# Output regeneration defense against membership inference attacks for protecting data privacy

Yong Ding
*Guangxi Key Laboratory of Cryptography and Information Security, Guilin, China; School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China and Department of New Networks, Peng Cheng Laboratory, Shenzhen, China*

Peixiong Huang
*School of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin, China*

Hai Liang
*School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China and Guangxi Key Laboratory of Cryptography and Information Security, Guilin, China*

Fang Yuan
*Communication Center of the Ministry of Foreign Affairs, Beijing, China, and*

Huiyong Wang
*School of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin, China*

Received 22 March 2023
Revised 12 April 2023
3 May 2023
Accepted 8 May 2023

## Abstract

**Purpose** – Recently, deep learning (DL) has been widely applied in various aspects of human endeavors. However, studies have shown that DL models may also be a primary cause of data leakage, which raises new data privacy concerns. Membership inference attacks (MIAs) are prominent threats to user privacy from DL model training data, as attackers investigate whether specific data samples exist in the training data of a target model. Therefore, the aim of this study is to develop a method for defending against MIAs and protecting data privacy.

**Design/methodology/approach** – One possible solution is to propose an MIA defense method that involves adjusting the model's output by mapping the output to a distribution with equal probability density.

This article is supported in part by the National Key R&D Program of China under project (2020YFB1006003), the National Natural Science Foundation of China under project (62172119), the Guangxi Natural Science Foundation under grant (2019GXNSFGA245004) and the Major Key Project of PCL under grants (PCL2021A09, PCL2021A02, PCL2022A03).

International Journal of Web Information Systems
Vol. 19 No. 2, 2023
pp. 61-79
Emerald Publishing Limited
1744-0084
DOI 10.1108/IJWIS-03-2023-0050

This approach effectively preserves the accuracy of classification predictions while simultaneously preventing attackers from identifying the training data.

**Findings** – Experiments demonstrate that the proposed defense method is effective in reducing the classification accuracy of MIAs to below 50%. Because MIAs are viewed as a binary classification model, the proposed method effectively prevents privacy leakage and improves data privacy protection.

**Research limitations/implications** – The method is only designed to defend against MIA in black-box classification models.

**Originality/value** – The proposed MIA defense method is effective and has a low cost. Therefore, the method enables us to protect data privacy without incurring significant additional expenses.

**Keywords** Artificial intelligence security, Deep learning, Privacy protection, Membership inference attack

**Paper type** Research paper

## 1. Introduction

With the rapid advances in machine learning (ML), artificial intelligence is continually enhancing and surpassing human performance in various endeavors (Lewis *et al.*, 2020). Deep learning (DL), which has been extensively used in daily life (Tong *et al.*, 2020), has consequently attracted model-related privacy concerns (Gao *et al.*, 2022). The data used for training DL models may contain users' private information, such as human faces in image data or the content of conversations in voice data. Disclosure of such private information seriously violates user privacy. Moreover, with the application of DL to medical therapy (Pandey and Janghel, 2021), patients may face threats because of the leakage of medical information (Santhi and Saradhi, 2021). Some DL models are trained on data monopolized by firms rather than publicly available data, and disclosure of these data may result in significant losses for the firms. Membership inference attacks (MIAs) aim to determine whether given sample data exist in the training data of a target model by analyzing the target model, which may raise severe privacy issues in the training data of the model (Gao, 2022; Hu *et al.*, 2021).

The MIA task is regarded as a binary classification problem (Chen *et al.*, 2020). Specifically, the attacker aims to construct an attack model that captures the discrepancies between the training and non-training sets of the target model on the target domain. By leveraging such differences, the attacker infers whether a given sample belongs to the training set of the target model, thereby compromising the privacy of the training data.

Many defense methods are available for addressing MIAs. Examples include various regularization techniques (Shokri *et al.*, 2017; Salem *et al.*, 2019; Nasr *et al.*, 2018) used for reducing the difference between training set outputs and non-training set outputs. These methods improve the performance of the target model to some extent. However, they have limited effectiveness in defending against MIAs. When the model is highly overfitted, the attack still maintains a relatively high accuracy. Jia *et al.* (2019) proposed the MemGuard, which adds specially crafted noise to the output of a model to make it difficult to distinguish between the training set output and the non-training set output. While this method effectively defends against MIA without compromising the performance of the target model, the cost is relatively high because of the need for repeated iterations to determine the noise.

This study aimed to mitigate the risk of MIAs and enhance the privacy of the target model training set by reducing the dissimilarities between the training and non-training sets. The defense method of adjusting the model output, which is similar to that used in MemGuard, was adopted. To address the problem of excessive noise production in the MemGuard and further enhance the privacy security of DL models, a generating model-based output adjustment defense method was developed. The study is summarized as follows:

1. Two output reconstruction methods were designed: a non-member mask and a member mask method:

(1) Multiple classification models were trained on different data sets as target models, and the proposed defense method was evaluated on all models. The results indicated that the defense had a minor impact on the models, with most experiencing a change in accuracy of no more than 2%.

(2) The defense method was compared with two other defense methods, and it was found that the proposed defense method reduced the attack success rate by at least 9.2%. Additionally, after deploying the defense, the highest attack accuracy was 59.5%, a decrease of 38.88% compared with the accuracy of attacks with no defense.

The remainder of this paper is structured as follows. Section 2 presents a review of related work, and Section 3 provides a detailed explanation of the proposed method. The experimental results are presented in Section 4, while Section 5 summarizes the findings of the study.

## 2. Related work

### *2.1 Member inference attack*

Since the introduction of the concept of MIAs targeting DL in 2017, various types of such attacks have emerged. These include shadow model-based attacks, binary comparison-based attacks and difference comparison-based attacks. This section provides a comprehensive introduction to these methods.

*2.1.1 Shadow model-based attack.* In the earliest shadow model-based attack, multiple shadow models are used to replace the target model and extract effective information, as proposed by Shokri *et al.* (2017). The attacker prepares $n$ shadow training and testing sets and uses the $n$ training sets to train multiple shadow models similar to the target model. The trained shadow models are then used to output their corresponding shadow training and testing set prediction results. These outputs are labeled to produce a data set containing membership information, which is used to train an attack model. The attack model is then used to classify the output of a test sample on the target model to determine its membership status. However, this method requires training a large number of shadow models, which is costly. Therefore, subsequent research (Salem *et al.*, 2019) has focused on reducing the number of shadow models to lower the cost of the attack. Another significant drawback of this method is that the performance of the attack model is closely related to that of the shadow model, which is often significantly different from the target model.

