Label-Only Membership Inference Attacks and Defenses In Semantic Segmentation Models

Guangsheng Zhang, Bo Liu, Tianqing Zhu*, Ming Ding, and Wanlei Zhou

Abstract—Recent research has discovered that deep learning models are vulnerable to membership inference attacks, which can reveal whether a sample is in the training dataset of the victim model or not. Most membership inference attacks rely on confidence scores from the victim model for the attack purpose. However, a few studies indicate that prediction labels of the victim model's output are sufficient for launching successful attacks. Besides the well-studied classification models, segmentation models are also vulnerable to this type of attack. In this paper, for the first time, we propose the label-only membership inference attacks against semantic segmentation models. With a well-designed framework of the attacks, we can achieve a considerably higher successful attacking rate compared to previous work. In addition, we have discussed several possible defense mechanisms to counter such a threat.

Index Terms—Membership Inference Attacks, Semantic Segmentation, Differential Privacy, Deep Learning

1 Introduction

Deep learning technologies have brought numerious successful applications, such as face recognition [1], image classification [2], and semantic segmentation [3]. The success of these technologies is due to the availability of large-scale datasets. These datasets enable a better training process and in turn, more accurate deep learning models. However, the models in deep learning applications often contain sensitive information and pose privacy leakage risks. Although the deep learning model structures are usually hidden, the attackers can still extract private information by making queries to the victim model. The work of [4] demonstrated that attackers could reconstruct some information on the model's training data by identifying the data sample's membership. This type of attack is called the membership inference attack. The attacker was assumed to have black-box access to the victim model to obtain confidence scores of the model prediction after multiple queries. Using the queried confidence scores, the attacker could infer whether a specific data sample was in the training data or not.

A large group of research works focused on studying the design or understanding the membership inference leakage in deep learning models [5], [6], [7], [8]. Defense mechanisms against membership inference attacks have also been developed [4], [9], [10], [11]. However, obtaining confidence scores of the model prediction to launch membership inference attacks is not always practical because the deep learning models deployed in real-world applications usually do not have APIs to be queried with confidence scores, or the prediction scores have already been modified by internal defense mechanisms.

- *Tianqing Zhu is the corresponding author. G. Zhang, B. Liu, T. Zhu are with the Centre for Cyber Security and Privacy and the School of Computer Science, University of Technology, Sydney, Australia. Email: Guangsheng.Zhang@student.uts.edu.au, Bo.Liu@uts.edu.au, Tianqing.Zhu@uts.edu.au. M. Ding is with Data61, CSIRO, Australia. Email: Ming.Ding@data61.csiro.au. W. Zhou is with City University of Macau, Macao. Email: wlzhou@cityu.edu.mo.*

Recent work of [12] and [13] showed that obtaining confidence scores of the victim model is not mandatory to launch a membership inference attack. They made fewer assumptions of the attack by only having the victim model's prediction labels. The intuition is that deep learning models have higher confidence in predicting member samples than non-member samples. By making various data augmentations to the original training data, member and non-member samples have different performances in prediction labels. Then with the gathered prediction labels as the attack model input data, the attacker differentiates the member samples from non-member samples.

Previous research mostly concentrated on membership inference attacks in image classification models [6], or generative models [14]. Researchers of [15] demonstrated that membership inference attacks could also target other models, such as semantic segmentation models. However, such an initial work assumed that attacks should require the confidence scores.

Inspired by the previous work, we study the problem of label-only membership inference attacks against semantic segmentation models. That is, given only the victim model's prediction labels, can the attacker still tell if a specific record was used in a semantic segmentation model? This research has much potential in applying to real-world applications. Semantic segmentation tasks have already been adopted in many commercial or under-development products. Privacy leakage exists in image segmentation applications for autonomous driving [16] and robot navigation [17], because the attackers can utilize the segmentation results to figure out the users' location or other sensitive information. Applications for medical data segmentation [18] can leak the patients' diagnosis and health condition information. These applications are deployed with models with privacy risks, and the attacker can only obtain prediction labels from querying deployed applications. The label-only attacks make it possible to extract private information even from a seemingly private setting of revealing hard-value classification results. The discussed uniqueness of this work sets

us apart from the existing work on membership inference attacks.

Although techniques of membership inference attacks have been developing during the past few years, a few challenges still remain: Firstly, in order to deploy attacks in real-world applications, the adversary knowledge needs to be as little as possible (e.g. label-only attacks). Otherwise, the attacks might not be meaningful in practice. Secondly, the extension from classification tasks to semantic segmentation tasks is not trivial. Although semantic segmentation can be considered as a collection of pixel classification, the information contained in a single pixel is limited, which results in a unreliable indicator of classification label in the prediction output. Multiple procedures are needed to process these unreliable indicators of prediction labels in an efficient way. Also, the pixels are not equal in predicting the output, with some pixels carrying more information than the others, which should be considered in the data processing procedures as well. Further, the label-only attacks in the semantic segmentation tasks require more strategies to extract membership information from the data samples, which is a major difference from the label-only attacks in other tasks.

To tackle the above challenges, we design a new attack framework: We apply different data augmentations to the data samples to obtain more adversary knowledge, and then we adopt several post-processing strategies (prediction-label concatenation and patch cropping) to the victim model's prediction output to apply the attacks against semantic segmentation models. Our contributions in this paper are the following:

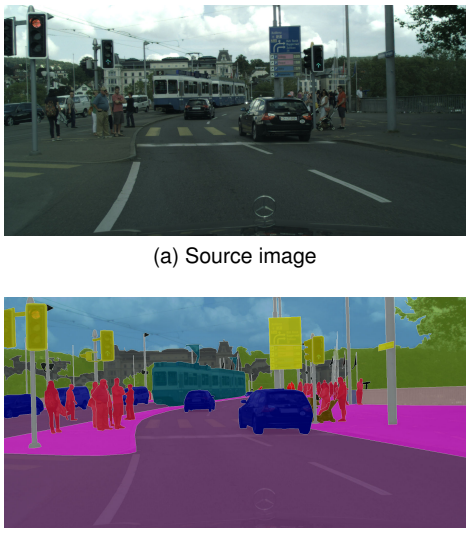
- We propose the first label-only membership inference attacks against semantic segmentation models.
- We design a framework for membership inference attacks by applying different data augmentations to the data samples, and several post-processing strategies to the victim model's prediction output.
- We discuss several defense mechanisms against membership inference attacks in semantic segmentation tasks.
- We achieve competitive experimental results of attack and defense methods compared to previous research.

2 Preliminaries and Related Work

2.1 Semantic Segmentation

Being a vital computer vision task towards complete scene understanding, semantic segmentation is a pixel-level labeling task for all image pixels, which labels all the objects, stuff, or background areas in the image to each category [19]. Figure 1 gives an example of what semantic segmentation looks like between a source image and its corresponding annotated labels using colorized visualization. A semantic segmentation label image is a gray-scale image where each pixel represents its class number in a real task.

With deep learning technologies widely adopted in computer vision, many semantic segmentation models have been developed based on deep learning. Several prominent deep learning based semantic segmentation models are fully convolutional networks [3], encoder-decoder based models [20], multi-scale and pyramid based models [21], dilated convolutional models [22], and attention based models [23].



(a) Source image

(b) Visualization of annotated labels

Fig. 1. An example of the semantic segmentation task.

One of the first deep learning models in semantic segmentation tasks applied a fully convolutional network (FCN) [3]. SegNet [20] was an encoder-decoder based model, which extracted feature maps in the encoder process and then up-sampled the lower resolution feature maps to the original resolution. Pyramid scene parsing network (PSPNet) [21] adopted a multi-scale network to learn the global context representation and then processed the patterns from different scales with a pyramid pooling. Deeplabv3+ [22] used dilated convolutional layers to solve the decreasing resolution in the model and an atrous spatial pyramid pooling (ASPP) to extract feature maps. Although different deep learning technologies are applied in these models, the main goal in semantic segmentation is to extract either local or global features in the image.

In this paper, we focus on membership inference attacks against semantic segmentation models. And we test our attacks on PSPNet [21], UperNet [24], DANet [25], and Deeplabv3+ [22].

2.2 Membership Inference Attacks

Membership inference attacks aim to find whether a specific data sample has contributed to the victim model's training. The adversary does not have direct access to the training dataset or the trained model. However, based on the observations of the victim model's prediction output, the adversary can predict whether a specific data sample is in the victim model's training data or not.

Shokri *et al.* [4] pioneered the topic of membership inference attacks against image classification, leveraging multiple shadow models to generate data to train multiple attack models. Salem *et al.* [6] relaxed the assumptions of the attacks, showing that models and datasets of the victim and shadow models can be independent. They also demonstrated that one shadow model was already enough for the attacks. Yeom *et al.* [7] showed that the overfitting

TABLE 1
Notations

Notation	Description
V	Victim model, or called target model
S	Shadow model
A	Attack model
D	Dataset for the model
X	A set of images in the dataset
Y	A set of corresponding ground truth labels in the dataset
P	Prediction result of a model, e.g. $P = V(X)$

feature of models could lead to the models' vulnerability to membership inference attacks. Choquette-Choo *et al.* [12] and Li *et al.* [13] both proposed attacks with only access to the victim model's prediction labels. Instead of using confident scores of the victim model's output, they applied various data augmentations to the original images and obtained corresponding prediction labels as membership information. More recent works studied the influence of data augmentations on membership inference attacks, highlighting the privacy risk in the models trained with augmented data [26], [27]. Other new works proposed novel attack methods [28] or conducted assessments on the performance of the attacks [29], [30].

