University of Zurich[UZH]

INDEPENDENT STUDY – Communication Systems Group, Prof. Dr. Burkhard Stiller

Design and Implementation of a Privacy Auditing Component for the Decentralized Federated Learning Framework

Yuanzhe Gao
Zurich, Switzerland
Student ID: 20-752-200

Supervisor: Chao Feng, Dr. Alberto Huertas Celdran
Date of Submission: June 4 , 2024

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Independent Study
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Declaration of Independence

I hereby declare that I have composed this work independently and without the use of any aids other than those declared (including generative AI such as ChatGPT). I am aware that I take full responsibility for the scientific character of the submitted text myself, even if AI aids were used and declared (after written confirmation by the supervising professor). All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Zürich, Yuanzhe Gao

Signature of student

Abstract

Member inference attack (MIA) poses a serious threat to model security in machine learning. An effective MIA can expose the user's training data to a large extent, resulting in incalculable privacy breaches. Although MIA is a very mature concept in the field of machine learning, there are many ways to deal with it. However, in the context of federated learning, especially decentralized federated learning, the performance of such attacks is still not much effectively evaluated and deeply logical analysis.

This work begins with an investigation and analysis of the well-known membership inference attacks that are effective in machine learning, and a comprehensive evaluation of them in decentralized federated learning context. At the same time, this assessment tool for MIA has been integrated into Fedstellar. The results show that the effectiveness of most MIA attacks is significantly reduced in the federated learning environment, which proves that the federated learning approach helps to further maintain the privacy of participants' user data. On the other hand, it is also found that different participants have different member reasoning abilities in different topologies. This distinction is often related to its position in the overall network. By exploiting this unique manifestation of decentralized federated learning, new causal inference attacks in the direction of network topology may be created. This is also an important contribution made by this work.

Acknowledgments

First of all, I would like to express my sincere gratitude to supervisor Chao Feng and Dr. Alberto Hueartas of my indenpendent study for their guidance. Especially Chao Feng, whose weekly meeting communication and his selfless help anytime and anywhere provided me with great motivation and help to complete this work. In addition, I am also very grateful to Prof. Dr. Stiller for allowing me to carry out this project at the Communication Systems Group. It was with this help that I was able to finish the work.

Contents

Chapter 1

Introduction

1.1 Motivation

Membership inference attack (MIA) poses a profound threat to the privacy of user training data in machine learning. There are many mature attack methods and defense methods, as well as in-depth research on attack mechanisms. However, the performance of this attack method in the federated learning environment is still an area that has not been deeply studied, especially in the decentralized federated learning context[1]. As the network topology structure and data distribution of the entire federation have undergone profound changes, the changes in the effectiveness of membership inference attacks are worthy of researchers' deep consideration. In this context, this work aims to comprehensively evaluate the effectiveness of effective attack methods in the original machine learning background in decentralized federated learning, and establish an effective connection between its performance and factors such as network settings.

1.2 Description of Work

This work aims to establish a privacy monitoring module to enable the existing federated learning platform to simulate the damage caused by membership inference attack, and at the same time evaluate the performance and impact of this attack method on machine learning and federated learning as comprehensively as possible. The entire project is mainly divided into several phases:

Design and implementation of Different MIAs: This phase includes the understanding of MIA knowledge and the exploration of DFL structure. By understanding the operating mechanism of the two, a solid foundation will be laid for subsequent practice and analysis.

Background Research and Problem Understanding: At this stage, the attack methods of MIA specifically planned to be implemented will be analyzed and reproduced in detail, and a specific machine learning and federated learning environment would be built to facilitate the following large-scale experimental evaluation.

Evaluation and Conclusion: In this phase, experiments on the implemented MIA in different scenarios will be tested on a large scale and the results will be recorded for detailed analysis. At the same time, a complete report will be also produced to summarize all the achievements.

In summary, this work aims to comprehensively evaluate the different performances of different membership inference attacks in the two environments of machine learning and federated learning. On the one hand, it aims to explore whether the original attack methods can still be effective in the case of federated learning. On the other hand, what impact will the unique mechanism and characteristics of federated learning have on MIA, and whether it will expose the privacy data of additional participants. At the same time, these attack methods have also been implemented and integrated into the Fedstellar this federated learning platform.

1.3 Report Outline

The structure of this work is outlined as follows. First, Chapter 2 establishes the theoretical foundation required for this work and analyzes the reasons and necessity for doing so. Next, Chapter 3 reviews the relevant achievements in this field and gives a general analysis of its operating mechanism. Chapters 4 and 5 introduce the specific details of the relevant MIAs and how it is integrated into Fedstellar from an engineering perspective. Finally, Chapter 6 shows all the experimental results and relevant conclusions are given in Chapter 7. In addition, in the last chapter, the limitations of this work and the future development prospects in this area are also mentioned.

Chapter 2

Background

This chapter introduces the characteristics and classification of federated learning, and explains its advantages over traditional machine learning. It also describes the background, purpose, and specific categories of inference attacks, and analyzes the privacy hazards that inference attacks, as a special attack method, bring to normal machine learning models.

2.1 Federated Learning

Traditional machine learning relies on transferring large amounts of data to a unified place (or device) for centralized training. However, this centralized approach has two main drawbacks: (1) With the increasing complexity of the model learning task, the data format needed by the model itself becomes more complex, which leads to the sharp increase of the data occupation space.This makes transmission efficiency often an important factor restricting performance when large batches of data are frequently transmitted; (2) On the other hand, this can create significant privacy and security issues, especially when sensitive private data is needed. Therefore, with the introduction of various privacy protection regulations and the gradual improvement of user security awareness, these drawbacks become more serious in the actual situation.
In this context, federated learning comes into being. The essence of federated learning is a distributed machine learning method[[2],[3] . By combining the learning results of multiple participants on their local machines, FL attempts to keep the data locally while obtaining learning performance comparable to those of centralized training. The key to this success lies in a specialized model aggregation approach.

2.1.1 Aggregation Method

Compared with traditional data sharing, federated learning achieves the effect of multi-party collaboration by sharing model training results such as gradients or parameters. Among them, FedAvg aggregation algorithm is the simplest but most popular method. It

allows each participant to transmit the newly calculated gradients or model parameters in encrypted form to a central server after completing local model training. The server then averages the collected results of all individuals and returns them to each participant as the starting parameters for the next round of local training. Through several rounds of this form of parameter transfer, the local models of each participant tend to converge, thus achieving the effect of training the data set together.

2.1.2 Different Types of Federated Learning

Federated learning can be divided into the following types based on its network topology and communication method:

- **Centralized Federated Learning (CFL)**
  Under this framework, all participants are interconnected with one central server together. This central server is responsible for aggregating the collected model parameters in each round and broadcasting the aggregated one back to all participants[4]. As for the local participants, they only need to handle the task of local model training. This federation is architecturally simple, but the central server can be a potential risk for single point of failure and privacy breach. Besides, because all participants depend on the central server's collaboration, when there are too many participants involved, the server needs to handle more communication requests, resulting in a significant increase in its transmission overhead.

- **Decentralized Federated Learning (DFL)**
  Unlike CFL, decentralized federated learning adopts a more autonomous communication method. There is no central server to coordinate the connections between participants. Instead, each participant takes on the task of aggregating and broadcasting the adjacent node models. The benefits include that since there is no single point server, the entire system does not need to worry about a single point of failure (SPOF), which enhances the overall robustness and fault tolerance when facing attacks. At the same time, the DFL architecture increases the scalability of the overall system. When new nodes join or existing nodes withdraw from the federation, it will not have a significant impact on the overall structure. However, due to the lack of a node to coordinate the overall communication, the communication between different rounds of DFL nodes needs to be treated more cautiously to ensure the aggregation between the correct models each time.

2.2 Inference Attack

Inference attack generally refers to the behavior that an attacker obtains the information that the model user does not want to be exposed by processing and utilizing the features of the machine learning model. Generally speaking, the types of information obtained by inference can be very wide, including the user's training data, the parameter Settings of the model itself, and the specific data characteristics. At the same time, because the

attacker usually does not actively interfere with the normal process of model training, it is difficult for the attacked party to detect the information leakage. This also leads to effective inference attacks that can often cause incalculable harm to privacy disclosure. In some industries with high data privacy requirements, such as financial transaction data, medical and health data, inference attacks will always be treated and prevented very carefully.

2.2.1 Types of Inference Attack

Inference attack can be mainly divided into the following types according to the different targets of attack:

- **Membership Inference Attack**
  Membership inference attack is the most basic but also the most widely used type of inference attack. Its goal is to determine whether a data point belongs to the training set of the target model. Because in machine learning, the trained model often performs differently to the data of the training set than other non-member data points. By capturing the difference between the two, the attacker can effectively identify the members of the model training set. MIA can be very harmful in many contexts, such as in the medical industry, where training data to identify medical models often result in a serious breach of a patient' s personal health information.

- **Attribute Inference Attack**
  Attribute inference attack generally aims to infer the missing or hiding attributes among training data samples[5]. Even though some kinds of sensitive attributes are not included in the raw data clearly, attackers can still speculate them through the model output or the activation values by model's intermediate layers. For example, in an e-commerce model that includes user purchase history, an attacker may be able to infer a user's income level from other known attributes, such as age, gender or frequency of purchase. Attribute inference attack not only can leak user's personal privacy, but also provide further attack vector to attackers for next time use.

- **Property Inference Attack**
  Unlike attribute inference attack, the goal of property inference attack is to infer some global properties or statistical characteristics of the entire training dataset rather than the specific attributes of individuals[6]. In this case, the attacker hopes to understand some global properties which are not usually present in explicit labels of the training samples through analyzing the behaviour or parameters of target model. For example, an attacker might try to determine if a particular pattern exist in the training set such as a large number of labels of a certain type are included inside or an attacker want to infer the distribution of the training set, like whether balanced or biased.

- **Model Inversion Attack**
  The object of model inversion attack is to reconstruct the characteristics of input data or training data[7]. An attacker uses reverse optimization or other techniques

to gradually deduce these characteristics by accessing the model output[8]. For example, in the image classification model, an attacker can infer the image of one tag reversely through analyzing its model output. Model inversion attack is a particularly serious threat to privacy since it can reconstruct the original data directly, such as the patient's medical image or user's photograph.