*2.1.2 Binary comparison-based attack.* To avoid the attack-prevention limitations of the shadow model and further reduce the cost of constructing numerous shadow models, Salem *et al.* (2019) proposed a novel MIA method that does not rely on the shadow model. The method uses binary comparison and maximum posterior probability to infer the membership of a sample. The attacker repeatedly queries the target model with numerous samples, extracts the threshold and uses it to determine if a sample belongs to the training set of the target model. However, the simplicity of the attack method results in a significant loss of high-dimensional features from the output of the target model. Moreover, the success of the attack is heavily reliant on the selected threshold.

*2.1.3 Difference comparison-based attack.* To eliminate the dependence on attack model thresholds, Hu *et al.* (2021) introduced a new MIA called BLINDMI, which does not require a shadow model. In this method, a non-training set is constructed, and samples are moved between the target sample set and the non-training set. By measuring the change in distance between the target sample set and the non-training set before and after the sample is moved, the method determines whether the moved sample belongs to the training set of the target model. However, this approach also relies on the non-training set.

### *2.2 Membership inference attack defense*

The training of the defense model is similar to that of a previous model (Gao *et al.*, 2022), which makes it difficult to fundamentally separate the member data. Fortunately, the attacker cannot directly use the information between the training set and the model. MIA distinguishes members from non-members through the difference between members and non-members in the target model. The corresponding defense methods narrow this gap, making it difficult to distinguish between members and non-members. Existing defense methods against MIAs roughly divide into defense against the generalization capability of the model, defense against the MIA using knowledge transfer and defense against fitting the model output:

- *Enhancing Model Generalization*: L2-regularization by Shokri *et al.* (2017) uses the L2 term to punish large parameters, decreasing the difference between the members and non-members of the model. The adversarial regularization proposed by Nasr *et al.* (2018) regularizes the target model by training an additional attack model. The dropout regularization of Salem *et al.* (Jia *et al.*, 2019) enhances the generalization ability of the model. Such methods often fail to eliminate model overfitting. Under these defenses, a poor model generalization ability does not make the attack any less potent.
- *Knowledge Transfer*: Examples of these defenses include the private aggregation of teacher ensembles, proposed by Nicolas *et al.* (2017), and the distillation for membership privacy, proposed by Shejwalkar and Houmansadr (2021). Here, the target model is transferred to the defense model by distillation to defend against the MIA. This method effectively prevents the model from being attacked by the MIA; however, the resultant defense model may not completely substitute the target model, and the defense considerably influences the target model.
- *Adding disturbance*: MemGuard, proposed by Jia *et al.* (Yeom *et al.*, 2018), adds noise to the model output to reduce the discrepancy between members and non-members. This method effectively defends against some early privacy attacks and maintains the output accuracy of the target model. However, the cost of noise generation is relatively high. Our method also aims to adjust the output of the model, but it uses the generative model to manage all the sample outputs in a unified way, which considerably improves defense efficiency. The experimental results demonstrate that our approach is also more effective in defending against the latest attacks.

## 3. Output regeneration method

### *3.1 Security threats and design objectives*

Because training data are an indispensable component of the ML model, and these data may contain the user's privacy information, in the application phase of the target model, MIA steals user privacy by extracting the target model information to determine the member membership. This attack is regarded as a binary classification task, that is, for a trained ML model S, the training data are denoted as $X_{train}$ (member data). The attacker uses the attack model $\mathcal{M}$ to determine whether the given sample x belongs to a specific group $X_{train}$, as represented by equation (1), where x represents the sample to be assessed, S is the target model, 0 indicates that sample $x$ is not in the member data of the target model and 1 indicates that sample x is in the member data of the target model:

$$\mathcal{M}\left(\mathrm{x}, \mathrm{S}(\mathrm{x})\right) = \{0, 1\} \tag{1}$$

In the case of the classification model, if the model is overfitted, the difference between the member and non-member data will be large. This difference is described as two problems. First, the probability distribution of the member data will be concentrated (maximum probability value close to 1, others close to 0), unlike that of the non-member data. Second, the performance of the member data will be significantly better than that of the non-member data.

This problem exists because the target model extracts redundant features during repeated training. Taking the classification model with SoftMax output as an example, the ground truth used for training is a vector with a real class of 1 and a rest of 0. When the model loss is minimal, the output belonging to the training data of the model is extremely close to the ground truth, which may be determined by the desired characteristics or other characteristics. These extra features come only from member data. Eventually, the difference is better represented by the target model output. As noted earlier, the attacker determines membership in the target sample by exploiting the large discrepancy between the output of the training data and non-training data. To minimize this discrepancy and prevent the target model from being attacked by membership inference, a defense method based on tuning the model output is proposed in this study. The method keeps the relative output size constant, that is, $Y = (y_1, y_2, y_3,\ldots y_n)$ is the output probability of the target model, and $Y' = \left(y_1', y_2', y_3' \ldots, y_n'\right)$ is the output probability of the generative model. Every $y_i > y_j$ guarantees $y_{\mathrm{i}}' > y_{\mathrm{j}}'$; compared with the output of the target model, the adjusted output probability does not change the classification result of the model, which ensures that the adjusted model is equivalent to the target model in use. The method is detailed in the following subsections. Refer to Table 1 for a summary of the notations used in the text. This table provides a comprehensive overview of the symbols and their corresponding meanings.

### *3.2 Overview of generating models*

The main structure of the approach is depicted in Figure 1, where X is the input of the target model, $\mathrm{S_m}$ is the target model, Y is the target model output, G is the generator and Y′ is the

| Symbol | Meaning |
|---|---|
| $S$ | target model classifier |
| $X_{train}$ | training data of target model (member data) |
| $X_{non}$ | non-training data of target model (non-member data) |
| $Y$ | target model outputs |
| $Y_{train}$ | target model outputs on the member data |
| $Y^{l}_{train}$ | labels of $Y_{train}$ used to train the generative model |
| $Y_{non}$ | target model outputs on the non-member data |
| $Y^{l}_{non}$ | labels of $Y_{non}$ used to train the generative model |
| $D_{train}$ | training data for generator |
| $G$ | generator |
| $Y'$ | probability distribution vectors generated by generator |

**Source:** Authors' own design

**Table 1.** Symbols used in the text

output when adjusted by the generator. Keeping the target model $S_m$ and the input $X$ unchanged, the output of the target model is taken as input to the generative model. The generative model was used to adjust the output probabilities of the target model. Furthermore, the SoftMax function was enabled for the output layer of the generative model to ensure that the generative model output remains consistent with the target model.