While prior work has mostly studied attacks against classification models, there are several papers extending the scenarios to attacks against other deep learning models or learning settings. Very recent studies discovered that membership inference attacks were also possible in semantic segmentation [15], object detection [31], generative models [14], [32], and transfer learning [33].

Defense mechanisms against membership inference attacks usually fall into two categories. The first category usually tried to solve the overfitting issue of the victim model, as overfitting could lead to the exposure of individual samples [4], [7]. Dropout [6] or differential privacy [34], [35] could be used to reduce the successful attack rate. The second category advocates perturbation to the victim model's confidence scores to break the attacks [4], [10], [36]. Shokri *et al.* [4] proposed several possible strategies for changing the confidence scores. Jia *et al.* [10] presented MemGuard, a strategy of adding noises to the victim model's prediction to confuse the attack model. However, this kind of defense mechanism cannot be applied to the label-only attacks since the confidence scores are not observable.

Our research focus is on label-only attacks against semantic segmentation models. The attacks can be deployed in real-world applications and a deep understanding of such attacks is in urgent need. In this work, we provide a thorough and systematic study of this research area.

2.3 Architecture of the Attack Framework

There are three models involved in the task of membership inference attacks. The *target model* or *victim model* is the target of the attack. Member samples are in the training data of the model, while non-member samples are not. The deep learning model usually behaves differently when the model meets member and non-member samples. The model's prediction output follows a different pattern or data distribution because of the overfitting nature of the model [4], [6].

To launch a more successful attack, a *shadow model* is created to mimic the victim model's prediction output, such as the different patterns or data distribution between member and non-member samples. Then, leveraging the shadow model's prediction output, we can train an *attack model*, which is a classifier to differentiate member samples from non-member samples.

In this paper, we follow the steps of [15] to study attacks in semantic segmentation models. Instead of using confidence scores of the model's predictions, we relax the membership inference attacks' assumptions to use the prediction labels only, which is inspired by [12], [13]. Table 1 lists the notations used in our framework.

3 Problem Formulation

3.1 Threat Model

As our goal is to show whether label-only membership inference attacks can match the performance of previous research in semantic segmentation models, we propose a threat model similar to prior work [4], [12], [15], which means the adversary only has black-box access to the model, i.e., the adversary does not have access to the model parameters and can only make queries to obtain model predictions or confidence scores. The details of the threat model are as follows.

3.1.1 Task Knowledge

Task knowledge refers to the type (in our case, semantic segmentation), the scenario (street view scenes), the class labels (cars, pedestrians, road, and other labels in street view scenes), the input image format (RGB images), etc. The task knowledge is assumed to be known to the adversary in this paper.

3.1.2 Model Knowledge

Model knowledge refers to any knowledge related to the victim model, including the model parameters, the size of the training dataset, the number of the training iterations and epochs, the setup of optimizers. The adversary does not have any model knowledge of the victim model.

The knowledge of the victim model's structure depends on the attack setting. This can either be known (dependent attacks) or unknown (independent attacks). Please refer to Section 6.1 for more information on our experimental setup.

As our study is on label-only attacks, we do not need the confidence scores of the victim model's output. The adversary can only obtain the prediction labels of the victim model's output, which is more realistic in real-world applications because the confidence scores might not be supported by the associated API or they might have been altered internally due to privacy protection reasons. Hence, the ground truth labels are essential for extracting the membership information.

3.1.3 Data Knowledge

The adversary is aware of the distribution of the training dataset of the victim model and can collect a new dataset based on the same distribution. The new dataset could be

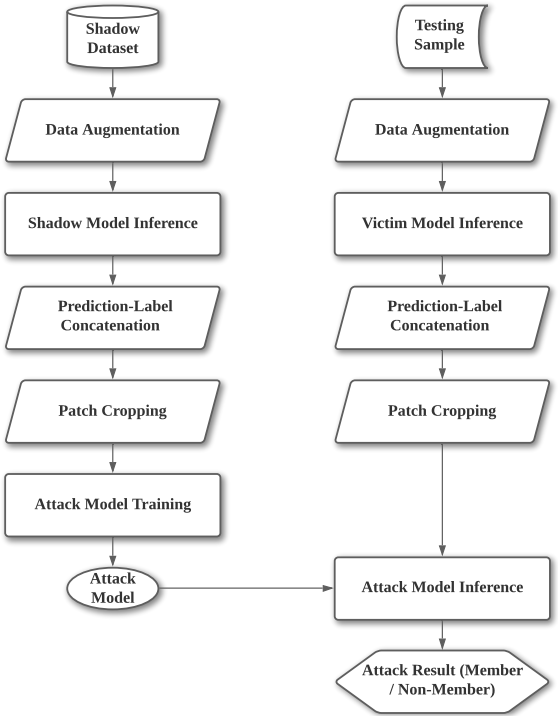


Fig. 2. The framework of the label-only membership inference attack against semantic segmentation models.

real or synthetic based on different experimental setups [4], and it should not overlap with the victim model's training dataset. We call this new dataset the shadow dataset. Besides, the adversary cannot directly access the victim model's training dataset. Please refer to Section 6.1 for more information on our experimental setup.

3.2 Design Goal

In order to tackle the limitations of current membership inference attacks, we design the framework of membership inference attacks in a label-only setting for semantic segmentation models. To this end, we aim to achieve the following design goals:

- The first goal is to design a pipeline framework to initiate the membership inference attacks in a label-only setting for semantic segmentation models. In Section 4, we describe our pipeline framework, including the training and testing of the attack model and also the details of our data representation procedures.
- The second goal is to evaluate whether our label-only attacks can achieve a better attack accuracy than other attacks against semantic segmentation models. In Section 6, we evaluate our attack framework in various settings.
- The third goal is to evaluate whether the current defense mechanisms can defend our attack settings. We introduce several defense mechanisms in Section 5 and test their effectiveness against our attack framework in Section 6.

4 Label-Only Attack Framework in Semantic Segmentation

In this section, the label-only membership inference attack against semantic segmentation models is introduced. Figure 2 shows the proposed attack framework, including the training and testing of the attack model. The following subsections will discuss the details of this framework.

4.1 Data Preparation

Before training the attack model, We need to prepare the datasets for the victim/shadow model for model-building. In more detail, we build a shadow model $\boldsymbol{S}$ similar to the victim model $\boldsymbol{V}$. The shadow model is expected to exhibit similar behavior in predicting outputs like the victim model when they infer member and non-member samples. Our attack model $\boldsymbol{A}$ is a classifier that differentiates member and non-member samples.

We prepare two datasets D^V and D^S for the victim model $\boldsymbol{V}$ and the shadow model $\boldsymbol{S}$, where $D^V = \{(X_i^V, Y_i^V)\}_i$, $D^S = \{(X_i^S, Y_i^S)\}_i$. Here, X_i represents a single image in the dataset, and Y_i denotes the corresponding ground truth labels, with one label for each pixel in the image. These two datasets can be dependent or independent [6], which will be discussed in detail in Section 6.1. We split each dataset into two sub-datasets for the victim/shadow model's training and testing. The shadow model's training data D_1^S and testing data D_0^S are considered as member/non-member samples for the attack model, where 1 and 0 denote the binary membership status. The same setting applies to the victim model's training data D_1^V and testing data D_0^V. In reality, sub-datasets D_1^S and D_0^S are prepared by the attacker, and the attacker's final goal is to determine whether the testing sample (x, y) belongs to D_1^V or D_0^V.

4.2 Training of the Attack Model

The training process of the attack model is discussed in this subsection. Algorithm 1 presents the pseudocode of the model training.

We prepare the dataset D^A of the attack model $\boldsymbol{A}$ based on the victim/shadow model's output. The training/testing data D_{train}^A and D_{test}^A of the attack model $\boldsymbol{A}$ are constructed by the shadow/victim model's output respectively. D_{train}^A and D_{test}^A is formulated by the following data representation procedure, including one (data augmentations) before the victim/shadow model inference, and the other two (prediction-label concatenation and patch cropping) after the inference:

- **Data Augmentation.** Multiple data augmentation methods are applied in our algorithm, which are denoted as $aug(\cdot)$. The original images X^V and X^S are processed by different data augmentation methods, the outputs of which are denoted by $aug(X^V)$ or $aug(X^S)$. Further, they are processed by the victim/shadow model. We take the prediction label of the output, which is denoted as $P1^V = V(aug(X^V))$ or $P1^S = S(aug(X^S))$.
- **Prediction-Label Concatenation.** We concatenate victim/shadow model predictions with labels to

give a second data representation, which is denoted as $P2^V = pl\_concat(P1^V, Y^V)$ or $P2^S = pl\_concat(P1^S, Y^S)$.

