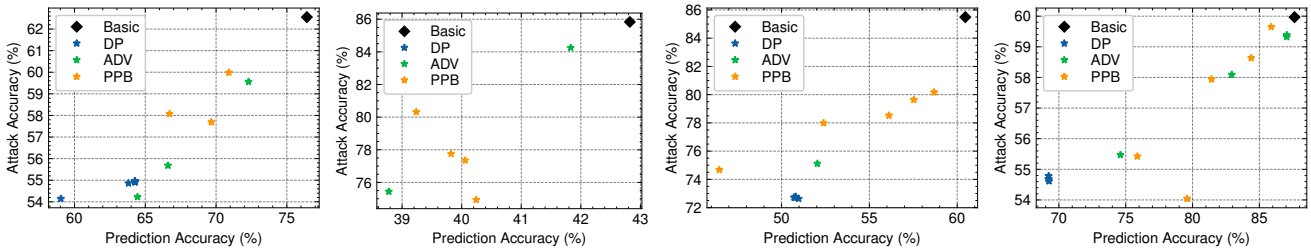
We use the same settings of attack evaluations in Section 5 and conduct the following experiments with defenses in the process of pruning. Since there is always a trade-off between privacy and prediction accuracy when implementing defenses, we explore different settings of hyper-parameters in the defensive mechanisms to thoroughly evaluate the defense performance. Specifically, we set hyper-parameter $\lambda \in \{1,2,4,8,16\}$ in PPB, $\sigma \in \{0.01,0.1,1,10,100\}$) in the DP noise vectors, $\alpha \in \{0.5,1,2,4,8,16\}$ in ADV, respectively.

Figure 16 and 17 illustrate the prediction accuracy and attack accuracy with different defense mechanisms. For better



(a) CIFAR10, ResNet18 (b) CIFAR10, DenseNet121 (c) CIFAR100, DenseNet121 (d) Location, FC

Figure 18: Performance of defenses against adaptive attacks for different pruning approaches (CIFAR10, ResNet18, Sparsity 0.6).



(a) CIFAR10, DenseNet121 (b) CIFAR100, DenseNet121 (c) Location, FC (d) Purchase, FC

Figure 19: Performance of defenses against adaptive attacks for different datasets (L1 Unstructured, Sparsity 0.6).

illustration, we remove the results if the model with a specific hyper-parameter cannot achieve 75% of the basic defense's prediction accuracy, *i.e.*, poor prediction performance, or result in a higher attack accuracy, *i.e.*, ineffective defense.

We observe that PPB is especially effective in protecting all pruning approaches from attacks, which can reduce the attack accuracy to around 50% (random guessing accuracy), while not degrading the prediction accuracy too much. Hence, *PPB defense provides a privacy-preserving approach with minimal degradation of prediction accuracy*. In addition, ADV is also effective in the L1 unstructured and Slimming pruning, but fails to achieve a good balance between prediction performance and privacy in the L1 structured and L2 structured pruning. Similar to the fact shown in recent work [32], we observe DP can hardly balance this utility-privacy tradeoff.

6.3.3 Defenses against Adaptive Attacks

To rigorously evaluate the defense performance, we consider adaptive attacks, where the adversary knows all the details of defenses along with the pruning information. In adaptive attacks, the adversary trains a shadow pruned model following the same defense mechanism (*e.g.*, Basic, DP, ADV, and proposed PPB) and pruning process. The adversary then performs the SAMIA attack based on the shadow pruned model.

As shown in Figure 18 and 19, we observe that PPB reduces the accuracy of adaptive attacks compared to the attacks on the pruned model without defenses and provides the best protection in L1 structured and L2 structured pruning. Besides, for the L1 unstructured and Slimming pruning, PPB and ADV are the best two defenses. *PPB is designed towards pruned models by reducing the confidence and sensitivity gap. Therefore, in general, PPB provides good protection in all pruning approaches.* In addition, ADV is designed to mitigate the confidence gap, which is largely increased in L1 unstructured and slimming pruning (as discussed in Section 5.2.2). Hence, ADV is also effective in protecting pruned models using L1 unstructured and slimming pruning.