The output probability distribution structure of the classification model is straightforward. The role of the generative model is to adjust the target model output to the same distribution while maintaining the relative probability size. Owing to the simplicity of this task, the trained generative model has a high generalization ability, and its output is resistant to MIAs. The effectiveness of the generative model in achieving good results without the need for complex neural network models was demonstrated experimentally, where only a simple multi-layer perceptron with fully connected layers (Figure 2) was trained as the output generator. The generative model first projects its input (i.e. the target model output) to a higher dimension to capture its high-dimensional features, and then it gradually restores the high-dimensional transformation to the same dimension as that of the input through two fully connected neural network layers. Because the generative model output satisfies the same distribution, and the difference between member and non-member data is small, the model is better equipped to defend against MIAs after being processed by the generative model. Additionally, to verify the strong transferability of the generative model, the same generative model was used to process data with the same number of categories in subsequent experiments.

### *3.3 Generative model training*

To preserve the original model's classification outcomes, the generative model is trained independently of the target model. Instead, the target model's output results serve as the training data for the generative model. Assuming the model owner is the defender, they have full access to the training set data and easily augment the volume of the non-training set data for the model.

This paper introduces two strategies, namely, member mask and non-member mask, to obfuscate the distinction between member and non-member data based on the output distribution characteristics of the generative model's task. The primary distinction between the two approaches lies in the labels used for the generative model during supervised learning.

*3.3.1 Member mask.* The member mask is used to mask non-member output as member output. Figure 3 depicts the implementation process of this method. As mentioned above, the defender has all the member data $X_{train}$ for the target model, meaning it easily obtains a non-member data $X_{non}$. The output probabilities $Y$ are obtained using the target model. The member data $X_{train}$ and non-member data $X_{non}$ are processed separately according to their respective characteristics.

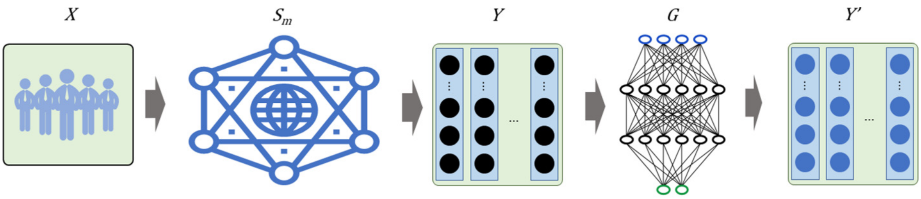
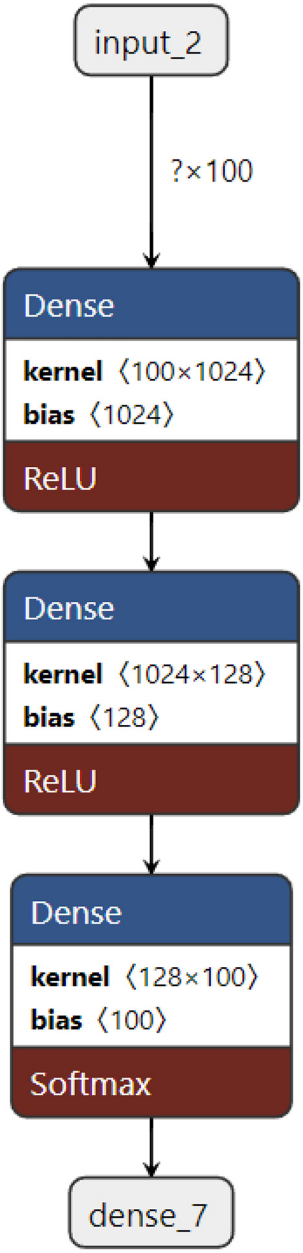


**Figure 1.**
Schematic diagram of output regeneration

**Source:** Author's own design



**Source:** Author's own design

**Figure 2.**
Generative model used for CIFAR-100

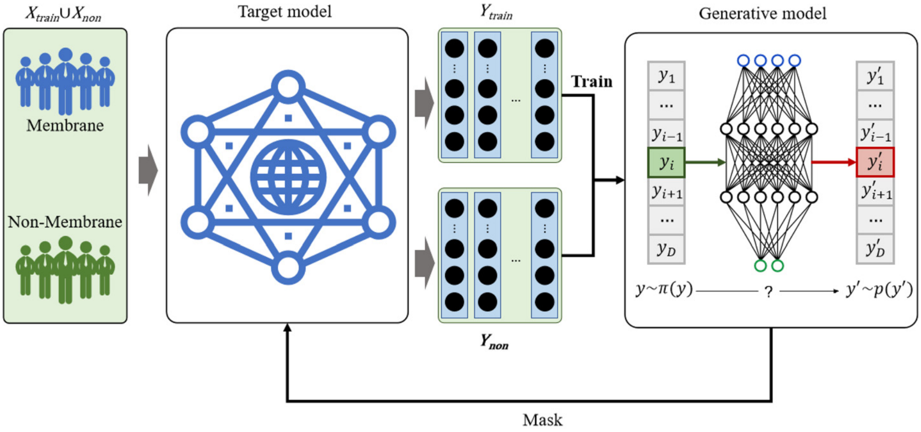


**Figure 3.** Member masquerade diagram

**Source:** Author's own design

As mentioned above, the output of the member data is extremely close to the ground truth set during the training of the target model. Therefore, a label similar to the target model ground truth is constructed for the member mask to train the generation model to simulate the member output, as expressed in equation (2):

$$y^l = \begin{cases} y; x \in X_{train} \\ max(y) = 1, other = 0; x \in X_{non} \end{cases} \tag{2}$$

As shown in equation (2) specifically, in the case of member data, the output is a probability distribution vector with a maximum probability value close to 1, while the remaining probabilities are close to 0. To preserve the maximum output information of the target model, the label for member data $Y^l_{train}$ is assigned as its output on the target model without any additional transformations during the generative model training process.