- **Patch Cropping.** We crop several patches by some rules from the second data representation result as the input of the attack model denoted as $P3^V = crop(P2^V)$ or $P3^S = crop(P2^S)$.

We introduce these three steps of the data representation procedure in the following paragraphs. These three steps for the attack model's training and testing are the same. The only difference is the dataset and the model (victim or shadow) used in these steps. In this subsection, we construct D^A_{train} with D^S and $\boldsymbol{S}$.

4.2.1 Data Augmentation

As mentioned in [12], [37], data augmentation is a very powerful tool to build deep learning models. In order to increase the performance of deep learning models, the goal is to minimize the distance between the training and testing data. The augmented training data will give a better representation of the training dataset to achieve this goal.

However, this data augmentation process makes deep learning models vulnerable to membership inference attacks, especially label-only attacks. With only the prediction labels available in the label-only attacks, the information is not enough to initiate the attack. The intuition here is to generate more information to extract membership status. If a data sample is used to train the victim/shadow model, the augmented data samples are likely to participate in training the victim/shadow model as well. In this way, the attacker leverages augmented input data to obtain a better membership signal.

We apply several different data augmentation methods, two of which (translation and rotation) are similar to data augmentations in [12], while the others are new in our paper.

- Translation. Given a translation scale s, we translate the image by $\pm s$ pixel horizontally and vertically. We output five images in total, including the original one.
- Rotation. Given a rotation scale s, we rotate the image by $\pm s^\circ$. We output three images in total, including the original one.
- Brightness, contrast, hue, or saturation. These four photometric distortions share the same strategy: we randomly apply one of them to the image and output five images, including the original one.
- Random. We randomly select the above six data augmentations and apply them to the image. We output five images in total, including the original one.

With the augmented image input, we receive the victim/shadow model's inference result, the prediction labels. As this is the training of the attack model, we receive the shadow model's prediction labels denoted as $P1^S$. In a normal membership inference attack, the inference result is the confidence scores of the prediction. This is formulated as a pixel-wise matrix of size (c, h, w), where c is the number of class labels of the dataset, h and w are the height and width of the image. With a softmax layer being the last layer of the model, the matrix values range between 0 and 1, each of which denotes the probability of a class for a single pixel. In our label-only attack, the prediction labels $P1^S$ is a matrix of size (n, h, w), where n means the number of data augmentations of one image. Each value in the matrix means the class ID in a single pixel.

Algorithm 1 Training of the Attack Model

Input: $D^S = D^S_1 \cup D^S_0 = \{X^S, Y^S\}, \boldsymbol{S}, epoch\_num$
Output: $\boldsymbol{A}$

$i = 0$
while $i < len(D^S)$ **do**
  $P1^S_i = S(aug(X^S_i))$
  $P2^S_i = pl\_concat(P1^S_i, Y^S_i)$
  $P3^S_i = crop(P2^S_i)$
  $i = i + 1$
end while
Reshape $P3^S_i$ to $P3^S$ for each $P3^S_i$ has k cropped patches
$P3^S$ can be split into $P3^S_1$ (from D^S_1) and $P3^S_0$ (from D^S_0)
$m = 0, n = 0$
while $m < epoch\_num$ **do**
  while $n < len(P3^S)$ **do**
    if $P3^S_{(m,n)} \in P3^S_1$ **then**
      $A(P3^S_{(m,n)})$ train as 1
    else
      $A(P3^S_{(m,n)})$ train as 0
    end if
    $n = n + 1$
  end while
  $m = m + 1$
end while
return $\boldsymbol{A}$

Algorithm 2 Testing of the Attack Model

Input: Testing sample (x, y) from $D^V, \boldsymbol{V}, \boldsymbol{A}$, membership threshold κ
Output: $Result$

$p1 = S(aug(x))$
$p2 = pl\_concat(p1, y)$
$p3 = crop(p2)$
$p3$ has k cropped patches
$m = 0$, $Result = 0$
while $m < k$ **do**
  $Result += A(p3_m)$
  $m = m + 1$
end while
$Result = \frac{Result}{k}$
if $Result >= \kappa$ **then**
  Testing sample (x, y) is a member sample
else
  Testing sample (x, y) is a non-member sample
end if
return $Result$

4.2.2 Prediction-Label Concatenation

In this step, we leverage the prediction labels $P1^S$ and the ground truth labels Y^S to provide a better data representation to the next step. In order to differentiate member samples from non-member samples, the attack model needs input data to contain information from both the prediction labels and the ground truth labels. The ground truth labels Y^S is a matrix of size $(1, h, w)$. Here we adopt three different strategies to form the output of the prediction-label concatenation $P2^S$.

- **Simple concatenation.** We simply perform a matrix concatenation between $P1^S$ and Y^S, leading to a matrix of size $(n+1, h, w)$ as the output $P2^S$:

$$P2^S = matrix\_concat(P1^S, Y^S), \tag{1}$$

where $maxtrix\_concat(\cdot)$ means the matrix concatenation process.

- **Mixup and one-hot concatenation.** Unlike confidence scores of the prediction output, the prediction label of each augmented output has only one channel instead of c channels. To mimic the matrix structure of the confidence score output, we perform the one-hot encoding to each channel of the prediction label matrix and then perform Mixup [38]:

$$M_{all} = \sum_{i=1}^{n} \lambda_i M_i, \ \sum_{i=1}^{n} \lambda_i = 1, \tag{2}$$

where M_{all} is the matrix after Mixup, M_i is the ith the prediction label matrix with the size (c, h, w), and λ_i is the corresponding weight. In this way, we receive the matrix M_{all} with the size (c, h, w). The Mixup process reduces matrix dimensions from $n \times c$ layers to c layers and still contains all the information on each prediction label.
We then concatenate the matrix M_{all} to the one-hot encoded ground truth label (a matrix of size (c, h, w)), leading to a matrix of size $(2c, h, w)$ as the output $P2^S$:

$$P2^S = matrix\_concat(M_{all}, one\_hot(Y^S)), \tag{3}$$

where $one\_hot(\cdot)$ means the one-hot encoding process.

- **Mixup and structured loss map (SLM).** The Mixup process is the same as Mixup and concatenation. Then we calculate the structured loss map using the Mixup output M_{all} and the one-hot encoded ground truth label [15]:

$$P2^S = -\sum_{i=1}^{c} one\_hot(Y^S)_i log(M_{(all,i)}). \tag{4}$$

The structured loss map calculates cross-entropy loss values across all locations, resulting in a matrix of size $(1, h, w)$ as the output $P2^S$. This process significantly reduces the dimension of the output $P2^S$ from $2c$ to 1.

Based on different strategies, we obtain different data representations $P2^S$ with various sizes of the matrix.

4.2.3 Patch Cropping

Patch cropping is the final step of our data representation procedure, which is a procedure to crop one image into several patches based on a specific rule. The membership indicator information in the data representation is still weak and not good enough to support an effective attack [15]. Applying patch cropping methods to the data representation $P2^S$ can aggregate more information over patches to launch a stronger attack. For each $P2_i^S$ in $P2^S$, we make the cropping procedure to form k patches, denoted as PA^S. PA_k^S has 4 variables $(x\_idx, y\_idx, pa\_h, pa\_w)$, where x_idx and y_idx are the coordinate of the upper-left point of the patch, and pa_h and pa_w are the height and width of the patch. In this paper, we crop patches of the prediction-label concatenation output $P2^S$ by the following rules.

- **Random.** We crop patches across the matrix $P2^S$ randomly.
- **Sliding windows.** We crop patches with a fixed step size from the upper-left to the bottom-right of the matrix. This method can guarantee that all the information in the matrix is included in the cropped patches.
- **Random rejection to preserve diversity in labels.** We reject some randomly cropped patches if the patch has most pixels of one label. We set a rejection degree η for determining when to reject the cropped patches. For example, road areas are prevalent in street scenes and may take a large portion in a single image. Here, η is set to 80%, and we reject this type of patch if road labels are observed for more than 80% of the pixels.

After the patches are cropped, we obtain the final data representation $P3^S$, which forms the training dataset of the attack model D_{train}^A. $P3^S$ can be described as:

$$P3^S = \{PA_k^S\}_k = crop(P2^S). \tag{5}$$

Then we construct a classification model $\boldsymbol{A}$ to differentiate member and non-member samples in the original shadow dataset $\boldsymbol{S}$.

4.3 Testing of the Attack Model

In this subsection, we apply the testing sample (x, y) from D^V and $\boldsymbol{V}$ to construct $P1^V$, $P2^V$, and $P3^V$, which finally forms the testing dataset of the attack model D_{test}^A. Then we obtain the inference result of $\boldsymbol{A}$ to determine whether (x, y) is a member sample or a non-member sample. Algorithm 2 presents the pseudo-code of the attack model testing. The membership threshold κ required in the algorithm is usually set to 0.5.