2.2.2 Adversarial Knowledge

On the other hand, inference attack can also be classified into the following three categories according to the level of knowledge possessed by the attacker:

- **White-box Attack**
  The attacker has the full access rights to the target model's inner structure and parameters. By means of analyzing the gradient, parameters and training process of the model, the attacker is able to obtain more detailed information to utilize. The white-box attack generally has the simplest implementation difficulty and the best attack performance.

- **Black-box Attack**
  In this case, the attacker can merely access the inputs and outputs of the target model but not inspect its internal information directly. The attacker usually make use of statistical and machine learning techniques through attempting to send some specific inputs here to make up for the gap of adversarial knowledge. Establishing the valid connection between input and output is the primary goal for black-box attackers. Corresponding, this kind of attack is the most difficult to carry out and the hardest to defend[9].

- **Semi-white-box Attack**
  On the basis of black-box attack information, the attacker own partial model information additionally, including some parameters or parts of the model architecture. This setting is somewhere between a white-box and a black-box (sometimes referred to as gray-box) and its performance and difficulty are in the middle accordingly.

All in all, inference attacks pose a serious privacy and security threat in the field of machine learning. With the evolution and complexity of attack methods, it becomes particularly important to understand and explain them. This not only helps in building defences, but also in understanding and improving the bottlenecks in the original models.

Chapter 3

Related Work

This chapter introduces the existing platforms on the market for privacy assessment against different attacks, as well as the means and defense methods for existing membership inference attacks in machine learning and federated learning environments.

3.1 Existing Privacy Audit Platform

With the wide application of machine learning technology, privacy protection is increasingly concerned. In response to various privacy breach threats, including membership inference attacks (MIA), several privacy auditing platforms and tools have emerged. These platforms mainly ensure the privacy and security of user data through monitoring and protection mechanisms. Here are the main privacy monitoring platforms and their features:

- **TensorFlow Privacy**
  TensorFlow Privacy is a library for enhancing the privacy protection of TensorFlow models. It implements differential privacy technology, which protects data privacy mainly by adding noise to the gradient during training. While its primary purpose is to protect against differential attacks, it is also somewhat resistant to MIA because the introduction of noise reduces the degree to which the model overfits the training data.

- **PySyft and PyGrid**
  PySyft is an in-depth learning library for encryption and privacy protection that supports federated learning and differential privacy. It can train and reason models without exposing real data. PyGrid is an extension of PySyft that provides a distributed platform for managing and deploying privacy protection models. These two libraries protect against many privacy attacks including MIA through encryption and differential privacy.

- **Google JAX with DP-SGD**
  Google Jax is a high-performance machine learning library that combines a differential privacy gradient (DP-SGD) algorithm to protect the privacy of training data. By adding noise and clipping gradient in the training process, JAX can effectively reduce the dependence of the model on training data and reduce the success rate of MIA attack.

- **IBM Differential Privacy Library** IBM's differential privacy library provides a set of tools specifically designed to incorporate differential privacy protection into the implementation of data analysis and machine learning tasks. The core of the library is to ensure the security of data privacy by mathematical methods, so as to prevent malicious attackers from using query and reasoning to obtain sensitive information. Although its main focus is to protect the privacy of statistical query, the protection mechanism can also effectively resist the security threats such as membership inference attack.

There are a number of privacy audit platforms that rely on differential privacy and advanced encryption techniques to protect machine learning models from all kinds of privacy violations, including MIA. These platforms not only ensure the security of model training and reasoning process, but also set a new benchmark for data privacy protection. In the future, these platforms will evolve to address increasingly complex privacy security challenges as the means of privacy attacks evolve.

3.2 Membership Inference Attack

There are two primary forms of membership inference attacks. The first involves training an additional model to determine whether specific data was used to train the target model. This approach is known as a *binary classifier-based attack* because it yields binary outcomes (in or out). The second method utilize calculating a specific metric based on the prediction vectors from the dataset and compares it to a selected threshold to make an inference judgment. Next, the discussion will focus on the well-known MIAs results in machine learning and federated learning, based on these two perspectives.

3.2.1 MIA against Machine Learning

Binary Classifier Based MIAs. Shokri et al.[10] proposed the first membership inference attack against machine learning models by training several binary classifiers (one for each prediction class), known as *attack models*, to determine whether data points belong to the target model's training set. To gather sufficient data for training attack models, they created several *shadow models* with the same architecture as the target model and collected the confidence vectors produced after training these shadow models. Due to the similarity between the shadow models and the target model, these confidence vectors could serve as 'in' training samples for the attack model. Similarly, 'out' training samples

were obtained in the same manner. Since executing the attack only requires access to the posterior confidences generated by the target model, it is classified as a black-box attack.

The success of this attack relies on two key assumptions: (1) The shadow model must have the same architecture and parameter settings as the target model, including training specifics such as the optimizer and learning scheduler. Only the posterior confidences generated by the shadow model under these conditions can be used as a substitute for the actual 'in' sample ones. (2) The training data for the shadow model should originate from the same distribution as the real training data, ensuring consistency in the dimensions of their feature spaces. Although Shokri et al. later mitigated this requirement by generating synthetic data, it remains essential for the evaluation of most data categories that are not binary features.

Afterwards, Salem et al.[11] extended this shadow training method by progressively relaxing the original assumptions regarding model and data requirements. First, they demonstrated that employing a single shadow model and attack model can achieve attack performance comparable to the original method. Although the accuracy of inference attacks is somewhat diminished, the reduction in complexity compensates for this drawback. Second, the authors found that while using a single shadow model differing from the target model results in lower attack accuracy, an attacker can enhance performance by combining a set of such machine learning models, each employing a different classification algorithm, together into one shadow model as long as the target algorithm is included. This property is achievable given the limited number of feasible classification algorithms. Finally, they relaxed the assumption that the attacker must possess a shadow model training set isodistributed with the target training data set. By merely collecting a fixed number of maximum posteriors to train the attack model, this so-called *data transfer attack* mitigates the impact of differing numbers of output classes between various data sets, so that the attacker is no longer constrained by the need to know the type of the target data set.

In summary, Salem et al. relaxed the assumptions of the original shadow model training MIA regarding model architecture and data distribution, making the attack more threatening and practical, and significantly expanding its application scope.

Nasr et al.[12] further enhanced the shadow training method to develop a membership inference attack based on a white-box scenario. Unlike the previous method, which only requires knowledge of the target model's output for a given data point, this white-box attack assumes that the attacker is aware of the model's architecture (typically a deep learning model using the SGD algorithm), including specific parameter values and the gradient values of each layer with respect to the loss. The authors argue that each training data point leaves a unique impact on the gradient values of the model parameters relative to the loss. By fully leveraging this information to train the attack model, it becomes possible to effectively distinguish between training and non-training data. Additionally, recognizing that the attacker may lack information about the shadow training data, they also proposed one alternative using unsupervised learning methods, such as clustering algorithms, to differentiate potential target dataset from non-members.

Metric Based MIAs. Compared to membership inference attacks based on binary classifiers, metric based MIAs are often simpler and easier to implement. There is no need to

train a sophisticated attack model. Instead, the attacker can distinguish between members and non-members by identifying a typical metric and determining an appropriate threshold value. Because the principle behind this attack is the belief that the metric values of data points after model training often differ from those of untrained data points, some classic metrics for measuring the performance of machine learning models, such as accuracy, prediction confidence, and entropy, are selected to test the effectiveness of the attack[10], [13], [14].

- **Prediction Correctness Based MIA[15]**
  This MIA considers data points whose model-predicted labels differ from the true labels as non-members. This approach is based on the observation that a trained model's predictions for its training members are often significantly more accurate than those for non-members. Unlike other attack methods, this technique only requires the predicted values of the dataset, making it very easy to implement. However, its performance heavily relies on the model's poor generalization.
  Mathematically, the classification method can be represented as follows:

$$MIA_{\text{Correctness}}(x_i, y_i) = \begin{cases} \text{Member (M)} & \text{if } \arg\max_j f(x_i)_j = y_i \\ \text{Non-member (N)} & \text{if } \arg\max_j f(x_i)_j \neq y_i \end{cases} \tag{3.1}$$

  where x_i is an individual data point, y_i is the true label for x_i, and $\arg\max_j f(x_i)_j$ is the predicted label by the target model for x_i.

- **Prediction Loss Based MIA[13]**
  This MIA uses the prediction loss of data points as a metric. It compares the prediction loss of each data point to the average training loss of the model, considering data points with a prediction loss greater than or equal to this threshold as non-members. The rationale behind this approach is that during model training, the objective is typically to minimize the prediction loss for training members. Consequently, the prediction loss for member data points should generally be lower than that for non-members. Mathematically, the classification method can be represented as follows:

$$MIA_{\text{Loss}}(x_i, y_i) = \begin{cases} \text{Member (M)} & \text{if } L(f(x_i), y_i) < \bar{L} \\ \text{Non-member (N)} & \text{if } L(f(x_i), y_i) \geq \bar{L} \end{cases} \tag{3.2}$$

  where x_i is an individual data point, y_i is the true label for x_i, $f(x_i)$ is the prediction vector by the target model for x_i, $L(f(x_i), y_i)$ is the prediction loss for x_i, and $\bar{L}$ is the chosen threshold, which is the average training loss of the target model.

- **Prediction Confidence Based MIA[11]**
  Similar to loss-based MIA, this confidence based MIA determines whether a data point is a training member by comparing its prediction confidence to a specific threshold value. The underlying principle is that the model tends to exhibit greater prediction confidence for training members, often close to 1. In this context, Naser uses the maximum prediction confidence as the metric, while Yeom uses the prediction confidence of the true label. These two variations can be expressed as follow:

$$MIA_{\text{Maximal\_confi}}(x_i) = \begin{cases} \text{Member (M)} & \text{if } \max_j f(x_i)_j \geq \tau \\ \text{Non-member (N)} & \text{if } \max_j f(x_i)_j < \tau \end{cases} \tag{3.3}$$

$$MIA_{\text{Label\_confi}}(x_i, y_i) = \begin{cases} \text{Member (M)} & \text{if } f(x_i)_{y_i} \geq \tau \\ \text{Non-member (N)} & \text{if } f(x_i)_{y_i} < \tau \end{cases} \tag{3.4}$$

where x_i is an individual data point, y_i is the true label for x_i, $f(x_i)$ is the prediction vector for x_i, $\max_j f(x_i)_j$ is the maximum confidence across all classes for x_i, $f(x_i)_{y_i}$ is the prediction confidence for the true label y_i, and τ is the threshold value.