7 Conclusion

This paper conducted the first analysis of privacy risks in neural network pruning. We first explored the impacts of neural network pruning on prediction divergence, based on which, a new membership inference attack, *i.e.*, self-attention membership inference attack (SAMIA), is proposed against the pruned neural network models. Through comprehensive and rigorous evaluation, we demonstrated the substantially increased privacy risks of the pruned models. We found that the privacy risks of the pruned models are tightly related to the confidence gap, sensitivity gap, and generalization gap due to pruning. Besides, even without knowing the pruning approach, the membership inference attacks can still achieve high attack accuracy against the pruned model. Especially, the proposed SAMIA showed superiority in identifying the pruned models' prediction divergence by using finer-grained prediction metrics, which is recommended as a competitive baseline attack model for future privacy risk study of neural

network pruning.

In addition, to defend the attacks, we proposed a pair-based posterior balancing named as PPB by reducing the prediction divergence of fine-tuning process during neural network pruning. We experimentally demonstrated that PPB could reduce the attack accuracy to around 50% (random guessing accuracy) without considering adaptive attacks and achieve the best protection compared with the three existing defenses. Besides, PPB showed competitive performance even when defending adaptive attacks.

The proposed SAMIA attack will be further explored under more challenging MIA settings, such as the label-only MIA without available confidences, where the existing label-only MIA attacks using data augmentation [60] and black-box adversary [61] can be potentially integrated for more powerful attack capability. We hope our work convinces the community about the importance of exploring innovative neural network pruning approaches by taking privacy-preserving into consideration.

Acknowledgement

We would like to thank our shepherd, Yinzhi Cao, and the anonymous reviewers for their constructive suggestions. This work was supported in part by National Science Foundation (CCF-2106754).

Availability

Our code is publicly available at https://github.com/Machine-Learning-Security-Lab/mia_prune for the purpose of reproducible research.

References

[1] Michael Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in Neural Information Processing Systems (NIPS)*, 1988.

[2] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[3] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR) (Poster)*, 2017.

[4] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[5] Davis W. Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John V. Guttag. What is the state of neural network pruning? In *MLSys*, 2020.

[6] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, 2019.

[7] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019.

[8] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.

[9] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *Conference on Data and Application Security and Privacy (CODASPY)*, 2021.

[10] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

[11] Iyiola E. Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. *arXiv preprint arXiv:2101.06570*, 2021.

[12] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Trans. Assoc. Comput. Linguistics*, 2020.

[13] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2019.

[14] Junjie Chen, Wendy Hui Wang, and Xinghua Shi. Differential privacy protection against membership inference attack on machine learning for genomic data. In *PSB*, 2021.

[15] Yang Zou, Zhikun Zhang, Michael Backes, and Yang Zhang. Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning. *arXiv preprint arXiv:2009.04872*, 2020.

[16] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations (ICLR)*, 2020.

[17] Bailin Li, Bowen Wu, Jiang Su, and Guangrun Wang. Eagleeye: Fast sub-net evaluation for efficient neural network pruning. In *European Conference on Computer Vision (ECCV)*, 2020.

[18] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations (ICLR)*, 2020.

[20] Michela Paganini. Prune responsibly. *arXiv preprint arXiv:2009.09936*, 2020.

[21] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? arxiv e-prints, art. *arXiv preprint arXiv:1911.05248*, 2019.

[22] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.

[23] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

[24] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[25] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *USENIX Security Symposium*, 2020.

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

[27] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[28] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2021.

[29] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.

[30] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. *J. Priv. Confidentiality*, 2016.

[31] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2016.

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

[33] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR) (Poster)*, 2015.

[34] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[35] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[36] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *AISec@CCS*, 2017.

[37] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision (ECCV)*, 2018.

[38] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

[41] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2019.

[42] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[45] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *Network and Distributed System Security Symposium (NDSS)*, 2021.

[46] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[47] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6(1):1–11, 2016.

[48] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[49] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Trans. Intell. Syst. Technol.*, 2016.

[50] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *J. Netw. Comput. Appl.*, 2015.

[51] Hospital discharge data public use data file.

[52] Acquire valued shoppers challenge.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR) (Poster)*, 2015.

[54] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[55] David Saad. Online algorithms and stochastic approximations. *Online Learning*, 5:6–3, 1998.

[56] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[57] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[58] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. Effects of differential privacy and data skewness on membership inference vulnerability. In *TPS-ISA*, 2019.

[59] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. Differential privacy defenses and sampling attacks for membership inference. In *AISec@CCS*, 2021.

[60] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International Conference on Machine Learning*, pages 1964–1974. PMLR, 2021.

[61] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–895, 2021.