There is another difference between the training and testing of the attack model. The inference result of the attack model is the classification result of the patches, not the whole image. The result of the patches in the same image should be calculated together, and the result of a single image should be as follows [15]:

$$Result = \frac{1}{N} \sum_{i=1}^{N} A(X_i^A, Y_i), \tag{6}$$

where X^A means the input of the attack model and (X_i^A, Y_i) means the i-th patch in the same image with the corresponding ground truth label.

5 Defense Mechanisms Against Label-Only Attack Framework In Semantic Segmentation

This section presents possible defense mechanisms for label-only membership inference attacks to protect private infor-

mation in semantic segmentation models. Previous research proposed several defense mechanisms in the training or testing phase of the victim model to mitigate membership leakage.

Defense mechanisms in model testing usually try to add perturbations or make changes to the confidence scores of model predictions. In this way, the defender does not need to retrain the model to protect privacy. There are several developed strategies in the existing work, such as restricting the prediction scores to top k classes [4], coarsening precision of the prediction scores [4], MemGuard (a mechanism to add noises to prediction scores) [10], and adding Gaussian noises to prediction scores [15]. However, these defense mechanisms are not suitable for our attack algorithm since only the victim model's prediction labels are available in our scheme. The mentioned mechanisms were not meant for the prediction labels.

Therefore, defense mechanisms in model training could be a possible strategy to protect privacy from label-only membership inference attacks. We discuss Dropout [39] and DPSGD [34] in the following subsections.

5.1 Dropout

To reduce overfitting of deep learning models, Hinton *et al.* [39] proposed dropout, a regularization method. The idea is to randomly drop a few units and their connections from deep neural networks during training. The dropout ratio is a hyper-parameter for setting the probability of retaining a unit in the network.

Salem *et al.* [6] demonstrated that dropout could also be used to defend against membership inference attacks because one of the requirements of successful attacks is the overfitting feature of the victim model.

5.2 DPSGD

Differential privacy [40] provides a standard for privacy guarantees of neighboring datasets. Differential privacy defines privacy from a mathematical perspective. The definition of differential privacy is as follows:

Definition 1. ((ϵ, δ) - Differential Privacy). For any two neighboring datasets D and D' that differ in only a single entry, a randomized mechanism $\mathcal{A}$ provides (ϵ, δ) - Differential Privacy, if for $\forall S \subseteq Range(\mathcal{A})$,

$$Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in S] + \delta, \qquad (7)$$

where ϵ measures the privacy loss between the neighboring datasets and δ is related to the size of the dataset.

Adopting differential privacy in deep learning model training can prevent the model from memorizing any individual data record. Abadi *et al.* [34] proposed a differentially private stochastic gradient descent (DPSGD) to provide strict privacy bound with similar utility compared to models without defense mechanisms. Compared to a standard SGD optimizer, DPSGD introduces the following changes to guarantee privacy: adding Gaussian noises to the gradient and clipping the gradient based on the gradient norm. Therefore, two parameters (noise scale σ and gradient norm bound C) should be calculated to ensure a better privacy guarantee. In [34], they also proposed a moments accountant (MA) method to calculate these parameters more tightly.

TABLE 2
Dataset and Model Settings

Setting	Dataset	Model
Dependent	Cityscapes, BDD100K, Mapillary	PSPNet
Independent	Cityscapes, BDD100K, Mapillary	PSPNet, DANet, UperNet, Deeplabv3+

6 Performance Evaluation

In this section, we first present our membership inference attack's experimental setup, then we demonstrate and analyze our experimental results.

6.1 Experimental Setup

6.1.1 Settings

As [6] has previously reported, the dataset and model settings in the victim and shadow model can be dependent or independent. The *dependent attack* means that the attacker has a dataset from the same data distribution as the victim model's training data. The shadow model has the same architecture as the victim model. The *independent attack* is more realistic, meaning sometimes the attacker has no knowledge of the victim model and its dataset. The attacker only knows the functionality of the victim model, which is semantic segmentation in our case. The attacker also knows that the samples in the victim model's training data are street scene images, so the shadow model is trained by some other similar datasets. The shadow model is not for mimicking the victim model's behavior but for capturing the dataset's membership status.

Table 2 shows the dataset and model settings in each experiment. Setting Dependent is used in the dataset and model-dependent attacks, while Setting Independent is used in the dataset and model-independent attacks. The dataset is usually split in a balanced setting (4 even subsets as victim member, victim non-member, shadow member, and shadow non-member) unless stated otherwise.

6.1.2 Datasets

We perform experiments using three well-known street scene semantic segmentation datasets:

- Cityscapes [16], a diverse set of images of street scenes from 50 different cities. As seen in Table 2, we split the dataset into different numbers of subsets based on different settings.
- BDD100K [41], a large-scale image dataset captured from driving videos by Berkeley AI Research.
- Mapillary Vistas [42], a fine annotated segmentation dataset of images of street scenes with various weather, season, camera, and viewpoint conditions. This dataset has the largest number of images in our experiments.

These three datasets have different image sizes. In order to have a unified model input, we resize the images of the

studied datasets to the same size. We also divide the images into member samples and non-member samples.

These three datasets also have different numbers of class labels. We transform the class labels in BDD100K and Mapillary Vistas to be the same as those in Cityscapes.

6.1.3 Models

We evaluate our attacks using the following semantic segmentation models (victim or shadow): PSPNet [21], UperNet [24], DANet [25], and Deeplabv3+ [22]. We select Resnet-50 [2] as our attack model.

6.1.4 Evaluation Metrics.

As there are only two classes (member and non-member samples) in our attack model, we denote member samples as positive samples and non-member samples as negative samples. To evaluate the effectiveness of our attack model, we count true positive (TP), false positive (FP), true negative (TN), and false negative (FN), precision and recall of our result.

To compare different attacks quantitatively, we used the above experimental results to evaluate our attack model in terms of three metrics:

- Attack accuracy: Attack accuracy means the rate of the accurate class of the attack model;
- F1 score: F1 score measures the overall performance of precision and recall;
- AUC score: AUC score quantifies the area size under ROC curve, which measures the trade-off between the true positive rate and the false positive rate [43].

In summary, the higher attack accuracy, F1 score and AUC score we get, the more effective the attack model is. We measure our attack methods in different settings using the above three metrics.

The experimental setup for defense mechanisms is similar to that for attack methods. As defense mechanisms are adopted in the victim model training, we will compare the attack model performance between models with and without defense mechanisms. The privacy metrics are the same, i.e., attack accuracy, F1 score, and AUC score. We also evaluate the performance between utility and privacy. The utility metrics in semantic segmentation models is mean intersection over union (mIoU) [44]. It is a standard metric for semantic segmentation, which computes a ratio between the intersection and the union of two sets (the ground truth and the prediction) on a per-class basis. It is then averaged over the results, which can be calculated as follows:

$$mIoU = \frac{1}{k}\sum_{i=1}^{k}\frac{ground\_truth \bigcap prediction}{ground\_truth \bigcup prediction} \quad (8)$$

6.2 Attack Experiment 1: Different Data Augmentation Scales

In our first experiment, we evaluate the performance of our membership inference attack under different scales of translation and rotation augmentations (from scale 1 to 11). As this is a dependent attack, we use Cityscapes as the victim and shadow dataset. We split the images in Cityscapes randomly into four subsets, as member/non-member data of the victim/shadow model. PSPNet is used as the victim/shadow model. We apply translation and rotation augmentations to the input images. And we adopt Resnet-50 as the attack model.

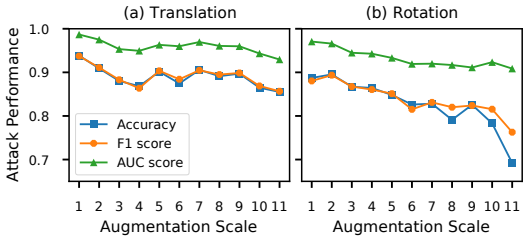


Fig. 3. Evaluation of the attack performance with translation and rotation augmentations under different scales.

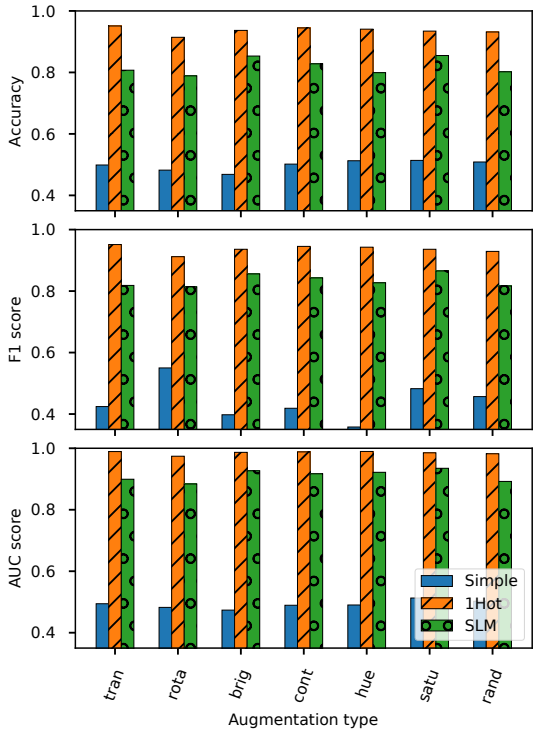


Fig. 4. Evaluation of the attack performance under different prediction-label concatenation strategies.