- **Prediction Entropy Based MIA[11]**
  In machine learning, entropy is often viewed as an important measure of uncertainty for a given dataset. MIA attackers exploit the fact that the entropy of training member data points is often lower than that of non-members. This is because the target model is more certain about its predictions for training members. By calculating the entropy of data points and comparing it to a specific threshold value, attackers can make inferences about membership. Its mathematical form is as follows:

$$MIA_{\text{Entropy}}(x_i) = \begin{cases} \text{Member (M)} & \text{if } H(f(x_i)) < \tau \\ \text{Non-member (N)} & \text{if } H(f(x_i)) \geq \tau \end{cases} \tag{3.5}$$

$$H(f(x_i)) = -\sum_j f(x_i)_j \log f(x_i)_j \tag{3.6}$$

  where x_i is an individual data point, $f(x_i)$ is the prediction vector by the target model for x_i, $H(f(x_i))$ is the entropy of the prediction vector for x_i, and τ is the threshold value.

- **Modified Prediction Entropy Based MIA[16]**
  The authors identified a limitation in the original prediction entropy value: it does not incorporate any ground truth label information. For example, both a correct prediction with a confidence of 1 and an incorrect prediction with a confidence of 1 will result in a zero prediction entropy value. To more accurately capture the entropy difference between training members and non-members, they introduced the concept of ground truth label confidence into the original calculation method. They proposed a modified version of the prediction entropy metric and made member inference judgments by comparing this modified entropy. The specific calculation method is as follows:

$$MIA_{\text{Mod\_Entropy}}(x_i, y_i) = \begin{cases} \text{Member (M)} & \text{if } MH(f(x_i), y_i) < \tau \\ \text{Non-member (N)} & \text{if } MH(f(x_i), y_i) \geq \tau \end{cases} \tag{3.7}$$

$$MH(f(x_i), y_i) = -(1 - f(x_i)_{y_i}) \log(f(x_i)_{y_i}) - \sum_{j \neq y_i} f(x_i)_j \log(1 - f(x_i)_j) \tag{3.8}$$

  where x_i is an individual data point, $f(x_i)$ is the prediction vector by the target model for x_i, $MH(f(x_i), y_i)$ is the modified entropy of the prediction vector for x_i, and τ is the threshold value. Inside $MH(f(x_i), y_i)$, $f(x_i)_{y_i}$ is the confidence score for the ground truth label y_i, and $f(x_i)_j$ is the prediction confidence for class j.

- **Prediction Sensitivity Based MI[14]**
  The authors observed that the norm of the Jacobian matrix for training members

is often smaller than that for non-members. This observation stems from the fact that the Jacobian matrix effectively indicates the model's sensitivity to data point predictions. A well-trained machine learning model is usually less sensitive to perturbations in the feature space of the training set. Consequently, training members tend to have smaller Jacobian norms. Based on this insight, the attacker collects the second-order norms of the Jacobian matrices and applies a clustering algorithm to divide the data points into member and non-member sets.
Mathematically, this approach can be represented as follows:

$$MIA_{\text{Sensitivity}}(x_i) = \begin{cases} \text{Member (M)} & \text{if } \|J(f, x_i)\|_2 < \tau \\ \text{Non-member (N)} & \text{if } \|J(f, x_i)\|_2 \geq \tau \end{cases} \tag{3.9}$$

where x_i is an individual data point, f is the trained target model, $J(f, x_i)$ is the Jacobian matrix of the model f with respect to x_i, $\|J(f, x_i)\|_2$ is the second-order norm of the Jacobian matrix, and τ is the threshold value determined by the clustering algorithm.

3.2.2 MIA against Federated Learning

Compared to membership inference attacks in traditional machine learning, MIAs in the context of federated learning remain relatively unexplored. Existing MIAs for federated learning are mostly simple extensions of attacks designed for centralized machine learning models and rarely exploit the unique characteristics of federated learning (for example, attackers are more often in a white-box attack environment)[[17], [18]]. Moreover, these attacks have primarily been evaluated in centralized federated learning (CFL) environments, without considering the context of decentralized federated learning. In decentralized federated learning (DFL), varying network topologies and diverse data distributions among participants complicate the evaluation of the effectiveness of membership inference attacks further.

Naser et al. were the first to test the effectiveness of a white-box membership inference attack based on model gradient changes in a federated learning environment regulated by a central server. They examined the attack from both the server's and participants' perspectives and demonstrated that even individual participants could achieve significant inference accuracy. They attributed this capability to the repeated updates of model parameters on the same training set in federated learning[12].

Gu, Bai, and Xu[19] argued that most existing membership inference attacks against FL are less effective in multi-participant settings. To address this, they proposed CS-MIA, a novel membership inference attack based on prediction confidence series, which poses a more significant privacy threat to FL. This attack leverages the differences in prediction confidence between training and testing data, as well as multiple model versions over FL rounds. By employing a neural network to learn features from the confidence series, the authors designed effective inference algorithms for both local and global adversaries and also introduced an active attack for global adversaries to extract further information.

Hu et al.[20] proposed a specialized inference attack method for federated learning, known as the source inference attack. The goal of this attack is not only to obtain information about the training data within the entire federation but also to attribute this information to a specific participant. They demonstrated that in a federated learning system using the FedAvg algorithm with a malicious server, an attacker can effectively infer which participant's training data is involved. This is achieved by exploiting the characteristic that the local training data of different individuals results in varying prediction losses in the local models of other participants. This attack method broadens the application scenarios of MIA and poses a significant threat to participants with unique data in the federated learning system.

3.3 Defense Methods for MIA

In view of the increasingly serious threat of membership inference attack to the privacy of ML models, several defense strategies are proposed and deeply explored. This section introduces three efficient defense mechanisms: **Differential Privacy** , **Knowledge Distillation**, and **Dropout** techniques. These defenses provide strong academic support and practical guidance for privacy protection of machine learning models and even federated learning.

- **Differential Privacy[21]**
  Differential privacy (DP) is a powerful technology widely used in data privacy protection, especially in defense against MIA. By introducing noise into the data or model parameters during training, the technique ensures that the effect of any single data point on the model output is limited to an acceptable range, thus effectively protect the privacy of individual data.
  Differential privacy is mainly realized by the method of differential privacy gradient descent (DP-SGD). DP-SGD adds appropriate noise to the gradient in each step of the gradient update, and implements gradient clipping to control its influence range. This approach not only reduces the degree of over-fitting of the model to the training data, but also makes it difficult for the attacker to infer the specific information of the training data by analyzing the model parameters. Many studies show that the DP-SGD model performs well in defense against MIA attacks, which provides a solid theoretical basis and experimental support for privacy protection of ML model.

- **Knowledge Distillation[22]**
  Knowledge distillation is an efficient model compression and transfer learning technique. It simulates the behavior of a large and complex model (teacher model) by training a smaller model (student model).
  In implementation, knowledge distillation usually follows two main steps. First, a large, complex model (the teacher model) is trained to learn the intrinsic characteristics of the original data set. Next, a smaller model (the student model) is trained to produce outputs as close as possible to the predictions of the teacher model, rather than training directly on the original data set.

Because the student model does not contact the original training data directly, but obtains the knowledge by studying the prediction result of the teacher model, this makes it difficult for attackers to infer information from the original training data by analyzing the behavior of student models. This kind of indirect training method effectively separates the model from the original data, which improves the privacy protection ability of the model significantly.

- **Dropout[23]**
  Dropout is a widely used regularization strategy designed to suppress overfitting during training by randomly discarding a subset of neurons in a neural network, thereby improving the model's generalization performance.
  It works by randomly selecting and temporarily "turning off" (that is, setting their output to zero) a subset of neurons in the network during each training iteration. These neurons that are "switched off" are randomly selected in each iteration and do not repeat, and this randomness prompts the rest of the model to learn more robust and generalized features.
  In effect, Dropout effectively reduces the model's dependence on specific training data points by reducing overfitting, thereby reducing the success rate of membership inference attacks. The model does not overly "memorize" the features of a particular data point during training, making it difficult for an attacker to infer the content of the training data by analyzing the model's behavior. Therefore, Dropout provides an effective technical approach to privacy protection of machine learning models.

3.4 Research Motivation

Although membership inference attacks have been widely designed and evaluated in the field of machine learning, in federated learning, especially in decentralized federated learning, MIA research still faces many challenges and unsolved problems. The impact of unique features of DFL, such as network topology and non-iid distribution of datasets, on MIA has not been fully explored and understood.

First of all, the network topology in decentralized federated learning is complex and variable, and the communication and cooperation patterns among participants may significantly affect MIA performance. For example, in different topologies, the path of data transmission, the update frequency of model parameters and the way of synchronization will have different effects on the attack. In addition, datasets in a federated learning environment are typically non-independent and identically distributed, and the data held by each participant is highly personalized and biased, further adding complexity to the defense against Mia. Second, migrating some of the classic and effective MIA methods in machine learning to the federated learning case to explore their performance differences is meaningful. This not only can verify the effectiveness of these methods in the decentralized scene, but also provide new ideas and methods for the federated learning MIA defense.

To sum up, the motivation of this work is to fill the gap of MIA research in current federated learning, especially in view of the characteristics of DFL and to explore the

adaptability and effectiveness of classical MIA methods in the new environment. It is expected to provide theoretical support and practical guidance for improving the privacy protection capability of system. In this case, a summary of previous work whether focusing on either machine learning or federated learning is available in Table3.1. This overview highlights the need for innovative research and robust techniques to safeguard DFL systems against memberhsip inference attacks effectively.

Ref	Year	Scenario	Attack Knowledge	Attack Approach	Defense	Evaluation Metric
[[10]]	2017	ML	Black-box	Shadow model training	Top-K confidence, L2-regularization	AP,AR
[[11]]	2018	ML	Black-box	Shadow model training, Prediction Maximal Confidence, Prediction Entropy	Dropout, Model stacking	AP,AR
[[15]]	2018	ML	Black-box	Prediction Correctness, Prediction Loss	-	AP,AR
[[12]]	2019	ML CFL	White-box	Intermediate computation	-	AP,AR
[[13]]	2019	ML	Black-box	Prediction Label Confidence	-	AP,AR
[[16]]	2021	ML	Black-box	Modified Prediction Entropy	Adversarial regularization	AAR
[[20]]	2021	CFL	White-box	Prediction Loss	-	ASR
[[14]]	2023	ML	Black-box	Prediction Sensitivity	-	AP,AR
[[19]]	2023	CFL	Black-box	Shadow model training	-	ASR
[[18]]	2023	CFL	Black-box	Prediction Confidence Series	Dropout,Knowledge distillation, Secure Aggregation, DP	AAR, F1-Score

Table 3.1: Summary of membership inference attacks work and defense strategies in Machine Learning and Federated Learning scenarios (time ascending).