For non-member data, the maximum probability value output on the target model is set to 1, whereas the rest is set to 0, serving as the corresponding label $Y^l_{non}$ for the generative model. Subsequently, the output $Y$ of member data and non-member data on the target model is used as input for the generative model, which is then trained with the corresponding labels. Finally, the trained generative model is used to adjust the target model, with mean square error $loss = mse(Y, Y_1)$ set as the loss function for the member mask.

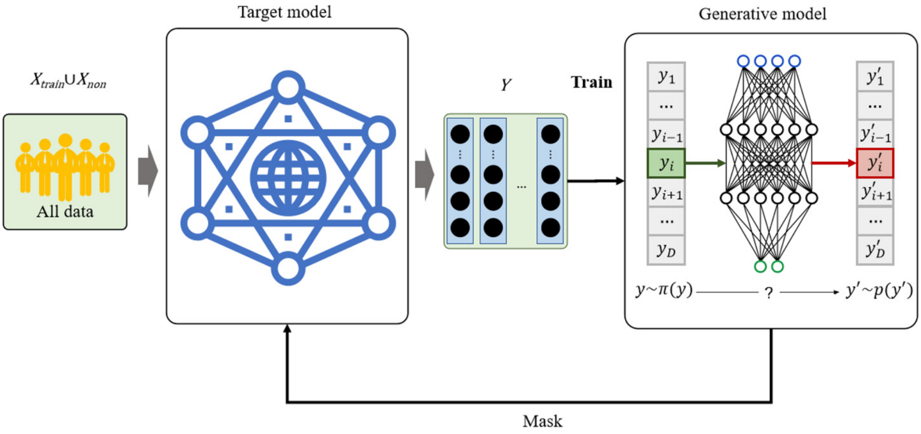
```
Algorithm 1: Training data for member mask
  Input: $Y$
  Output: $D_{train}$
    for each $y \in Y$ do
      if $y \in Y_{train}$ then
        $y^1 \leftarrow y$
      else if $y \in Y_{non}$ then
      $y^1 \leftarrow max(y) = 1, other = 0$
      $D_{train} \leftarrow D_{train} \cup \{(y, y^1)\}$
    return $D_{train}$
```

According to the description provided in Algorithm 1, the output $Y$ of the target model is used as input, and the output $Y$ and its corresponding label $Y^l$ are used as the training set $D_{train}$ for the generative model. For the output $y$ of all samples, if $y$ is the output of a member sample, the corresponding label is $y$ itself. On the contrary, if $y$ is a non-member output, the corresponding label is a vector of the same dimension, where the maximum value is 1 and the remaining values are 0. Finally, $y$ and the corresponding $y^l$ are added to the training set of the generative model.

*3.3.2 Non-member mask.* As shown in Figure 4, similar to the member mask, the non-member mask method starts with a data set that includes both member data $X_{train}$ and non-member data $X_{non}$. The probability distribution of the outputs $Y_{train}$ and $Y_{non}$ is obtained using the target model. Because the features outputted by non-members are generally more diverse than those outputted by members, and unlike the member mask, the non-member mask processes all outputs equally when creating labels. The specific method is as follows.

For all outputs $Y$, let $Y = Y_{train} \cup Y_{non}$. Rank the probability distributions in decreasing order of magnitude: $y_1 \geq y_2 \geq y_{3\ldots} \geq y_n$. Create labels $y^l = \left(y_1^l, y_2^l, y_3^l, \ldots, y_n^l\right)$, where $\frac{1}{k} < y_1^l < \frac{2}{k}$ simultaneously satisfies $y_1^l > y_2^l > y_3^l > \ldots > y_k^l$, $y_1^l + y_2^l + y_3^l + \ \ldots + y_k^l = 1$ and $y_i^l = 0$ when $i > k$. Use the labels to train the generative model to adjust the output of the target model. For the non-member mask, it is recommended to use cross-entropy $loss = CrossEntropyLoss(Y, Y^l)$ as the loss function.

As shown in Algorithm 2, similar to Algorithm 1, the output $Y$ of the target model is used as input, and the output $Y$ and its corresponding label $Y^l$ are used as the training set $D_{train}$ for the generative model. The specific procedure is as follows: to ensure that the labels conform to the same distribution, the first $k$ largest values of the labels, i.e. $y^l_{1\_row}, y^l_{2\_row}, y^l_{3\_row}, \ldots, y^l_{k\_row}$, are fixed. Then, the maximum value $y^l_{1\_row}$ is determined, satisfying $\frac{1}{k} < y^l_{1\_row} < \frac{2}{k}$, and the subsequent values are determined based on the previous value, ensuring that $y^l_{i\_row} < y^l_{i-1\_row}$. When $1 - \sum_{n=1}^{i-1} y^l_{n\_row} > y^l_{i-1\_row}$, $y^l_{i\_row} = 1 - \sum_{n=1}^{i-1} y^l_{n\_row}$ is set to ensure that $y_1^l + y_2^l + y_3^l + \ldots + y_k^l = 1$, and $k$ is determined at this point. Then, the



**Source:** Author's own design

**Figure 4.**
Non-member masquerade diagram

$k$ values are assigned to the negative values of the first $k$ largest values of $y$, i.e. $y_i^l = y_{i\_row}^l$ when $i \leq k$, and the rest are set to 0, as the labels of $y$. Finally, $y$ and the corresponding $y^l$ are added to the training set of the generative model:

```
    Algorithm 2: Training data for non-member mask
Input: $Y$
Output: $D_{train}$
  $y_{1\_row}^l \leftarrow random\left(\frac{1}{k}, \frac{2}{k}\right)$
  $i \leftarrow 2$
  while: true do
    if $1 - \sum_{n=1}^{i-1} y_{n\_row}^l > y_{i-1\_row}^l$ then
      $y_{i\_row}^l \leftarrow random\left(0, y_{i-1\_row}^l\right)$
      $i \leftarrow i + 1$
      continue
    else then
      $y_{i\_row}^l \leftarrow 1 - \sum_{n=1}^{i-1} y_{n\_row}^l$
      $k \leftarrow i$
      break
  for each $y \in Y$ do
    $j \leftarrow 1$
    while true do
      If $j \leq k$ then
        $y_j^l \leftarrow y_{i\_row}^l$
        $j \leftarrow j + 1$
        continue
      else if $j \leq len(y)$ then
        $y_j^l \leftarrow 0$
        $j \leftarrow j + 1$
        continue
      else
        break
    $D_{train} \leftarrow D_{train} \cup \{(y, y^{1})\}$
  return $D_{train}$
```

## 4. Results

This section first introduces the experimental environment and data set used in the experiment, comparing the performance of different target models before and after deployment of defense, as well as during defense against two methods of attack and, finally, comparing the two methods of defense under the same attack.