The attack performance is shown in Figure 3. For both translation and rotation augmentations, we can see that the performance decreases from around 0.9 to below 0.9 in terms of accuracy, F1, and AUC scores. The overall performance of translation augmentation is better than that of rotation augmentation, and the best performance is with a translation and rotation scale of 1.

In [12], they evaluated the performance of these two scales in image classification tasks, proving that too small or too large augmentations may harm the attack performance. We have obtained a similar result in semantic segmentation tasks: if the original victim/shadow model input images are largely augmented, the data augmentations cannot help to enhance the data sample's membership status. As a result, large data augmentations cause poor attack performance.

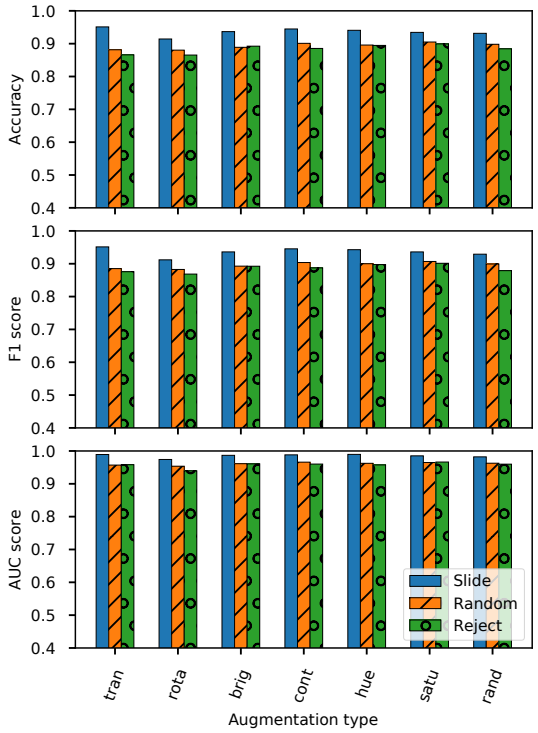


Fig. 5. Evaluation of the attack performance under different patch cropping methods.

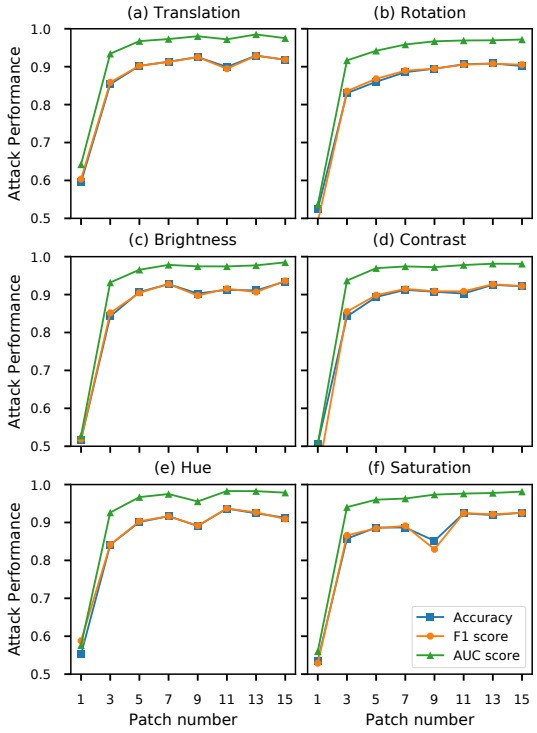


Fig. 6. Evaluation of the attack performance under different patch numbers.

We also discover that translation augmentation has a better attack performance than the rotation augmentation on most scales. In later experiments, we adopt translation and rotation augmentations with scale 1.

6.3 Attack Experiment 2: Different Prediction-Label Concatenation Strategies

In our second experiment, the attack performance of three different prediction-label concatenation strategies is evaluated. In this and later experiments, the attack performance is compared among various data augmentations, which include translation (tran), rotation (rota), brightness (brig), contrast (cont), hue, saturation (satu), and random (rand). We use the same dependent attack settings, with the "sliding-windows" patch-cropping method. We evaluate the performance of simple concatenation (Simple), Mixup and one-hot concatenation (1Hot), Mixup and SLM (SLM).

Figure 4 shows the evaluation of the attack performance. We discover that the Simple strategy performs the worst, with attack accuracy, F1 and AUC scores of around 0.5. This means the attack will not work when simply concatenating the prediction labels with the ground truth labels. This is expected because the output data in this strategy has the values of class IDs, which does not give a reasonable representation of the data.

1Hot and SLM strategies achieve over 0.8 in terms of attack accuracy scores, F1 scores and AUC scores, and the 1Hot strategy has the best performance. This is different from the conclusion in [15] that the SLM strategy performs the best. This is because of the different experimental settings between both papers. With limited label-only knowledge in our attack, our prediction map in the SLM strategy is different, which leads to different experimental results. In our attack framework, compared to 1Hot strategy, SLM strategy reduces the prediction matrix in the data representation matrix from $2c$ dimensions to 1, which leads to inevitable information loss. This eventually causes a lower attack accuracy.

It should be noted that the attack performance of different data augmentation types is very close. We can conclude that all types of data augmentation presented in this paper can improve attack performance. In later experiments, we apply 1Hot strategy as the first choice when concatenating prediction labels with ground truth labels in the data representation step.

6.4 Attack Experiment 3: Different Patch Cropping Methods

In our third experiment, we evaluate the attack performance with different patch cropping methods. We compare the attack performance in all seven data augmentations with SLM concatenation strategy. First, we set patch number to 5 and evaluate sliding-windows (Slide), random-cropping (Random), random-with-rejection (Reject) methods. The result in Figure 5 shows all three patch cropping methods can achieve scores over 0.8 in terms of attack accuracy, F1, and AUC scores. The Slide method has the best results (over 0.9), but all three approaches have comparable outstanding scores overall.

Next, we evaluate the attack performance in different patch numbers (from 1 to 15) with "Random" patch cropping method, as illustrated in Figure 6. The figures indicate that more patch numbers lead to better attack performance, with a peak attack accuracy of around 0.9. However, after

TABLE 3
Dataset size setting for Attack Experiment 4

Dataset Split Setting 1					Dataset Split Setting 2				
	Victim		Shadow			Victim		Shadow	
Type	M	NM	M	NM	Type	M	NM	M	NM
LessM	10%	40%	10%	40%	LessV	10%	10%	40%	40%
	15%	35%	15%	35%		15%	15%	35%	35%
	20%	30%	20%	30%		20%	20%	30%	30%
Balanced	25%	25%	25%	25%	Balanced	25%	25%	25%	25%
MoreM	30%	20%	30%	20%	MoreV	30%	30%	20%	20%
	35%	15%	35%	15%		35%	35%	15%	15%
	40%	10%	40%	10%		40%	40%	10%	10%

[1] M: Percentage of member samples; NM: Percentage of non-member samples.

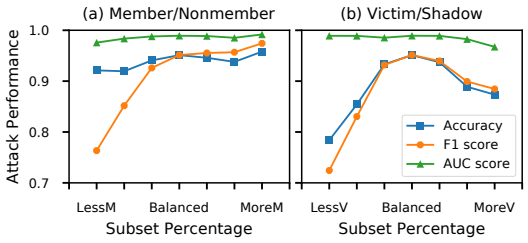


Fig. 7. Evaluation of the attack performance in different dataset sizes.

the patch number increases to 5 or 7, the attack performance does not change much, which even begins to drop slightly in patch numbers of 9 and 11 in translation, hue or saturation. The result indicates that cropped patches with 5 or 7 are enough to represent membership information in one image, which is similar to the conclusion in previous research.

6.5 Attack Experiment 4: Different Datasets for Dependent Attacks

In this experiment, we evaluate the performance of the proposed framework under various datasets as well as their sizes for dependent attacks. The following tests utilize PSPNet as the victim/shadow model, and Cityscapes as the dataset. Translation augmentation, 1Hot, and Slide are chosen as the data representation procedures to collect the performance results.

First, we evenly divide the dataset into two subsets as the victim and shadow datasets, and then we divide the member and non-member subsets unevenly. Some subset percentage cases have been tested:

- Several cases with fewer member samples denoted as LessM;
- A case with balanced member samples and non-member samples denoted as Balanced;
- Several cases with more member samples denoted as MoreM.

Please refer to the dataset split setting 1 in Table 3 for more details. Figure 7 (a) demonstrates the attack performance, indicating that most of the tests can achieve high accuracy and AUC scores over 0.9, and F1 scores increase from around 0.75 to around 0.95. This means that the membership percentage in the dataset has little impact on attack accuracy. As the F1 score is calculated using precision and recall, it is reasonable to observe a higher attack performance with more positive data (member samples).