Chapter 4

Architecture

This chapter introduces the integration of the audit component designed to evaluate privacy leaks caused by membership inference attacks in Fedstellar, a comprehensive federated learning simulation platform[24]. It mainly includes the analysis of the content of this new module and how to be integrated into Fedstellar.

4.1 Privacy Audit Component

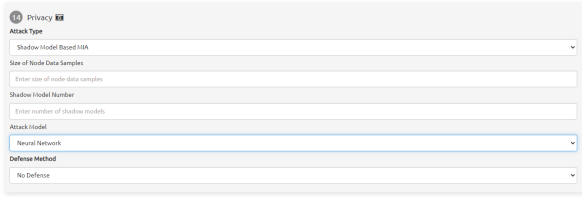
The structure of this privacy audit component mainly includes the following parts:

- **Front-end:** By providing a user-friendly front-end interface, users of Fedstellar can manually select the MIA method to be applied under the desired federated learning, and also allow users to adjust the specific attack parameters of different MIAs to better compare the effects of the attacks.
- **Attack Performing Module:** The attack module mainly contains the details of the specific MIA implementation. By using the model and data information contained in the original normal federation training, it simulates the possible attack attempts made by the attacker.
- **Logging Module:** After completing the MIA attack, the logging module will be responsible for recording the specific manifestations of the attack and displaying these manifestations through the original results presentation method of the Fedstellar platform, so that users can obtain different types of information data in a unified manner.

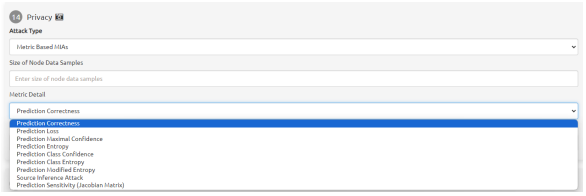
Figure4.1 shows the newly added privacy assessment options on the original front-end interface of Fedstellar. Users can first determine the type of MIA they want to evaluate, and then select the specific MIA from the list that appears. In the category of shadow model based MIA, users can also define the number of shadow models and the specific attack model types to adapt to their own computing power limitations. In addition,

users can choose to try to implement a defense against MIA in the overall federated learning process (currently only differential privacy is provided). Through this option menu, Fedstellar users can easily make the attack selection they want to evaluate without having to specifically touch the back-end code.

(a) Privacy Option



(b) Shadow Model Based MIA



(c) Metric Based MIA

(d) DP Defense

Figure 4.1: Privacy Auditing Frontent Interface

As for the attack performing module, which is the core part to implement different MIAs, from the perspective of software architecture, it can be seen as a part independent of the original Fedstellar modules. Unlike other attacks such as poison attacks, when users want to apply MIA to federated learning, they first need to obtain a trained model, which can be the version of each round or the final version, without interfering with the original communication process or training process. This also makes it difficult to detect membership inference attacks in the general sense. Its specific implementation code is shown below:

Listing 4.1: Membership Inference Attack Class Implementation

```
1
2  class MembershipInferenceAttack:
3      def __init__(self, model, global_dataset, in_eval, out_eval, logger,
           indexing_map):
4          ...
5
6      def compute_predictions(self, model, dataloader):
7          ...
8          return predictions, labels
9
10     def execute_attack(self):
11         # To be overridden by specific attack implementations
12         raise NotImplementedError("Must override execute_attack")
13
14     def evaluate_metrics(self, true_positives, false_positives):
15         ...
16         return precision, recall, f1
17
18 class ShadowModelBasedAttack(MembershipInferenceAttack):
```

```
    def __init__(self, model, global_dataset, in_eval, out_eval, logger,
        indexing_map, max_epochs,shadow_train, shadow_test, num_s, flag,
        file_name):
        super().__init__(model, global_dataset, in_eval, out_eval,
            logger, indexing_map)
        ...

    def _generate_attack_dataset(self):
        ...

    def MIA_shadow_model_attack(self):
        ...

class MetricBasedAttack(MembershipInferenceAttack):
    def __init__(self, model, global_dataset, in_eval, out_eval, logger,
        index_mapping, train_result):
        super().__init__(model, global_dataset, in_eval, out_eval,
            logger, index_mapping)
        ...

    def execute_attack(self):
        ...
```

After the attack is performed, the Logging module displays the attack performance data recorded by the node to the user in a visual way through Fedstellar's original Tensorboard logging way, making it easier for users to review all information data of federated learning in a centralized manner. The final display effect is as follows:

Figure 4.2: MIA Performance Logging in Fedstellar Tensorboard

Chapter 5

Design and Implementation

This chapter focuses on discussing the details of the implemented membership inference attacks in this work, which are listed in Table5.1. The characteristics of each attack are analyzed and demonstrated from the algorithm level. This helps to understand the circumstances of the attacker's attack, including the level of knowledge, attack considerations, and the cost of launching the attack. For ease of reference, all MIAs will be referred to by their abbreviations below.

5.1 Binary Classifier Based MIA

As explained above, MIA based on binary classifiers can also be called shadow model based MIA (SM MIA). Its main attack process relies on training an independent attack model to determine which data points may come from the training set of the target model. In order to obtain enough data to train this attack model, the attacker often trains one or more shadow models to generate prediction vectors that are comparable to actual members and non-members, which are the input data of the attack model.

The purpose of using shadow model technique is mainly to enable the attacker to obtain an object that can mimic the behavior of the target model so that it can clearly determine which data points will be correctly identified as actual members based on its own known divided shadow training sets (S_{train}) and shadow test sets (S_{test}). In this way, the attacker fully simulates the different reactions of the target model to its own training set and non-training set members, from the model to the data set, making this MIA extremely threatening.

Since the purpose of building shadow models is to obtain a representation similar to the target model, this MIA must rely on the following two key assumptions:

- **Shadow Model should have the same architecture and setting as the target model.** This assumption is particularly important. The consistency of this model not only includes the same type and parameters of the model used, but also the consistency of

the epochs and training set size. If this cannot be guaranteed, the resulting shadow model will never be able to simulate the behavior of the target model, even if a large number of trainings are performed. In this case, the attacker will never be able to obtain a valid decision boundary for the target data set.

- **Shadow training data should come from the same distribution of the target training set.** This assumption is to ensure the consistency of the target model and the shadow model results from the data perspective. Assuming that the shadow training data is completely different from the real training data set in terms of feature dimensions, the results it produces cannot be used as an effective input for the attack model.

Based on these two essential assumptions, the attacker can implement SM MIA in the following order. First, prepare a shadow dataset with the same distribution as the target training set, and divide it into two independent parts to ensure that there is no overlap between them. One of them is used to train the shadow model, refered as S_{train}. The other part is used as a non-member set that has not been accessed by the shadow model, called S_{test}.Afterwards, according to the number of shadow models to be implemented, an equal number of parts are extracted from these two groups as the training and test sets of the shadow model. The training sets of different shadow models can be crossed, which in fact also indirectly shows that the more shadow models, the better, unless there are extremely sufficient data sets for attackers to choose from. As the number of shadow models increases, the duplication between different shadow training datasets will increase, and eventually multiple shadow models will produce the same results, which is not conducive to improving the MIA effect. At the same time, it is also worth noting that the shadow training data set should be as non-intersecting as possible with the actual target training set to simulate a most realistic attack environment.

After preparing the dataset, the attacker should train the prepared shadow model. After training, the attacker will take S_{train} and S_{test} as input into the shadow model and obtain the corresponding prediction vector as output. The necessity of this process lies in that the original SM MIA is a membership inference attack created in a black-box environment, which means that the attacker can only access the input and output of the target model. Therefore, by obtaining the output results from the shadow model, this accurately utilizes the data form that the attacker can collect in this case. After that, these two different types of prediction vectors are used as in and out samples to train the final attack binary classifier. Based on the original assumption, the attacker has reason to believe that this attack model can effectively judge data similar to it (i.e., the real members and non-members).

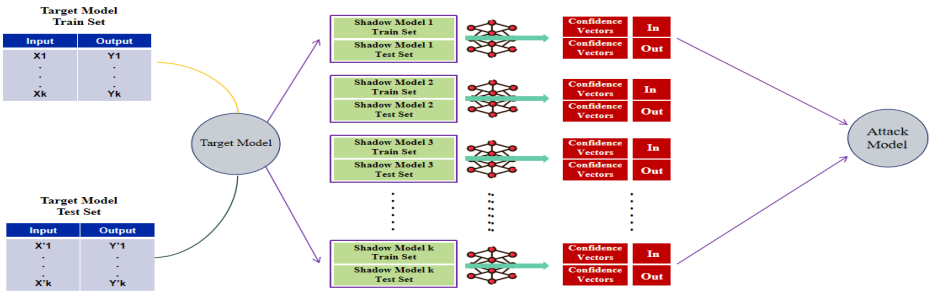
Although SM MIA for the first time innovatively uses the technique of training shadow models to enable attackers to transform the distribution of unknown member sets and non-member sets into judgments of known shadow training sets and test sets, its complex and cumbersome process will cause relatively high attack costs for attackers. In the original form of SM MIA proposed by Shokri et al, the attacker would need to even train multiple shadow models (up to hundreds in their case). As a result, many attackers choose to pursue lower-cost, easy-to-carry out member inference attacks category-metric based MIA.

Name of MIA	Abbreviation	Evaluation Context
Shadow model based MIA	SM MIA	ML / FL
Prediction Correctness based MIA	PC MIA	ML / FL
Prediction Loss based MIA	PL MIA	ML
Prediction Maximal Confidence based MIA	PMC MIA	ML / FL
Prediction Entropy based MIA	PE MIA	ML / FL
Prediction Sensitivity based MIA	PL MIA	ML
Class Label Confidence based MIA	CLC MIA	ML / FL
Class Label Entropy based MIA	CLE MIA	ML / FL
Modified Class Label Entropy based MIA	MCLE MIA	ML / FL
Source Inference Attack	SIA	ML / FL

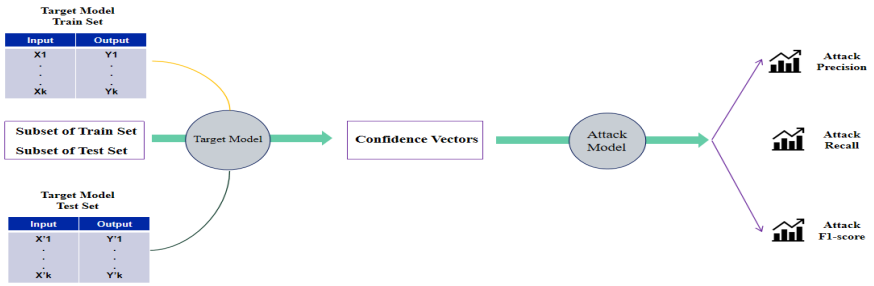
Table 5.1: Summary of Implemented MIAs in this work.