### *4.1 Experimental environment*

The experimental environment was a computer equipped with an R7-4800h CPU and an RTX2060 graphics card; running Windows 10. Python 3.7 was the programming language; and TensorFlow2.3 was the main DL tool.

### 4.2 Data set

As shown in Table 2, the following data sets were used in accordance with the number of categories:

- *EyePACS*: EyePACS is a competition data set for diabetic retinopathy in Kaggle. There were five categories, including 35,108 images. Among them, 10,000 images were used as the training data, whereas 10,000 images were used as the test data.
- *CH-MNIST* (Kather *et al.*, 2016): CH-MNIST is a human colorectal cancer-related data set that is now publicly available for direct loading on TensorFlow data sets, with a total of eight categories and 5,000 images. Among them, 2,500 images were used as the training data, whereas 2,500 images were used as the test data.
- *CIFAR-100*: CIFAR-100 is a data set consisting of 60,000 32 × 32 color images of 100 categories. The data set is loaded and used on Keras. Here, 10,000 images were used as the training data, whereas 10,000 images were used as the test data.
- *Caltech-UCSD Birds 200* (Welinder *et al.*, 2010): Caltech-UCSD Birds 200 is a data set consisting of 11,788 images of 200 bird species, which is available for download from Kaggle. Among them, 5,994 images were used as the training data, whereas 5,794 images were used as the test data.
- *Texas100*: Texas100 is a data set that includes hospital discharge data. The data set contained inpatient information from multiple medical facilities and was published by the Texas Department of Health Services. A total of 67,330 processed data were obtained by Shokri (Salem *et al.*, 2019), which recorded 6,170 binary characteristics of external causes of harm (e.g. suicide and drug abuse), diagnosis (e.g. schizophrenia and illegal abortion), procedures performed by patients (e.g. surgery) and general information such as gender, age, ethnicity, hospital ID and length of stay. These data had 100 categories, and 10,000 disjoint data were selected as the training data and 10,000 as the test data.

### 4.3 Influence of the generative model on the target model

DL has abundant and diverse potential applications (Gao *et al.*, 2022). In this study, four commonly used models in computer vision, namely, residual neural network (ResNet) (He *et al.*, 2016), visual geometry group (VGG) (Simonyan and Zisserman, 2014), densely connected convolutional networks (DenseNet) (Huang *et al.*, 2017) and convolutional neural networks (CNN), were primarily selected as classification models. Detailed information about the model is shown in Table 3. The training data used for this study were member data. To showcase the transferability of our approach, the same generative model was used to process the output of different classifier models on the same data set. The differences in classification accuracy between the models adjusted using our defense method and the target model are illustrated below.

| Data set | No. of categories | Training data size | Test data size |
|---|---|---|---|
| EyePACS | 5 | 10,000 | 10,000 |
| CH-MNIST | 8 | 2,500 | 2,500 |
| CIFAR-100 | 100 | 10,000 | 10,000 |
| Caltech-UCSD | 200 | 5,994 | 5,794 |
| Texas100 | 100 | 10,000 | 10,000 |

**Source:** Authors' own design

**Table 2.** Details of each data set, training data and test data

Figures 5–8 illustrate the accuracy results of using ResNet (He *et al.*, 2016), VGG (Simonyan and Zisserman, 2014), DenseNet (Huang *et al.*, 2017) and CNN to develop classification models for EyePACS, CH-MNIST (Kather *et al.*, 2016), CIFAR-100 and Caltech-UCSD Birds 200 (Welinder *et al.*, 2010). The results of applying our defense method to these models show that, with the exception of the CH-MNIST data set (for which the model accuracy difference was approximately 0.06), the classification accuracy of the other models remained almost unaffected. The difference between the adjusted model accuracy and the unadjusted model accuracy was maintained at approximately 0.02 or less for both the training and test data sets.

### *4.4 Performance of defense method under attack*

*4.4.1 Attack model.* Existing attack models are mainly divided into two categories according to the use of a shadow model. The first category is the attack that includes the shadow model. This model simulates the output distribution of the target model by creating a shadow model of the known training data, which is also the primary mode of the early member inference attack. Here, the attack mode with shadow model Ml-leaks of Salem *et al.* (2019) is selected. The other category is the shadow model-free attack mode; that is, no shadow model training is required. Classification is performed using methods such as comparing the threshold differences or directly differentiating between the performance of the target model on training data and non-training data. The attack mode without the shadow model used in this study was the blind membership proposed by Hui *et al.* (2021):

**Table 3.**
Number of layers constructed for each classification model

| Model | No. of layers |
|---|---|
| ResNet | 50 |
| VGG | 16 |
| DenseNet | 121 |
| CNN | 2 |

**Source:** Authors' own design

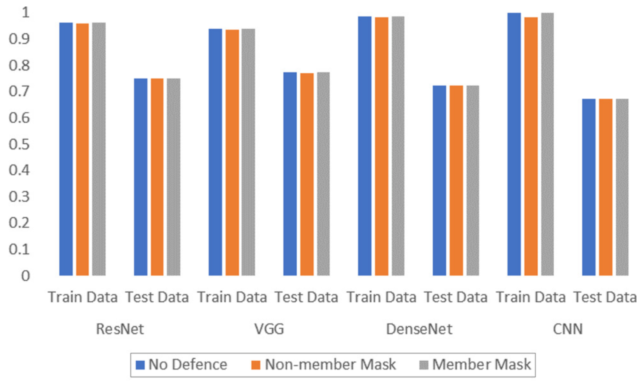
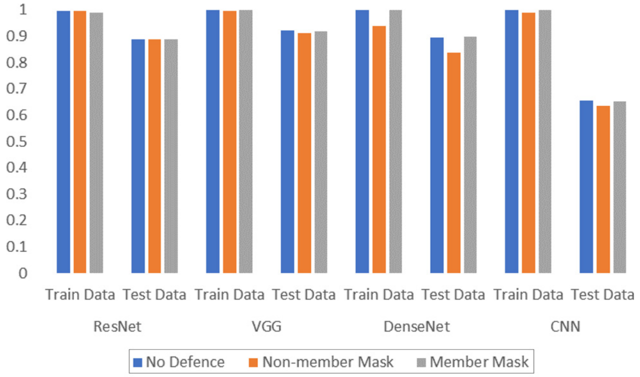