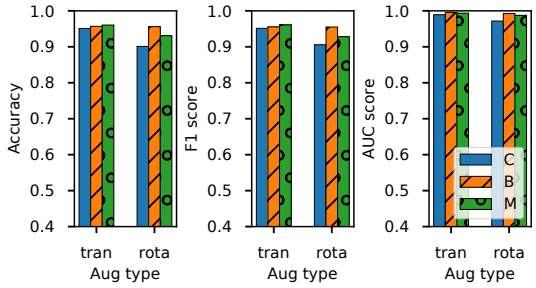


Fig. 8. Evaluation of the attack performance in different datasets. C: Cityscapes; B: BDD100K; M: Mapillary

Second, we evenly divide the dataset into two subsets as the member and non-member datasets, and then we split the victim and shadow subsets unevenly. Some subset percentage cases have been tested:

- Several cases with fewer victim subsets denoted as LessV;
- A case with balanced victim and shadow subsets denoted as Balanced;
- Several cases with more victim subsets denoted as MoreV.

Please refer to the dataset split setting 2 in Table 3 for more details. Figure 7 (b) displays the attack performance, showing that the balanced setting yields the best attack performance with high accuracy scores around 0.95. The accuracy scores drop below 0.9 when the attacks have more victim or shadow subsets.

Next, we compare the dependent attack performance using various datasets. The experiments are conducted using Cityscapes, BDD100K, and Mapillary, and each of these datasets is divided evenly into four subsets as victim member, victim non-member, shadow member, and shadow non-member subsets. Other experiment settings are 1Hot and Slide strategies with translation and rotation data augmentations, and PSPNet is used as victim and shadow models. Figure 8 shows that the attack performance of these datasets all have accuracy, F1 and AUC scores of around 0.9, and the attacks with BDD100K and Mapillary can achieve higher scores. Although these tests are under various subset settings, they exhibit that the larger dataset the network is trained upon, the better generalization ability the attack model can get. The attacks with Cityscapes already achieves a relatively high accuracy, but the attacks with BDD100K and Mapillary can further increase the performance by several percentages.

6.6 Attack Experiment 5: Dependent Attacks With or Without Data Representation Procedures

In our fifth experiment, we provide an evaluation on the dependent attacks with or without any one of the three data representation procedures. The experiment is based on PSPNet as the victim/shadow model and Cityscapes as the dataset (divided evenly). In Figure 9, we show the evaluation results of the attacks with or without data augmentations, prediction-label concatenation, and patch cropping strategies. The default strategies of the three data

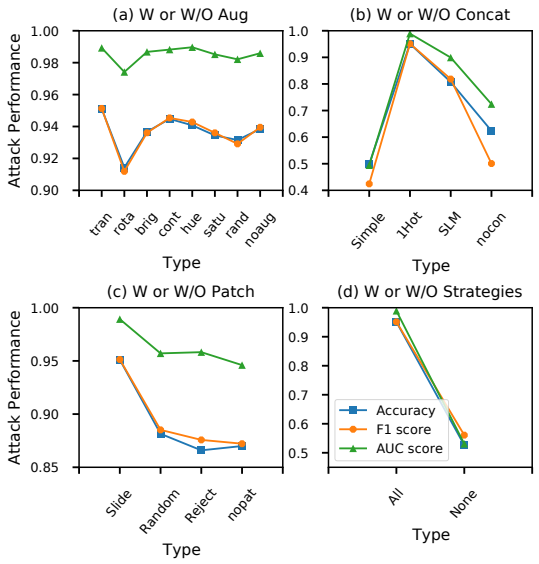


Fig. 9. Evaluation of the attack performance with/without data representation procedures.

representation procedures are translation, 1Hot and Slide, respectively, when the procedures are not turned off.

In Figure 9 (a), we evaluate the attacks with or without data augmentations. 1Hot prediction-label concatenation and Slide patch cropping are chosen as the default strategies. The attack with translation augmentations exhibits the best performance. Surprisingly, the attack without augmentations (noaug) does not have the worst performance. This indicates that some data augmentations are not very effective in the attack process. We would like to analyze this phenomenon in our future research.

In Figure 9 (b), we test the attacks with or without prediction-label concatenation. Translation augmentation and Slide patch cropping are chosen as default. The attack with 1Hot strategy achieves the highest accuracy, whereas the attack without concatenation (nocon) only gets accuracy scores of around 0.6. The results show that the attacks with 1Hot and SLM strategies significantly improve the attack performance.

In Figure 9 (c), the attacks with or without patch cropping are tested. Translation and 1Hot are the default strategies. We discover that the attack with Slide strategy has the best performance, while the attack without patch cropping (nopa) demonstrates the worst performance. The Random and Reject strategies can increase the attack performance to several percentages, but the Slide strategy is extremely effective in this label-only attacks.

We also provide the results of the attacks with or without all three data representation procedures (Figure 9 (d)). The attack with all three strategies (translation, 1Hot, and Slide) increases the performance from around 0.55 to 0.95, indicating that all these data representation procedures can contribute to the final result and enable the attack framework to successfully launching the label-only membership inference attacks.

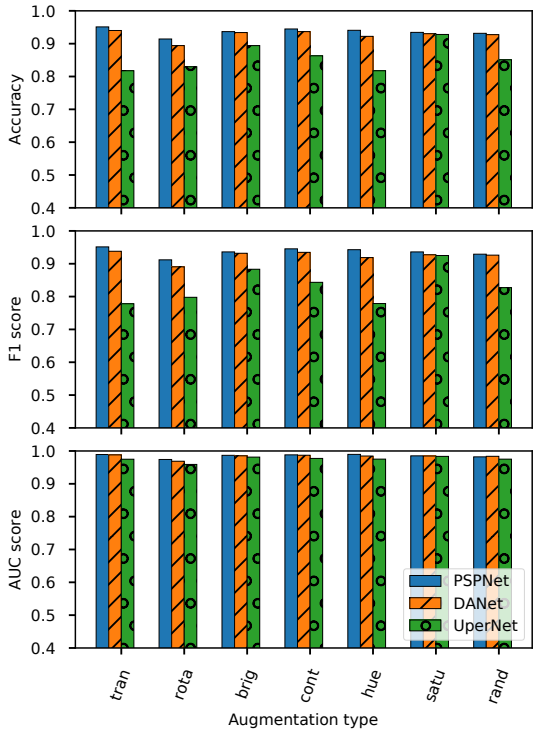


Fig. 10. Evaluation of the attack performance in different shadow models.

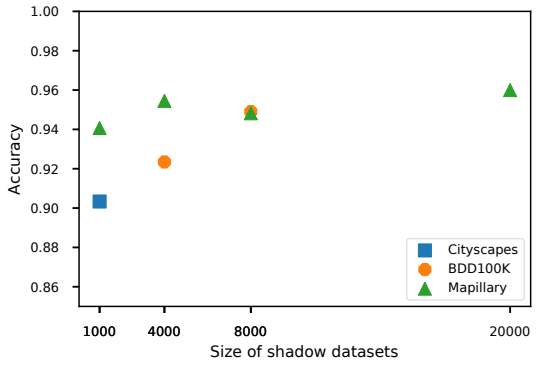


Fig. 11. Evaluation of the attack performance in independent attacks.

6.7 Attack Experiment 6: Different Attack Settings

In this experiment, we evaluate our framework with independent attacks. Figure 10 provides an evaluation of the attacks with various shadow models, including PSPNet, DANet, and UperNet. We still use PSPNet as the victim model, Cityscapes as the dataset, and ResNet-50 as the attack model. We compare the attack performance with 1Hot and Slide strategies with various data augmentations. The attacks using PSPNet as the shadow model have the best performance, all of which values are above 0.9 in terms of attack accuracy, F1 and AUC scores. The reason is that the more consistency between the shadow model and the attack model, the easier it is to mimic the attack model, resulting in a better performance.

Next, we provide an evaluation of our framework in independent attacks using various datasets as well as with different dataset sizes. Figure 11 shows statistics of the

TABLE 4
Attack Performance Comparison

Methods	Dependent PSP - PSP F1	Dependent PSP - PSP AUC	Independent Deep - PSP F1	Independent Deep - PSP AUC
ML-leaks (learning-based)	0.772	0.672	0.924	0.635
ML-leaks (learning-free)	0.774	0.620	0.923	0.634
Segmentations-leak (SLM, Random)	0.848	0.846	0.957	0.908
Segmentations-leak (SLM, Reject)	0.867	0.871	0.959	0.911
Ours (Tran, SLM, Random)	0.869	0.887	0.959	0.907
Ours (Tran, 1Hot, Slide)	**0.927**	**0.979**	**0.977**	**0.976**

[1] Attack performance with previous research of dependent and independent attacks. The best is marked bold.