Algorithm 1 Shadow Model Based Membership Inference Attack

Require: M_T: Target model
$\mathcal{D}_{S_{\text{train}}}$, $\mathcal{D}_{S_{\text{test}}}$: Training and testing datasets for shadow models
$\mathcal{D}_A$, $\mathcal{L}_A$: Data and labels for training the attack model
N: Number of shadow models
Ensure: M_A: Trained attack model
Train Shadow Models
for $i = 1$ to N **do**
  Initialize M_{S_i} with the same architecture with M_T
  Train M_{S_i} on $\mathcal{D}_{S_{\text{train}}}$
  Collect predictions:
  $\mathcal{P}_{S^i_{\text{train}}} = M_{S_i}.\text{predict}(\mathcal{D}_{S_{\text{train}}})$
  $\mathcal{P}_{S^i_{\text{test}}} = M_{S_i}.\text{predict}(\mathcal{D}_{S_{\text{test}}})$
end for
Prepare Attack Model Data
Initialize empty sets $\mathcal{D}_{A_{\text{train}}}$ and $\mathcal{L}_{A_{\text{train}}}$
for $i = 1$ to N **do**
  Append $\mathcal{P}_{S^i_{\text{train}}}$ to $\mathcal{D}_{A_{\text{train}}}$ with labels 1 (member)
  Append $\mathcal{P}_{S^i_{\text{test}}}$ to $\mathcal{D}_{A_{\text{train}}}$ with labels 0 (non-member)
end for
Train Attack Model
Initialize M_A with an appropriate architecture
Train M_A on $\mathcal{D}_{A_{\text{train}}}$ with labels $\mathcal{L}_{A_{\text{train}}}$
Perform Membership Inference Attack
for each $x \in \mathcal{D}_A$ **do**
  Predict using M_T: $p_T = M_T.\text{predict}(x)$
  Infer membership using M_A: $is\_member = M_A.\text{predict}(p_T)$
end for



(a) Shadow Model Training



(b) SM MIA Attack

Figure 5.1: Process of Performing SM MIA.

5.2 Metric Based MIAs

Compared with the shadow model-based MIA, a major advantage of metric based MIA is that it is simple and easy to implement. Compared with the former, which requires the preparation of a series of models and data conditions, metric based MIA only needs to select a metric value and determine a valid threshold value, and then compare the calculated value in the evaluation set with the threshold value to make membership inferences. It can be seen that the effectiveness of metric-based Membership Inference Attacks (MIAs) often depends on the comparative threshold value selected by the attacker. Next, the selection logic of the threshold value for each attack will be discussed and analyzed.

PC MIA This MIA is the simplest and easiest attack method. The attacker determines the in and out samples by judging whether the true label of the evaluated dataset is consistent with the model prediction. This attack itself does not involve the selection of a specific threshold value, because the data label is an inherent feature of itself.

PL MIA This MIA uses the average loss during model training as the threshold value to distinguish between member sets and non-member sets. Although this approach is reasonable, because a trained model does produce lower prediction losses when facing its own training set, in practice, the average loss of model training is not necessarily easy for an attacker to obtain. This MIA itself is in a relatively white-box attack mode, which limits the possibility of many scenarios. In addition, because the mean of the model training loss is used as the boundary, this is bound to mean that there will inevitably be misestimated in samples, which makes the effect of this MIA have an upper limit.

PMC and PE MIA Both MIAs are from Salem et al.[11]. Although they use different metrics, one is to select the maximum confidence value in the prediction vector, and the other is to calculate entropy, they both use the same threshold value establishment method, that is, to obtain the threshold value for the optimal attack effect by generating random data in the same form as the target data set.

The exact threshold choosing method can be accomplished in this way: First, attacker try to generate a sample of random points in the feature space of the target data point. For example, if the target attack dataset is image data such as CIFAR-10, the attacker will create a series of random images where the value of each pixel is drawn from a uniform distribution. Given that the CIFAR-10 dataset format is 32 x 32 x 3, each image consists of 3072 pixels. For these 3072 points, each pixel value is randomly drawn from the range 0 to 255, corresponding to the possible pixel values in the dataset. Then after getting enough random images, the attacker would query these random points to the target model to get the corresponding metric values (maximal posteriors or entropies here). Because the authors believe that these newly-generated random data are sound substitutes for non-member samples, choosing their top t percentile of metrics can serve as a good threshold to perform attack. Although Salem et al. finally selected a fixed top 10 quantile as a benchmark for threshold selection for different datasets, in this work, a slightly different approach was taken for the thresholds of PMC and PE MIA.

In fact, no matter which percentile of the random metrics the attacker chooses as the final threshold, it will have two-fold impact. If this metric has a positive impact on reasoning, such as confidence applied here, then the smaller quantile often means that the attacker has chosen a more relaxed threshold, which will increase the value of attack recall (because more training samples are identified). But correspondingly, the precision will decrease, because relaxed conditions also mean an increase in false positives. Negative signs such as entropy are just the opposite. In reality, which percentile to choose depends on the attacker's goal, whether to focus more on the number or accuracy of inferences. Therefore, a compromise method is used here: by calculating the precision and recall for specific percentiles (10, 20, 30, ..., 90, increasing by 10 each time), the percentile with the largest F1-score is selected. This approach balances both aspects of attack performance effectively.

PS MIA PS MIA is an attack designed to capture the difference in prediction sensitivity between the training set and the non-member set. It is based on the observation that a well-trained machine learning model usually exhibit less sensitivity to the perturbations to the feature values.In essence, this is similar to the attack logic of PMC MIA, where the machine learning model becomes more confident in the predictions of its training data, and thus exhibits greater robustness to tiny changes in input values. In order to effectively quantify this concept, Liu et al. proposed to use the Jacobian matrix, which contains the first-order derivative of each output value of the model with respect to the input(5.1), to measure its sensitivity to changes in the input value. At the same time, in order to simulate a black-box attack, the author proposed an approximate solution to the Jacobian matrix of the model through an approximate estimation method, shown as 5.2. The advantage of this is that the attacker does not need information about the specific architecture of the training model, but only needs access to the model input and output to carry out this attack. In this way, the second-order norm value of the Jacobian matrix

of the target model for each data point can be obtained. This metric is a negative sign like entropy, and members of the training set tend to have lower Jacobian norm values.

$$\mathbf{J}(\mathbf{x};\mathcal{M}) = \begin{bmatrix} \frac{\partial \mathcal{M}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \mathcal{M}(\mathbf{x})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}, \tag{5.1}$$

where $\mathbf{y} = \mathcal{M}(\mathbf{x})$. The input sample is $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, and the corresponding prediction is $\mathbf{y} = [y_1, y_2, \ldots, y_m]$. $\frac{\partial y_i}{\partial x_j}$ represents the first derivative between the input sample's ith feature value and its prediction confidence to jth class.

$$\frac{\partial y_j}{\partial x_i} \approx \frac{\mathcal{M}(\mathbf{x}+\epsilon) - \mathcal{M}(\mathbf{x}-\epsilon)}{2\epsilon}, \tag{5.2}$$

where ϵ is a tiny value added to the ith feature value of the input sample.

Apart from this unique metric value, compared to the above-mentioned MIA, PS MIA does not select a real threshold value in a general sense. Instead, it divides all data sets into two categories, in group and out group, by applying clustering algorithms in unsupervised learning. For example, Liu et al. chose spectral clustering for the final division. This method applies the characteristics of unsupervised learning, which is quite different from the first proposed MIA by Shokri et al.[10] that trained the attack model in a supervised learning environment.

CLC, CLE, and MCLE MIA The types of indicators selected by these three MIAs are essentially the same as the previous ones, except that the prediction confidence and entropy of the data are differentiated by different labels. However, among the implemented metric-based MIAs, these three MIAs, compared to others, apply another special method to determine the optimal threshold value. The attacker uses the idea of training a shadow model to obtain a tool that simulates the performance of the target model. Through the performance of the shadow model's metric in its shadow data set (the in, out relationship is known by the attacker), the attacker selects a threshold value that makes the attack effect the best, and transfers this value to the attack on the target model. In this way, the attacker can inherit the advantages of the shadow model training technology to make the attack more targeted, but at the same time, the necessary preparations and assumptions mentioned above also need to be maintained, which also increases the burden of MIA implementation.

Chapter 6

Evaluation

This chapter presents a comprehensive evaluation of the performance of different types of membership inference attacks in Table5.1 in the context of machine learning and federated learning. In machine learning, the primary focus of evaluation is on how different models, varying volumes of training data, and differing numbers of training epochs influence the effectiveness of attacks. Conversely, in federated learning, the analysis shifts to examining the impact of network topology changes and the number of training rounds. This chapter will first present the setup for the experiments, then list the specific experimental results, and finally conduct a comparative analysis of experiments under different circumstances.

6.1 Experimental Setup

This section outlines the experimental setup, detailing the datasets employed, the model architecture and training parameters, as well as the metrics used to evaluate MIA performance.

6.1.1 Datasets and Preprocessing

All experiments are conducted on the following three datasets: MNIST, Fashion-MNIST, and CIFAR-10. These datasets are not only used as benchmarks in the field of model training but are also widely referenced in MIA-related literature. Figure 6.1 shows the grayscale and RGB visualizations of these datasets.

- **MNIST[25]** is a freely accessible dataset that contains 70,000 images of handwritten digits, 60,000 images of the training set, and 10,000 images of the testing set. Each image is formatted as 28 x 28 and processed so that the digit is in the center. The MNIST dataset is a 10-class classification problem in which the task is to determine which digit between 0 and 9, inclusive, is present in a given image.