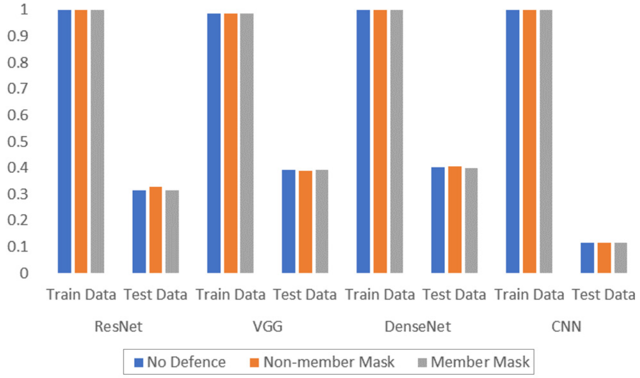
**Figure 5.**
Pre- and post-defense accuracy of each model on EyePACS

**Source:** Author's own design



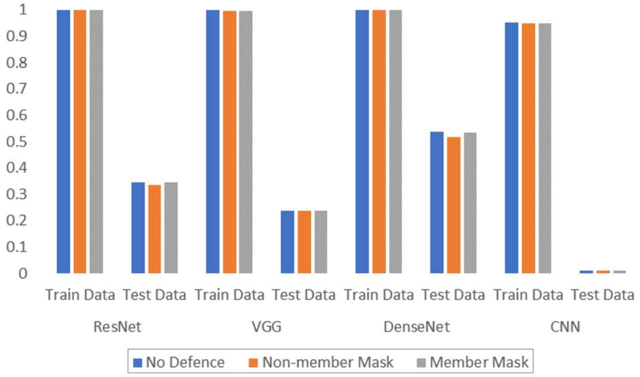
**Source:** Author's own design

**Figure 6.**
Pre- and post-defense accuracy of each model on CH-MNIST



**Source:** Author's own design

**Figure 7.**
Pre- and post-defense accuracy of each model on CIFAR-100



**Source:** Author's own design

**Figure 8.**
Pre- and post-defense accuracy of each model on Caltech-UCSD Birds 200

- *Ml-leaks* (Salem *et al.*, 2019): Ml-leaks trains a shadow model across the known training data. Then, it takes the known training data and non-training data to output with the shadow model; takes the first three maximum probability values of the probability output; and conducts supervised learning according to the two labels of both data outputs to obtain the classifier used to distinguish both outputs. The classifier is then used to classify the output of the target model to determine whether the input samples are the training samples for the target model. For Ml-leakage (Salem *et al.*, 2019), a black-box setup is used, where the attacker knows the true class of the sample.
- *BlindMI* (Hui *et al.*, 2021): BlindMI uses the difference between the output of the member data on the model and the output of the non-member data on the model. Assuming the attacker has a data set containing data from a member of the target model, the attacker refers to this as the target data set. The attacker constructs a non-member data set with no member data and moves a target sample to the non-member data set. The change in distance between the non-member data set and the target data set before and after motion is then used to determine whether the moved sample belongs to the training data of the target model. For BlindMI, the Blackbox-Blind of Hui *et al.* (2021) was adopted; that is, the attacker knows only the output of the model but not the ground truth of the sample.

The target model was ResNet; the shadow model used in the Ml-leaks was VGG; and our defense method was evaluated by both attacks. The results are presented in Tables 4 and 5. For the Ml-leaks attacks, our defense method effectively controlled the classification accuracy of the attack to approximately 0.5 or less. The precision rate of the Ml-leaks attack tends to zero when the data set with more classification categories is classified under the non-member mask because the attack at this point classifies all samples in the non-membership data. To achieve the goal of having the attacker mistake a non-member sample for a member sample, the member mask also effectively improves the recall rate of the Ml-leaks attack while maintaining the precision at approximately 0.5.

For the BlindMI attack, although our defense method effectively reduced the accuracy of the attack to approximately 0.5, it cannot control the classification result of the attack through the defense. This is because the Ml-leaks attack uses the shadow model to simulate the output of the target model to train the attack classifier with a known member; moreover, its essence is to create a high probability for the attack classifier through the similarity between the output of the shadow model and the output of the target model. In contrast, for

| Attack method | Metric | No. defended | Non-member mask | Member mask |
|---|---|---|---|---|
| Ml-leaks | Accuracy | 0.9134 | 0.5000 | 0.5396 |
| | Precision | 0.9033 | 0.0000 | 0.5206 |
| | Recall | 0.9702 | 0.0000 | 0.9999 |
| | F1-score | 0.9355 | – | 0.6847 |
| BlindMI | Accuracy | 0.8830 | 0.5010 | 0.5031 |
| | Precision | 0.8119 | 0.5005 | 0.5039 |
| | Recall | 0.9969 | 1.0000 | 0.4033 |
| | F1-score | 0.8958 | 0.6671 | 0.4480 |

**Source:** Authors' own data, using the Ml-leaks described by Salem *et al.* and BlindMI described by Hui *et al.*

**Table 4.**
Pre- and post-defense accuracy of attack model on CIFAR-100

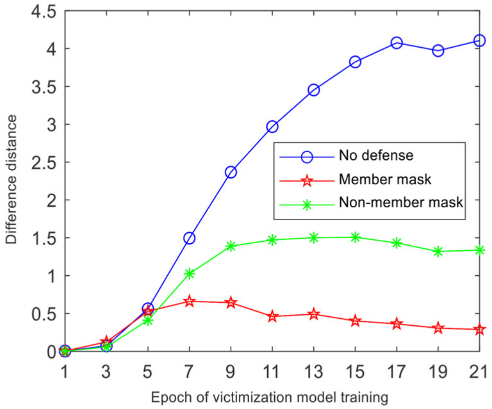
BlindMI, the attack is judged solely by the distance of the target data from the non-member data before and after the sample movement, and the distance is considered constant. After deploying the defense method, the difference between the target and non-member data decreases, so moving the sample does not change the distance between the two data sets. Eventually, all of the target data are classified as member data. Thus, the recall rate of BlindMI attacks on the CIFAR-100 data set may approximate 1. The attack classification accuracy was significantly reduced to approximately 0.5, and the attack was also invalid. Although the attack detected a change in distance in some cases, its accuracy rate was low, and these changes were not used to determine membership.