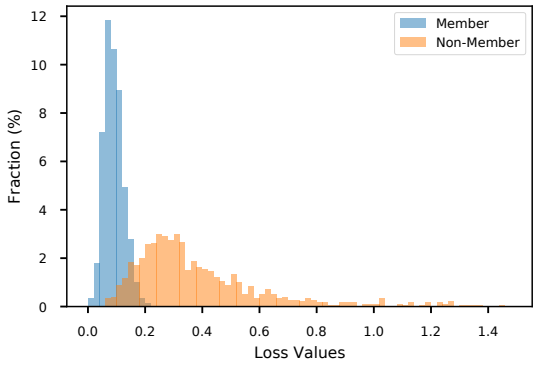


Fig. 12. A histogram of cross entropy loss values of member and non-member samples for the victim model.

attack performance in three different datasets as the shadow datasets: Cityscapes, BDD100K, and Mapillary. The size of the shadow datasets varies from 1,000 to 20,000. The test of Cityscapes is a dependent attack and the others are independent attacks. The other experiment settings include Cityscapes as the victim model, PSPNet as the victim model, and Deeplabv3+ as the shadow model. We choose translation, 1Hot and Slide for data representation procedures. Generally, we do not compare performance among different datasets. Instead, we focus on the investigation on the effect of dataset sizes on the independent attacks.

The results show the relationship between the dataset size and the attack accuracy. As the dataset size increases, the attack accuracy grows. The attacks using Mapillary outperform the others in most cases. When the whole Mapillary dataset is used as the shadow dataset (20,000 samples), the best attack performance can be achieved with an attack accuracy of over 0.95.

6.8 Comparison with Previous Research

We evaluate our experimental results (F1 scores and AUC scores) with ML-leaks [6] and Segmentations-leak [15] in Table 4. In dependent attacks, the best attacking scheme is our translation method with an F1 score of 0.869 and an AUC score of 0.887. In independent attacks, we illustrate our results in Mapillary Vistas. Our translation method and Segmentations-leak method have the equally best F1 score around 0.959. Our methods in the other settings also achieve outstanding performance.

This comparison with previous research means that our membership inference attack with limited knowledge (label-only) can achieve similar performance compared with the state-of-the-art results [15]. The major difference between [15] and our paper is that a stronger threat model (the attacker has access to the model's prediction scores) is needed in [15], while the assumption in our framework is relaxed to obtaining only the prediction labels. In particular, with only prediction labels (hard values) provided to the adversary, the information of classification likelihood/confidence is obscured, making it challenging to differentiate the non-member samples and the member samples used in training. In other words, it is common for a member sample and a non-member one to have exactly the same one-hot encoded vector as the neural network output, which makes them indistinguishable. Our framework can launch the attack successfully with the adoption of three data representation procedures (data augmentations, prediction-label concatenation, and patch cropping). This shows that the combination of the three data representation procedures is extremely helpful to extract membership signals from data samples.

6.9 Discussion on Attack Methods

The reasons why general membership inference attacks work are two-fold: the overfitting property of deep learning models and data distribution of the model prediction results.

First, the overfitting issue of deep learning models causing membership inference attacks is widely discussed in prior research [6], [7], [14]. A deep learning model usually performs much better on the training dataset than on the testing dataset. Yeom *et al.* [7] demonstrated that overfitting was sufficient for an attacker to perform an attack. Salem *et al.* [6] discovered that a more overfitted deep learning model would be more vulnerable to membership inference attacks. Chen *et al.* [14] discussed that overfitting also caused membership inference attacks on generative adversarial networks.

Second, the model prediction results show different data distributions. Figure 12 shows a histogram of cross entropy loss values of member and non-member samples for the decoder loss of the victim model PSPNet in our experiment. We can see that member samples have small and concentrated loss values, while non-member samples have larger and more widely-spread loss values. The deep learning model is trained by member samples, leading to two different data distributions of loss values. Membership inference attacks can leverage this difference to extract membership signals.

Our label-only attack framework in semantic segmentation models is a bit different. The weak membership signals in a single pixel and the limited information from prediction results make our attack framework harder than general attacks. With three data representation procedures being processed (data augmentation, prediction-label concatenation, and patch cropping), our label-only attack framework is effective in semantic segmentation. We have analyzed our attack framework in the above experiments, which shows our membership inference attacks can successfully extract private information from the victim model.

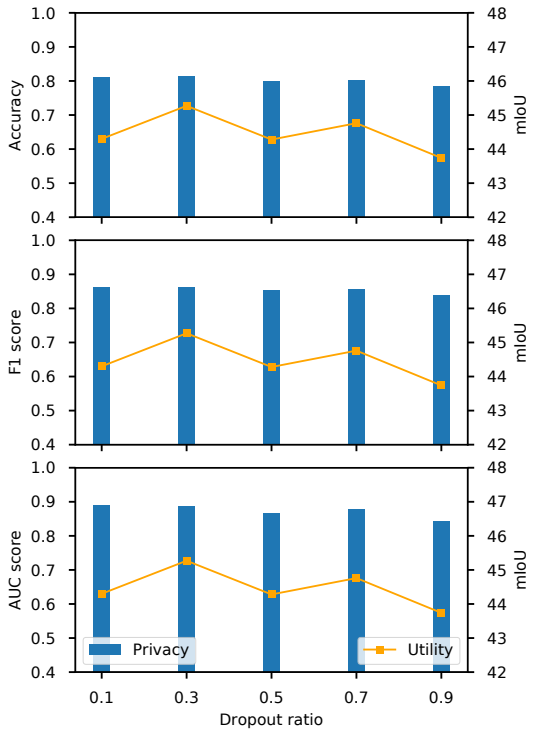


Fig. 13. Evaluation of the defense performance using dropout.

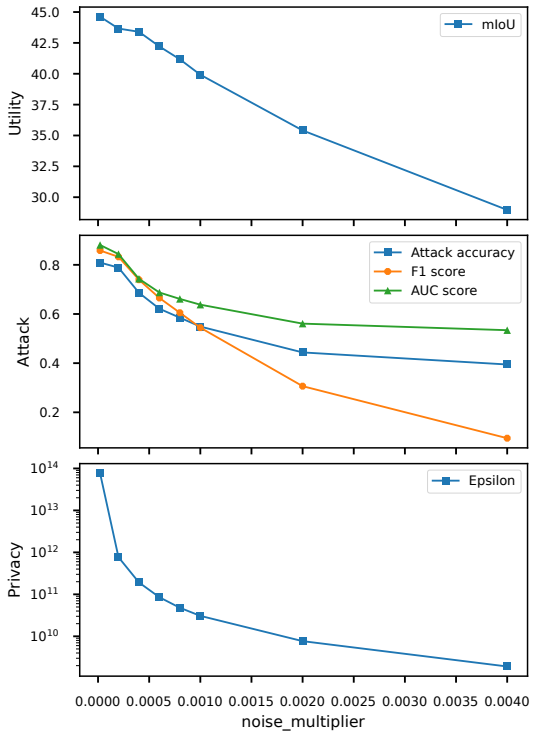


Fig. 14. Evaluation of the defense performance using DPSGD.

6.10 Defense Experiment 1: Dropout

In our first defense experiment, we evaluate the defense performance using dropout. As our task is for semantic segmentation models, we have a dropout layer before the final classification layer in the victim model. We set a dropout ratio (ranging from 0 and 1) for the dropout layer and activate it during training. In our experiment, the dropout ratio is set to 0.1 (default), 0.3, 0.5, 0.7, 0.9. The other settings are similar to prior dependent attacks. We apply translation data augmentation, SLM concatenation and Slide patch-cropping strategies when preparing the dataset of the attack model.

The experiment result is shown in Figure 13. We can see that as the dropout ratio increases, the utility performance only decreases a little (from 44.3% to 43.74%), and the privacy performance also decreases a little (attack accuracy from 0.81 to 0.78). And they do not decrease monotonously. As a result, we can conclude that dropout is not an effective defense mechanism against the proposed label-only membership inference attacks in semantic segmentation models.

6.11 Defense Experiment 2: DPSGD

In our second defense experiment, the defense performance using DPSGD is evaluated. We apply DPSGD in the victim model in dependent label-only attacks using PSPNet and Cityscapes dataset. We adopt translation data augmentation, SLM concatenation and Slide patch-cropping strategies when processing data representation. We set the gradient norm bound C to 48.0 based on the gradient norm distribution and δ to 6×10^{-4} based on the size of the victim dataset. We evaluate the utility and privacy performance under different noise multipliers (the squared Gaussian noise scale σ).

From the experiment result in Figure 14, we can see that as the noise multiplier increases, both the utility and privacy performance decrease. When the noise multiplier ranges from 0 to 0.004, the utility performance (mIoU) decreases by over 15%, and the privacy performance (attack accuracy) decreases from above 0.8 to around 0.4. A relatively balanced result is when the noise multiplier is 0.0008, the mIoU drops 3%, and the attack accuracy reaches 0.58. The differential privacy budget ϵ in this experiment is tremendously large, ranges from 10^9 to 10^{13}. Theoretically, large ϵ values can not provide meaningful differential privacy guarantees. This means that when DPSGD is used in a large deep learning model (Resnet is often selected as the backbone in semantic segmentation models with large numbers of layers), a very small noise added in the model gradients can cause the membership inference attacks to fail. The work of [15], [31] draw similar conclusions when DPSGD is adopted to defend the membership inference attacks.