- **FMNIST[26]** is a dataset that consists of 70,000 images of Zalando's article images, 60,000 images of the training set, and 10,000 images of the testing set. Each image is a 28 x 28 grayscale image associated with a label from 10 classes.
- **CIFAR-10[27]** dataset is also freely accessible and contains 60,000 color images, 50,000 images of the training set, and 10,000 images of the test set. Each image is again formatted to be 32 x 32. There are also ten classes in the CIFAR-10 dataset: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class has 6,000 images available. The problem is a 10-class classification problem to determine which of the ten classes is depicted in a given image.

(a) MNIST (b) Fashion-MNIST (c) CIFAR-10

Figure 6.1: Visualization of MNIST, Fashion-MNIST, and CIFAR-10 datasets.

In addition to these three original datasets, the impact of data augmentation on model training is also considered. **Data augmentation** is commonly applied to complex datasets to enhance the robustness and generalization capabilities of machine learning models[28]. By artificially increasing the diversity of the training data through various transformations—such as rotation, scaling, flipping, and cropping to image data—models can learn to recognize patterns and features more effectively, which helps reduce overfitting. As discussed in the previous chapter, the level of overfitting significantly influences the effectiveness of MIA. Therefore, it is both reasonable and meaningful to evaluate the impact of data augmentation in the following analysis.

Considering the characteristics of above three datasets, data augmentation measures are specifically applied to the CIFAR-10 dataset. Its RGB three-channel structure often hinders the model from fully learning image features during training, resulting in poor generalizability. Applying data augmentation to the CIFAR-10 dataset during training has become a widely accepted practice in the field of machine learning. Here, two different data augmentation methods are adopted to more comprehensively evaluate its impact on membership inference attacks:

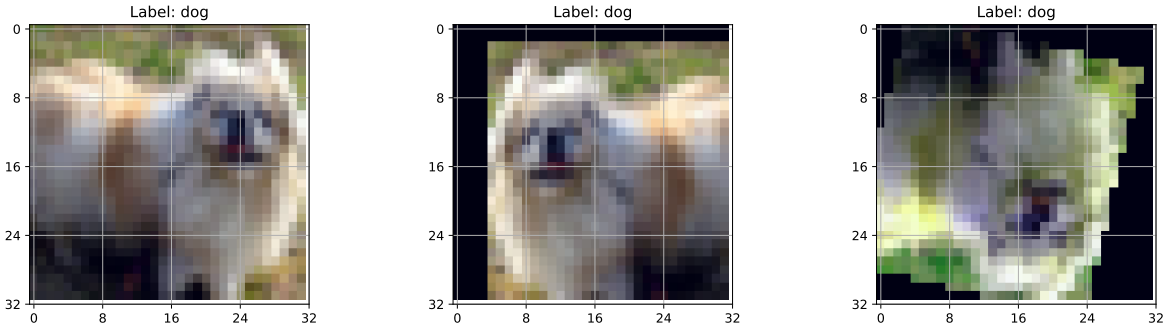
The first method includes the following transformations:

- Random cropping of images to 32 x 32 pixels with a padding of 4 pixels
- Random horizontal flipping

The second method incorporates additional operations:

- Random cropping of images to 32 x 32 pixels with a padding of 4 pixels
- Random horizontal flipping
- Random rotation of images by up to 15 degrees
- Color jittering to adjust brightness, contrast, saturation, and hue
- Random vertical flipping

Figure 6.2 illustrates the impact of two different data augmentation methods on original CIFAR-10 dataset images. It is evident that the second augmentation method is more intense than the first. This method not only alters the style and position of the images but also significantly changes their color and rendering compared to the original data. Regardless of the augmentation method used, the goal is to expose the model to a wider variety of data while preserving the essential features of the original images during training. This approach enhances the model's ability to generalize without compromising the core characteristics of the dataset.



(a) Standard (b) First Transformation (c) Second Transformation

Figure 6.2: Example of CIFAR-10 image with different augmentation methods.

In general, three datasets—MNIST, Fashion-MNIST, and CIFAR-10—are used to comprehensively evaluate the performance of membership inference attacks (MIA) in both machine learning and federated learning scenarios. This evaluation also includes the variants of the two data augmentation methods applied to the CIFAR-10 dataset. Detailed information on these datasets and augmentation methods is summarized in Table 6.1.

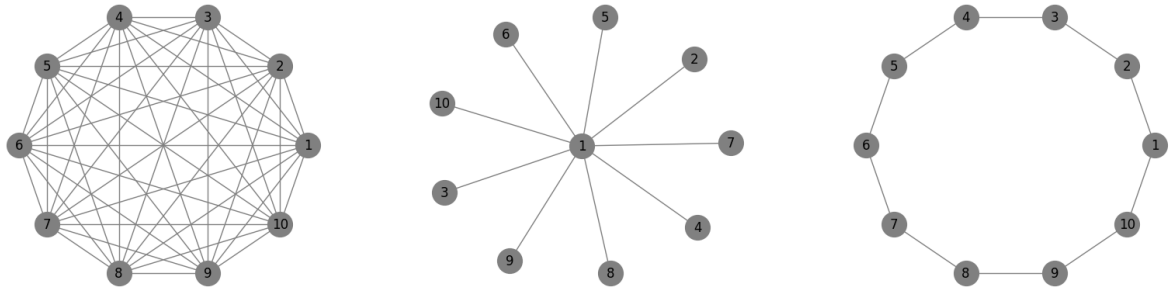
In the federated learning environment, different topological structures and data distribution forms (iid or non-iid) will cause differences in the data of each participant. The three network structures shown in Figure 6.3 are evaluated respectively in the experiments. Here, each participant is aggregated by the FedAvg algorithm. At the same time, considering whether the data distribution is independent and identically distributed, the data distribution of each node shown in Figure 6.4 is applied separately.

In the case of iid, each participant has a fixed number of 2500 training data and the number of data for each label is basically the same. However, considering non-iid, not only the amount of training data for each participant is very different, but the distribution

Name	Type	Number of Classes	Size of Training Records	Size of Test Records	Data Augmentation
MNIST	Image	10	60,000	10,000	No
FMNIST	Image	10	60,000	10,000	No
CIFAR-10no	Image	10	50,000	10,000	No
CIFAR-10aug	Image	10	50,000	10,000	First Aug
CIFAR-10extend	Image	10	50,000	10,000	Second Aug

Table 6.1: General information of the Experiment Datasets

of data labels is also extremely uneven, thus simulating the situation that may exist in the real environment. Here, this severe non-iid situation is achieved by using the Dirichlet distribution with alpha of 0.1.



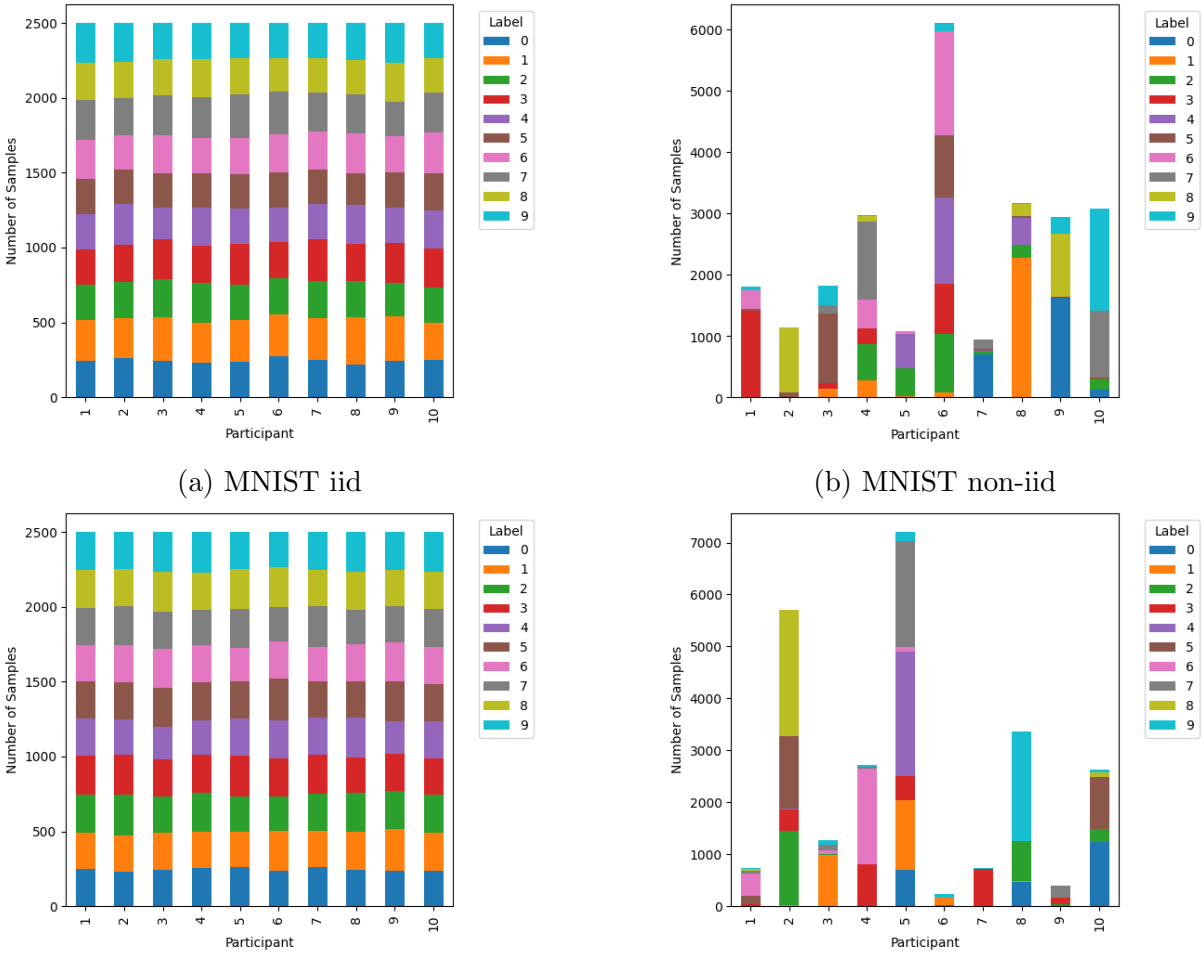
(a) Fully Connected (b) Star (c) Ring

Figure 6.3: Three Different Topologies for the Decentralized Federated Learning with 10 participants.