To control the difference in distance between the model outputs, the training times of the target models were controlled. The difference distance was measured using the centroid distance from the Hilbert space constructed by the Gaussian kernel used in BlindMI (Hui *et al.*, 2021). As shown in Figure 9, the difference distance of the target model output increased rapidly as the training time increased, whereas the output fitted by our defense method tended to be steady and did not increase significantly. Our defense system effectively controls the difference in model outputs. Figure 10 shows that the accuracy of the

| Attack method | Metric | No. defended | Non-member mask | Member mask |
|---|---|---|---|---|
| Ml-leaks | Accuracy | 0.9838 | 0.5000 | 0.5950 |
| | Precision | 0.9691 | 0.0000 | 0.5566 |
| | Recall | 1.0000 | 0.0000 | 1.0000 |
| | F1-score | 0.9843 | – | 0.7152 |
| BlindMI | Accuracy | 0.9801 | 0.0101 | 0.4983 |
| | Precision | 0.9623 | 0.0000 | 0.5143 |
| | Recall | 1.0000 | 0.0000 | 0.2397 |
| | F1-score | 0.9808 | – | 0.3270 |

**Source:** Authors' own data, using the Ml-leaks described by Salem *et al.* and BlindMI described by Hui *et al.*

**Table 5.** Pre- and post-defense accuracy of attack model on Caltech-UCSD Birds 200



**Source:** Authors' own data, using the BlindMI described by Hui *et al.*

**Figure 9.** Difference distance with target model training times

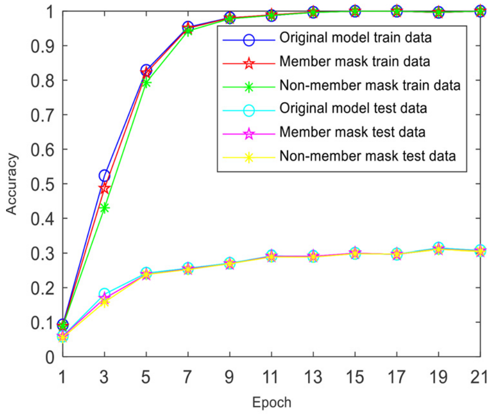


**Figure 10.**
Accuracy with target model training times

**Source:** Authors' own data, using the BlindMI described by Hui *et al.*

model output fitted by the defense method was consistent with that of the target model during the training process, indicating that the classification accuracy of the model is almost unaffected by the defense. The classification accuracy of BlindMI for the entire model training process is illustrated in Figure 11. The attack classification accuracy for the defense-free model increased rapidly in parallel with the increase in training time, rendering the model vulnerable to BlindMI attack. For the post-defense model attack, the accuracy increased slightly during the training process for the target model, and then it gradually tended to a steady state before eventually dropping to approximately 0.5. In the case of MIA as a binary classification, the result of the attack classification is unreliable, and the defense effectively protects the model.

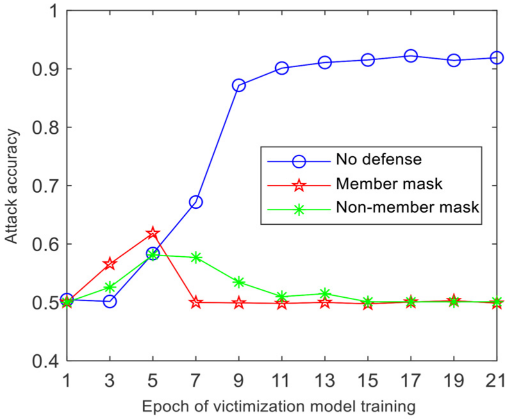


**Figure 11.**
Attack accuracy with target model training times

**Source:** Authors' own data, using the BlindMI described by Hui *et al.*

### *4.5 Comparison of defense methods*

The method selected for comparison with our defense method is MemGuard proposed by Jia *et al.* (2019), which also adapts the output of the model, and the dropout defense method proposed by Salem *et al.* (2019), which adjusts the model to improve its generalization ability.

For the purpose of comparison, MemGuard (Jia *et al.*, 2019) was selected, which also adapted the output of the model, and the dropout defense method (Salem *et al.*, 2019), which adjusted the model to improve its generalization ability. MemGuard (Jia *et al.*, 2019) uses an output adjustment technique to defend against MIA and adds noise to the model output to reduce the difference between member and non-member outputs while preserving the model classification results. The Texas100 data set was chosen to train the target model. The effectiveness of the defense method was evaluated using BlindMI as the MIA, and the results are presented in Table 6. Although MemGuard has minimal impact on the classification accuracy of the target model (with only a reduction of 0.007 in the test data accuracy), the noise generated by MemGuard retains the original output characteristics, resulting in residual output differences that are detected by BlindMI to distinguish between member and non-member (the precision of the BlindMI attack under MemGuard is 0.5930). In contrast, our defense method achieves a maximum classification accuracy of only 0.5 against Blind attacks. In binary classification, an attack at this level is not effective in distinguishing between members and non-members, rendering the attack invalid.