6.12 Discussion on Defense Mechanisms

We discuss and evaluate the defense performance against the proposed label-only attacks using dropout and differential privacy. The dropout strategy has little effect on our attacks, which fails to reduce the attack accuracy with any dropout ratio. Although the DPSGD strategy is more effective than the dropout strategy on our attacks, it reduces both mIoU and the attack accuracy. With the precise tuning of the DPSGD hyper-parameters, we manage to find a balanced result with the lowered attack accuracy and a reasonable mIoU. The balanced result is not ideal, proving that the defense mechanism against our proposed attack method can only be achieved at the cost of learning degradation, which

is the crucial reason of the success of our proposed attack method.

Defense mechanisms can protect the victim model from attacks, but they also jeopardize the model's performance. In this light, we should be aware of the privacy-utility trade-off when training models with defense mechanisms.

7 Conclusion

In this paper, we have established a well-designed framework for label-only membership inference attacks against semantic segmentation models. We apply different data augmentations to the original data and design the data representation procedures to generate datasets for the attack model. Our ablation analysis was conducted under various experimental settings. We conclude that with seven various data augmentations, several prediction-label concatenations, and patch cropping strategies, the label-only membership inference attacks can achieve a competitive performance compared to the previous work. We also demonstrate that the label-only attacks can be extended to other popular computer vision tasks such as semantic segmentation. Our future work is to evaluate our attack methods with more defense mechanisms and analyze more about the effect of data augmentations on membership inference attacks.

Acknowledgments

This work is supported by two ARC Projects (LP180101150 and DP190100981) from the Australian Research Council, Australia.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1106–1114.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*. IEEE Computer Society, 2016, pp. 770–778.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*. IEEE Computer Society, 2015, pp. 3431–3440.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *SP*. IEEE Computer Society, 2017, pp. 3–18.

[5] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.

[6] 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 *NDSS*. The Internet Society, 2019.

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

[8] G. Zhang, B. Liu, T. Zhu, A. Zhou, and W. Zhou, "Visual privacy attacks and defenses in deep learning: a survey," *Artificial Intelligence Review*, pp. 1–55, 2022.

[9] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

[10] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "MemGuard: Defending against black-box membership inference attacks via adversarial examples," in *CCS*. ACM, 2019, pp. 259–274.

[11] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," *CoRR*, vol. abs/2005.03915, 2020.

[12] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-Only Membership Inference Attacks," in *ICML*. PMLR, 2021, pp. 1964–1974.

[13] Z. Li and Y. Zhang, "Membership Leakage in Label-Only Exposures," in *CCS*. ACM, 2021.

[14] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "GAN-Leaks: A taxonomy of membership inference attacks against generative models," in *CCS*. ACM, 2020, pp. 343–362.

[15] Y. He, S. Rahimian, B. Schiele, and M. Fritz, "Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 12368. Springer, 2020, pp. 519–535.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*. IEEE Computer Society, 2016, pp. 3213–3223.

[17] Y. Zhang, H. Chen, Y. He, M. Ye, X. Cai, and D. Zhang, "Road segmentation for all-day outdoor robot navigation," *Neurocomputing*, vol. 314, pp. 316–325, 2018.

[18] S. Reiss, C. Seibold, A. Freytag, E. Rodner, and R. Stiefelhagen, "Every annotation counts: Multi-label deep supervision for medical image segmentation," in *CVPR*, 2021, pp. 9532–9542.

[19] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*. IEEE Computer Society, 2017, pp. 6230–6239.

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 11211. Springer, 2018, pp. 833–851.

[23] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *CVPR*. IEEE Computer Society, 2016, pp. 3640–3649.

[24] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 11209. Springer, 2018, pp. 432–448.

[25] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual Attention Network for Scene Segmentation," in *CVPR*. IEEE Computer Society, 2019, pp. 3141–3149.

[26] Y. Kaya and T. Dumitras, "When Does Data Augmentation Help With Membership Inference Attacks?" in *ICML*. PMLR, 2021, pp. 5345–5355.

[27] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, "How Does Data Augmentation Affect Privacy in Machine Learning?" *AAAI*, vol. 35, no. 12, pp. 10 746–10 753, 2021.

[28] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," in *NDSS*. The Internet Society, 2021.

[29] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, "ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models," *CoRR*, vol. abs/2102.02551, 2021.

[30] S. Rezaei and X. Liu, "On the Difficulty of Membership Inference Attacks," in *CVPR*, 2021, pp. 7892–7900.

[31] Y. Park and M. Kang, "Membership inference attacks against object detection models," *CoRR*, vol. abs/2001.04011, 2020.

[32] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "LOGAN: Membership inference attacks against generative models," *PoPETs*, vol. 2019, pp. 133–152, 2019.

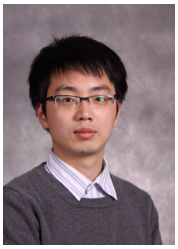
[33] Y. Zou, Z. Zhang, M. Backes, and Y. Zhang, "Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning," *CoRR*, vol. abs/2009.04872, 2020.

[34] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*. ACM, 2016, pp. 308–318.

[35] T. Zhu, D. Ye, W. Wang, W. Zhou, and P. Yu, "More than privacy: Applying differential privacy in key areas of artificial intelligence," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

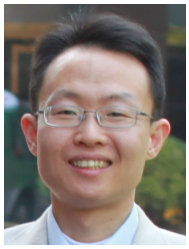
[36] Y. Zhao, B. Liu, T. Zhu, M. Ding, and W. Zhou, "Private-encoder: Enforcing privacy in latent space for human face images," *Concurrency and Computation: Practice and Experience*, p. e6548, 2022.
[37] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, 2019.
[38] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *ICLR*. OpenReview.net, 2018.
[39] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.
[40] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 4004. Springer, 2006, pp. 486–503.
[41] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*. IEEE, 2020, pp. 2633–2642.
[42] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *ICCV*. IEEE Computer Society, 2017, pp. 5000–5009.
[43] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.
[44] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, "A review on deep learning techniques applied to semantic segmentation," *CoRR*, vol. abs/1704.06857, 2017.

Guangsheng Zhang is currently working towards the Ph.D. degree in the School of Computer Science, University of Technology Sydney, Australia. He received the B.Eng. degree from Northeastern University, China in 2012, and the M.Sc. degree from Aberystwyth University, the UK in 2015. His research interests include data privacy, computer vision, and deep learning.



Bo Liu received the BEng degree from the Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004. He then received the MEng. and PhD. Degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively. He is currently a Senior Lecturer with the University of Technology Sydney, Australia. His research interests include cybersecurity and privacy, location privacy and image privacy, privacy protection and machine learning.

Tianqing Zhu is an associate professor in the School of Computer Science, University of Technology Sydney, Australia. She received the B.Eng. degree and M.Eng. degree from Wuhan University, Wuhan, China, in 2000 and 2004, respectively, and the Ph.D. degree from Deakin University, Australia, in 2014. She was a lecturer in the School of Information Technology, Deakin University, from 2014 to 2018. Her research interests include privacy preserving, cyber security and privacy in the artificial intelligence.



Ming Ding (M'12-SM'17) received the B.S. and M.S. degrees (with first-class Hons.) in electronics engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the Doctor of Philosophy (Ph.D.) degree in signal and information processing from SJTU, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked at Sharp Laboratories of China in Shanghai, China as a Researcher/Senior Researcher/Principal Researcher. Currently, he is a senior research scientist at Data61, CSIRO, in Sydney, NSW, Australia. His research interests include information technology, data privacy and security, and machine learning and AI. He has authored more than 150 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as a book "Multi-point Cooperative Communication Systems: Theory and Applications" (Springer, 2013). Also, he holds 21 US patents and has co-invented another 100+ patents on 4G/5G technologies. Currently, he is an editor of IEEE Transactions on Wireless Communications and IEEE Communications Surveys and Tutorials. Besides, he has served as a guest editor/co-chair/co-tutor/TPC member for multiple IEEE top-tier journals/conferences and received several awards for his research work and professional services.

Wanlei Zhou (Senior member, IEEE) is currently the Vice Rector (Academic Affairs) and Dean of Institute of Data Science, City University of Macau, Macao SAR, China. He received the B.Eng and M.Eng degrees from Harbin Institute of Technology, Harbin, China in 1982 and 1984, respectively, and the PhD degree from The Australian National University, Canberra, Australia, in 1991, all in Computer Science and Engineering. He also received a DSc degree (a higher Doctorate degree) from Deakin University in 2002. Before joining City University of Macau, Professor Zhou held various positions including the Head of School of Computer Science in University of Technology Sydney, Australia, the Alfred Deakin Professor, Chair of Information Technology, Associate Dean, and Head of School of Information Technology in Deakin University, Australia. Professor Zhou also served as a lecturer in University of Electronic Science and Technology of China, a system programmer in HP at Massachusetts, USA; a lecturer in Monash University, Melbourne, Australia; and a lecturer in National University of Singapore, Singapore. His main research interests include security, privacy, and distributed computing. Professor Zhou has published more than 400 papers in refereed international journals and refereed international conferences proceedings, including many articles in IEEE transactions and journals.