6.1.2 Model Architectures

The experiments were conducted on a server equipped with an AMD EPYC 7702 64-Core Processor running at 1.996 GHz. The server architecture supports both 32-bit and 64-bit operations with a total of 16 cores per socket and 64 threads, distributed across a single socket.
For graphical processing, the server utilizes an NVIDIA Tesla T4 GPU, with driver version 545.23.06 and CUDA version 12.3.
The specific machine learning model involved in the experiment applied the Pytroch Lightning library. Considering the different complexities of the three kinds of experimental datasets, the following two model architectures were selected and handled respectively:

- **Multilayer Perceptron (MLP)** One multilayer perceptron model was chosen for MNIST and FMNIST classification tasks. It consists of an input layer that flattens 28x28 pixel images into 784-dimensional vectors, followed by two fully connected layers with 256 and 128 neurons respectively, each with ReLU activation functions. The final layer is a fully connected layer with 10 neurons for class prediction, using



(a) MNIST iid

(b) MNIST non-iid

(c) CIFAR-10 iid

(d) CIFAR-10 non-iid

Figure 6.4: IID and non-IID distribution of MNIST and CIFAR-10 datasets for each participant in DFL

Cross-Entropy Loss for optimization and the Adam optimizer with a learning rate of 0.001.

- **Convolutional Neural Network (CNN)** As for CIFAR-10 class datasets, one convolutional neural network model was used. The implemented CNN model comprises three convolutional layers with 16, 32, and 64 filters respectively, each followed by ReLU activation and 2x2 max-pooling. The output from the convolutional layers is flattened and passed through a fully connected layer with 512 neurons and a final output layer with 10 neurons for class prediction. Besides, the model uses Cross-Entropy Loss for optimization, employs the Adam optimizer with specific beta parameters (0.851436 and 0.999689) and AMSGrad.

In the case of federated learning, all participants use the same model type to train local datasets, regardless of whether the dataset distribution is iid or non-iid. This can help the overall model converge and make the evaluation of the performance of MIA performed from each node more objective.

6.1.3 Evaluation Metrics

In order to objectively evaluate the performance of MIAs in various situations, the following metrics are selected to measure the quality of MIA from different perspectives.

- **Attack Precision (AP)** Attack Precision measures the proportion of true positive memberships (correctly identified as members) out of all instances that were predicted as members by the attack. It is defined as:

$$\text{AP} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.1}$$

  where TP is the number of true positives and FP is the number of false positives. A higher precision indicates that the attack model is more accurate in identifying true members without incorrectly labeling non-members as members.

- **Attack Recall (AR)** Attack Recall measures the proportion of true positive memberships out of all actual members. It is defined as:

$$\text{AR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.2}$$

  where FN is the number of false negatives. A higher recall indicates that the attack is more effective at detecting members from the actual member population, ensuring fewer actual members are missed.

- **F1-Score** The F1-Score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns. The F1 Score is defined as:

$$\text{F1-Score} = 2 \times \frac{\text{AP} \times \text{AR}}{\text{AP} + \text{AR}} \tag{6.3}$$

  A higher F1 Score indicates a better balance between AP and AR, making it a comprehensive metric for overall attack performance.

- **AUC Score** The AUC score is defined as the area under the ROC curve, which plots the true positive rate against the false positive rate. This metric aims to measure the attack's ability to distinguish between members and non-members across all threshold values. A higher AUC score (ranging from 0 to 1) indicates a better overall performance of the attack in distinguishing members from non-members, with a score of 0.5 representing random guessing and 1.0 representing perfect discrimination.

6.2 Experimental Results

This section specifically presents the performance results of different types of membership inference attacks in various situations in the context of machine learning and federated learning.

6.2.1 Machine Learning Case

Table 6.2 first shows the attack performance of shadow model based MIA (SM MIA) under the CIFAR-10 dataset without data augmentation. It shows the difference in attack performances caused by training different numbers of shadow models to carry out attacks. It can be seen that although the number of shadow models has increased by 10 times, the improvement in both attack precision and attack recall is very small. This verifies Salem et al[11]'s point of view - that training a single shadow model can achieve similar results to multiple shadow models. At the same time, this attack method is also evaluated under different training data size and different training epochs of the target model, which are both important factors affecting the level of model overfitting. By experimenting with different training data sizes and epochs, the connection between the reasons for MIA's success and the level of overfitting can be better established.

Based on the CIFAR10 experience, it has been demonstrated that simply increasing the number of shadow models does not enhance the effectiveness of the SM MIA. Therefore, to save computing resources, all SM MIAs evaluated subsequently use the setting of training only one shadow model. In this case, Table 6.3 illustrates the performance of SM MIA on MNIST and FMNIST datasets.

Next, when evaluating the performance of MIA on the dataset after data augmentation, it is necessary to consider whether the attacker knows that the target model uses data augmentation during training. This can be broken down into the following cases:

- **Adversary 1.** This attacker knows that the target model uses data augmentation to improve training performance. So it also takes corresponding measures when training the shadow model to mimic behaviour. However, due to practical limitations, it is not possible to obtain the original data form to evaluate the effect of the attack. This situation is very likely in reality. Assuming that the attacker's target is MLAS (Machine Learning as a Service), the platform that provides such services often only provides an API for users to obtain data, so that the attacker can only get the randomly transformed dataset.

- **Adversary 2.** This attacker also knows that data augmentation is used to train the target model, but unlike the first adversary, it has access to the original form of the training data.

- **Adversary 3.** The attacker is completely unaware that data augmentation is used for target model training. Therefore, based on its cognition, it will only choose to use the original data form to train the shadow model and evaluate the effectiveness of the attack.

The above adversary assumption is summarized in Table6.4 with regard to different shadow set and evaluation set. Based on this setting, Table6.5, Table7.1 and Table7.2 illustrate the performance of SM MIA on CIFAR-10aug and CIFAR-10extend datasets under different adversary assumptions respectively. Additionally, as analyzed in Chapter5, the three attacks—CLC MIA, CLE MIA, and MCLE MIA—also require training a

Number of Shadow Models	Size of Training Data	Epochs	AP	AR	F1-Score	Overfitting Level	AUC
k = 1	n= 2500	10	0.507	0.646	0.568	15.02%	0.516
		25	0.637	0.898	0.745	46.30%	0.737
		50	**0.688**	0.995	**0.814**	48.60%	0.800
		75	0.680	**0.999**	0.809	48.50%	**0.809**
		100	0.686	0.997	0.812	48.60%	0.807
	n=5000	10	0.505	0.573	0.537	11.88%	0.512
		25	0.613	0.873	0.720	43.21%	0.719
		50	**0.670**	0.981	0.796	43.70%	0.782
		75	0.667	**0.998**	0.800	44.05%	0.786
		100	0.669	0.996	**0.801**	43.80%	**0.792**
	n = 12500	10	0.531	0.516	0.524	14.75%	0.538
		25	0.638	0.940	0.760	35.70%	0.735
		50	0.648	**0.984**	**0.781**	35.66%	0.746
		75	**0.652**	0.971	0.780	35.76%	0.750
		100	0.647	0.953	0.771	35.80%	**0.755**
	n = 25000	10	0.526	0.704	0.602	17.26%	0.545
		25	0.583	0.987	0.733	30.78%	0.689
		50	0.621	0.971	0.757	30.41%	0.720
		75	**0.623**	0.883	0.730	30.52%	0.721
		100	0.621	**0.989**	**0.763**	30.53%	**0.722**
k = 10	n= 2500	10	0.534	0.667	0.593	15.02%	0.543
		25	0.664	0.901	0.765	46.30%	0.764
		50	**0.715**	0.994	0.832	48.60%	0.827
		75	0.707	0.999	0.828	48.50%	**0.836**
		100	0.713	**0.999**	**0.832**	48.60%	0.835
	n=5000	10	0.532	0.556	0.544	11.88%	0.539
		25	0.640	0.878	0.740	43.21%	0.746
		50	**0.697**	0.992	0.819	43.70%	0.809
		75	0.694	0.997	0.819	44.05%	0.813
		100	0.696	**0.999**	**0.821**	43.80%	**0.819**
	n = 12500	10	0.558	0.535	0.546	14.75%	0.565
		25	0.665	0.931	0.776	35.70%	0.762
		50	0.675	0.947	0.788	35.66%	0.773
		75	**0.679**	0.963	0.797	35.76%	0.777
		100	0.674	**0.995**	**0.803**	35.80%	**0.782**
	n = 25000	10	0.553	0.738	0.633	17.26%	0.572
		25	0.610	0.945	0.741	30.78%	0.716
		50	0.648	0.926	0.762	30.41%	0.747
		75	**0.650**	0.932	**0.770**	30.52%	0.748
		100	0.648	**0.999**	0.757	30.53%	**0.750**

Table 6.2: Performance of SM MIA with Different Number of Shadow Models under CIFAR-10no Dataset.

Dataset	Size of Training Data	Epochs	AP	AR	F1-Score	Overfitting Level	AUC
MNIST	n= 2500	10	0.513	0.750	0.609	3.24%	0.521
		25	0.527	**0.816**	**0.640**	6.98%	0.554
		50	0.535	0.601	0.566	7.10%	0.562
		75	0.537	0.270	0.360	7.00%	0.581
		100	**0.557**	0.460	0.504	6.90%	**0.587**
	n=5000	10	0.504	0.807	0.620	2.25%	0.511
		25	0.519	0.847	0.644	5.92%	0.538
		50	0.527	**0.962**	**0.681**	5.50%	0.550
		75	**0.531**	0.671	0.593	5.35%	0.557
		100	0.530	0.902	0.667	5.35%	**0.562**
	n = 12500	10	0.504	0.655	0.569	1.93%	0.503
		25	0.514	0.756	0.612	3.00%	0.526
		50	0.518	**0.986**	**0.679**	2.98%	0.538
		75	**0.520**	0.898	0.658	2.98%	**0.548**
		100	0.519	0.690	0.592	3.02%	0.539
	n = 25000	10	0.504	0.954	0.659	1.15%	0.503
		25	0.506	0.848	0.634	2.31%	0.513
		50	0.509	**0.971**	**0.668**	2.80%	0.517
		75	**0.509**	0.794	0.621	2.30%	0.518
		100	0.509	0.899	0.650	2.25%	**0.521**
FMNIST	n= 2500	10	0.500	0.568	0.532	6.32%	0.500
		25	0.526	0.614	0.567	14.02%	0.550
		50	0.558	**0.812**	**0.662**	16.50%	0.607
		75	**0.573**	0.779	0.660	16.80%	0.623
		100	0.571	0.778	0.659	16.80%	**0.631**
	n=5000	10	0.509	0.499	0.504	7.69%	0.515
		25	0.516	0.747	0.611	11.07%	0.537
		50	0.537	0.697	0.607	15.27%	0.575
		75	0.534	**0.848**	**0.655**	15.88%	0.573
		100	**0.552**	0.756	0.638	15.60%	**0.599**
	n = 12500	10	0.504	0.616	0.554	5.06%	0.506
		25	0.510	0.847	0.637	9.90%	0.523
		50	0.523	0.808	0.635	12.90%	0.551
		75	**0.539**	0.804	0.645	**13.92%**	0.583
		100	0.538	**0.996**	**0.699**	13.34%	**0.594**
	n = 25000	10	0.502	0.581	0.539	3.97%	0.502
		25	0.511	0.844	0.637	8.67%	0.523
		50	0.518	**0.910**	**0.660**	10.60%	0.544
		75	0.518	0.884	0.653	12.04%	0.535
		100	**0.529**	0.726	0.612	11.44%	**0.557**