Our method was also compared with the dropout defense (Nicolas *et al.*, 2017). The attack method used was BlindMI, and the CIFAR-100 and ResNet models were used. Our defense method and dropout defense were applied to defend against BlindMI for comparison purposes.

| Evaluation indicators | No. defended | MemGuard | Non-member mask | Member mask |
|---|---|---|---|---|
| Training accuracy | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Test accuracy | 0.5680 | 0.5610 | 0.5680 | 0.5590 |
| BlindMI accuracy | 0.7485 | 0.5930 | 0.5000 | 0.5010 |
| BlindMI precision | 0.7934 | 0.7236 | 0.5000 | 0.5005 |
| BlindMI recall rate | 0.6720 | 0.3010 | 0.9990 | 1.0000 |
| BlindMI F1-score | 0.7277 | 0.4251 | 0.6664 | 0.6671 |

**Source:** Authors' own data, using the Ml-leaks described by Salem *et al.* and BlindMI described by Hui *et al.*

**Table 6.** Performance of BlindMI attack in various defenses

| Training accuracy | Test accuracy | Attack accuracy with dropout | Attack accuracy with member mask | Attack accuracy with non-member mask |
|---|---|---|---|---|
| 0.5234 | 0.3024 | 0.7652 | 0.5009 | 0.5035 |
| 0.7792 | 0.3255 | 0.8580 | 0.5009 | 0.4905 |
| 0.9554 | 0.3542 | 0.8641 | 0.5027 | 0.5010 |

**Source:** Authors' own data, using the Ml-leaks described by Salem *et al.* and BlindMI described by Hui *et al.*

**Table 7.** BlindMI in models with different overfitting degrees and defense methods

As shown in Table 7, as the discrepancy between the accuracy of the training data and that of the testing data increases (i.e. the degree of model overfitting increases), the classification accuracy of the attack under dropout defense gradually increases, eventually reaching 0.8641. At this point, the dropout defense becomes vulnerable to BlindMI. However, the attack accuracy under our defense method remains at approximately 0.5 and is unaffected by the degree of model overfitting, indicating that our defense method effectively withstands the BlindMI attack.

## 5. Conclusion

This paper proposes a cost-effective and straightforward approach for safeguarding against MIAs by adjusting ML model outputs. The defense strategy involves a simple generative model structure, resulting in a low defense cost. The experimental findings demonstrate that the proposed method successfully preserves the classification information of the original model while simultaneously reducing the disparities between member and non-member outputs, thus effectively mitigating the risks associated with MIAs. Additionally, the proposed approach proves successful in classification models with simple output structures, but the effectiveness is reduced in generative models with more complex outputs. As the method does not account for the classification categories of non-member outputs, it is vulnerable when attackers have knowledge of the true sample categories. Finally, the method is suitable for black-box scenarios, and further investigation and enhancement are required for white-box scenarios.

## References

Chen, D., Yu, N., Zhang, Y. and Fritz, M. (2020), "GAN-Leaks: a taxonomy of membership inference attacks against generative models", *Computer and Communications Security*, pp. 343-362.

Gao, H., Fang, D., Xiao, J., Hussain, W. and Kim, J.Y. (2022), "CAMRL: a joint method of channel attention and multidimensional regression loss for 3D object detection in automated vehicles", *IEEE Transactions on Intelligent Transportation Systems*.

Gao, H., Huang, W., Liu, T., Yin, Y. and Li, Y. (2022), "PPO2: location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems", *IEEE Transactions on Intelligent Transportation Systems*.

Gao, H., Huang, J., Tao, Y., Hussain, W. and Huang, Y. (2022), "The joint method of triple attention and novel loss function for entity relation extraction in small data-driven computational social systems", *IEEE Transactions on Computational Social Systems*, Vol. 9 No. 6, pp. 1725-1735.

Gao, T. (2022), "Research progress and challenges of membership inference attacks in machine learning. (in Chinese)", *Operations Research and Blurring*, Vol. 12 No. 1, pp. 1-15.

He, K., Zhang, X., Ren, S. and Jian, S. (2016), "Deep residual learning for image recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *Las Vegas*, pp. 770-778.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S. and Zhang, X. (2021), "Membership inference attacks on machine learning: a survey", *ACM Computing Surveys (CSUR)*, Vol. 54, pp. 1-37.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017), "Densely connected convolutional networks", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *Honolulu*, pp. 2261-2269.

Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N.Z. and Cao, Y. (2021), "Practical blind membership inference attack via differential comparisons", NDSS.

Jia, J., Salem, A., Backes, M., Zhang, Y. and Gong, N.Z. (2019), "MemGuard: defending against black-box membership inference attacks via adversarial examples", *ACM SIGSAC Conference*, *London United Kingdom*, pp. 259-274.

Kather, J.N., Weis, C.A., Bianconi, F., Melchers, S.M., Schad, L.R., Gaiser, T., Marx, A. and Zöllner, F.G. (2016), "Multi-class texture analysis in colorectal cancer histology", *Scientific Reports*, Vol. 6 No. 1, p. 27988.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. (2020), "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871-7880.

Nasr, M., Shokri, R. and Houmansadr, A. (2018), "Machine learning with membership privacy using adversarial regularization", *ACM Conference on Computer and Communications Security*, pp. 634-646.

Nicolas, P., Martín, A., Úlfar, E., Goodfellow, I. and Talwar, K. (2017), "Semi-supervised knowledge transfer for deep learning from private training data", ICLR 2017, Palais des Congrès Neptune.

Pandey, S.K. and Janghel, R.R. (2021), "Classification of electrocardiogram signal using an ensemble of deep learning models", *Data Technologies and Applications*, Vol. 55 No. 3, pp. 446-460.

Salem, A., Zhang, Y., Humbert, M., Fritz, M. and Backes, M. (2019), "ML-leaks: model and data independent membership inference attacks and defenses on machine learning models", *Network and Distributed System Security Symposium (NDSS)*.

Santhi, J.A. and Saradhi, T.V. (2021), "Attack detection in medical internet of things using optimized deep learning: enhanced security in healthcare sector", *Data Technologies and Applications*, Vol. 55 No. 5, pp. 682-714.

Shejwalkar, V. and Houmansadr, A. (2021), "Membership privacy for machine learning models through knowledge transfer", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35 No. 11, pp. 9549-9557.

Shokri, R., Stronati, M. and Shmatikov, V. (2017), "Membership inference attacks against machine learning models", *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3-18.

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition".

Tong, M., Xu, J., Zheng, Y. and Pedrycz, W. (2020), "A survey on machine learning for data fusion", *Information Fusion*, Vol. 57, pp. 115-129.

Welinder, P., Branson, S., Mita, T., Catherine, W., Florian, S., Serge, B. and Pietro, P. (2010), "Caltech-UCSD birds 200", California Institute of Technology. CNS-TR-2010-001.

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

**Corresponding author**
Hai Liang can be contacted at: lianghai@guet.edu.cn