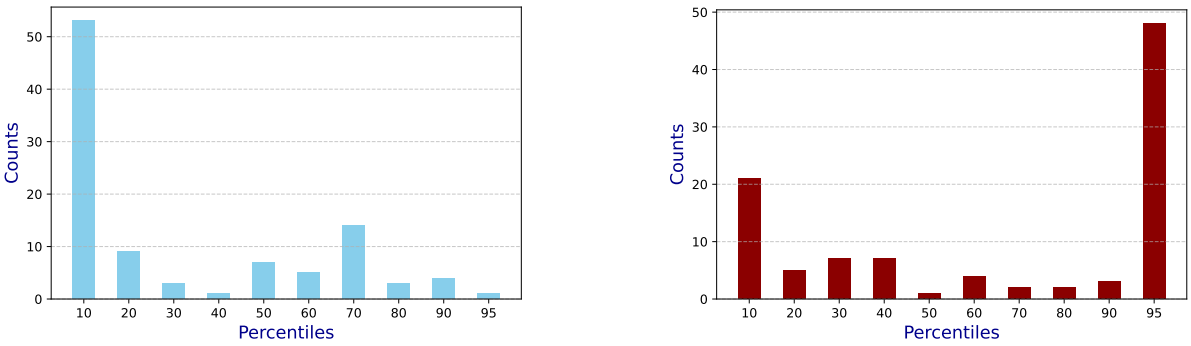
Table 6.3: Performance of SM MIA with a Single Shadow Model under MNIST and FMNIST Dataset.

Adversary	Knowledge of Data Augmentation Used	Shadow Model Training Set	Evaluation Set
Adversary 1	Yes	Yes	Yes
Adversary 2	Yes	Yes	No
Adversary 3	No	No	No

Table 6.4: Assumptions for Each Adversary Regarding the Use of Data Augmentation in Shadow Model Training and Evaluation.

shadow model to determine the optimal metric comparison boundary. Consequently, the same adversary analysis is conducted on these attacks. Figures 4, 5, and 6 present the corresponding results.

As for other metric based MIAs which does not involve using shadow model, their attack performance results across different datasets can be found in Figure6.7, Figure7.5, Figure7.6, Figure7.7, and Figure7.8. Among them, PMC and PE MIA are worth exploring further in depth for the threshold values selected each time. As discussed in chapter5,their threshold comparison values are generated by creating random data. Each time, the attacker would traverse trying different percentiles and use the one with the best F1-score at last. The cases where each quantile is selected as the optimal threshold in all datasets are counted here shown in Table6.8. It can be seen that PMC MIA mainly selects the 10th quantile, while PE MIA selects the 95th quantile as the best threshold. The values corresponding to these two quantiles generally mean that the attacker has chosen a looser boundary value to compare whether a data point is a training member(since the entropy is a negative sign and the smaller the entropy value of a data point, the higher the probability that it is a training member). The impact of this is that compared to stricter control of false positives, increasing the number of true positives can make the overall attack more effective. At the same time, although this method of selecting threshold values based on random data quantiles is only tested in the experimental dataset of this work, considering the huge difference between it and other quantile counts, it has great potential to be migrated to a wider datasets to form a paradigm of attack method.



(a) PMC MIA

(b) PE MIA

Figure 6.8: Total Counts of Different Percentiles Chosen for PMC MIA and PE MIA Across All Datasets

Dataset	Size of Training Data	Epochs	AP	AR	F1-Score	Overfitting Level	AUC
CIFAR-10 aug	n= 2500	10	0.506	0.630	0.561	-0.04%	0.503
		25	0.510	0.364	0.425	5.20%	0.507
		50	0.522	0.607	0.561	14.30%	0.522
		75	0.542	0.658	0.594	23.96%	0.570
		100	0.592	0.719	0.650	33.18%	0.637
	n=5000	10	0.510	0.380	0.436	-1.68%	0.516
		25	0.506	0.576	0.539	1.28%	0.517
		50	0.520	0.686	0.591	10.35%	0.535
		75	0.542	0.593	0.566	16.72%	0.565
		100	0.567	0.601	0.584	24.05%	0.597
	n = 12500	10	0.502	0.615	0.553	-0.80%	0.503
		25	0.505	0.688	0.582	3.24%	0.511
		50	0.522	0.692	0.595	10.41%	0.539
		75	0.543	0.719	0.619	15.93%	0.571
		100	0.550	0.747	0.634	19.69%	0.588
	n = 25000	10	0.497	0.570	0.531	-1.67%	0.496
		25	0.495	0.470	0.482	3.10%	0.497
		50	0.510	0.808	0.625	8.36%	0.516
		75	0.523	0.743	0.614	11.37%	0.540
		100	0.528	0.779	0.629	13.22%	0.551
CIFAR-10 extend	n= 2500	10	0.503	0.370	0.426	-1.58%	0.511
		25	0.519	0.440	0.476	3.28%	0.515
		50	0.509	0.452	0.479	3.28%	0.506
		75	0.510	0.609	0.555	8.56%	0.521
		100	0.516	0.524	0.520	9.40%	0.521
	n=5000	10	0.503	0.724	0.594	-1.90%	0.508
		25	0.510	0.254	0.339	-2.44%	0.507
		50	0.500	0.367	0.423	-1.38%	0.501
		75	0.516	0.439	0.475	2.12%	0.520
		100	0.523	0.346	0.417	5.34%	0.521
	n = 12500	10	0.499	0.581	0.537	-4.54%	0.499
		25	0.515	0.280	0.363	-2.56%	0.509
		50	0.506	0.531	0.518	-0.56%	0.509
		75	0.506	0.618	0.557	2.05%	0.509
		100	0.518	0.611	0.561	1.62%	0.523
	n = 25000	10	0.500	0.567	0.531	-2.38%	0.499
		25	0.493	0.414	0.450	-3.32%	0.495
		50	0.489	0.393	0.436	-1.75%	0.492
		75	0.491	0.443	0.466	-0.32%	0.489
		100	0.495	0.500	0.497	0.35%	0.494

Table 6.5: Performance of SM MIA for CIFAR-10aug and CIFAR-10extend Dataset based on Adversary 1 Assumption.

Dataset	Name of MIA	Fully Connected		Star		Ring	
		AP	AR	AP	AR	AP	AR
CIFAR-10no	SM MIA	0.535	0.293	0.535	0.303	0.529	0.274
	PC MIA	0.564	0.848	0.540	0.710	0.542	0.723
	PMC MIA	0.538	0.424	0.503	0.928	0.523	0.447
	PE MIA	0.535	0.509	0.504	0.907	0.519	0.564
	CLC MIA	0.549	0.276	0.552	0.271	0.540	0.246
	CLE MIA	0.534	0.264	0.535	0.272	0.528	0.246
	MCLE MIA	0.549	0.280	0.553	0.274	0.541	0.249
	Avg.	0.543	0.414	0.532	0.524	0.532	0.393
CIFAR-10aug	SM MIA	0.499	0.819	0.500	0.791	0.499	0.815
	PC MIA	0.518	0.758	0.517	0.711	0.518	0.706
	PMC MIA	0.510	0.557	0.509	0.527	0.509	0.507
	PE MIA	0.507	0.617	0.509	0.535	0.509	0.502
	CLC MIA	0.519	0.593	0.519	0.595	0.521	0.588
	CLE MIA	0.510	0.351	0.512	0.414	0.512	0.362
	MCLE MIA	0.522	0.620	0.519	0.590	0.521	0.589
	Avg.	0.512	0.616	0.512	0.595	0.513	0.581
CIFAR-10extend	SM MIA	0.535	0.293	0.535	0.303	0.529	0.274
	PC MIA	0.507	0.622	0.508	0.602	0.507	0.607
	PMC MIA	0.501	0.636	0.502	0.679	0.502	0.741
	PE MIA	0.502	0.519	0.503	0.482	0.501	0.752
	CLC MIA	0.507	0.727	0.505	0.736	0.506	0.715
	CLE MIA	0.499	0.691	0.501	0.655	0.502	0.563
	MCLE MIA	0.506	0.698	0.506	0.743	0.506	0.734
	Avg.	0.508	0.598	0.509	0.600	0.508	0.627

Table 6.6: Performance of Different MIAs in the Final Round (Round 10) Across Various Datasets and Topologies.

6.2.2 Federated Learning Case

In the context of federated learning, different attack methods are evaluated under different DFL topologies. In all experiments, federated learning is conducted for 10 rounds. After each round of aggregation, different attacks are launched by various participants to evaluate the effects. Table6.6 presents the results of different MIAs across different datasets and topologies. The performance value of each attack is the average value evaluated by all participants in the federation in this round. Only the results of the last round are shown here because the model generally converges to the highest degree at this time, and is closer to the results of large-scale data training in traditional machine learning. In fact, in the first few rounds of federated learning, the model information was not fully transmitted between different participants, resulting in a relatively random performance of the aggregated model, which caused great trouble for the launch of MIA.

Although Table6.6 provides a perspective to examine the MIA effect in different types of federated learning, the way calculating the average performance of all participants in a round invisibly ignores the uniqueness of the different participants. Compared with this

macro perspective, Figure6.9, Figure6.11, and Figure6.12 evaluates the different nodes themselves at a micro level. These tables depict how the MIA attack launched from a certain node can be used to infer the training data of each node in the entire federation. The horizontal axis represents node1 to node10, and the vertical axis represents round1 to round10. Through the changes in color depth in the heat map, it can be seen that different nodes in different topologies infer different amounts of training data for other nodes. This change is closely related to the location of the node and the topology of the overall network. This is also a special feature of MIA in DFL, because the aggregation model of each participant in DFL is different in different time periods.

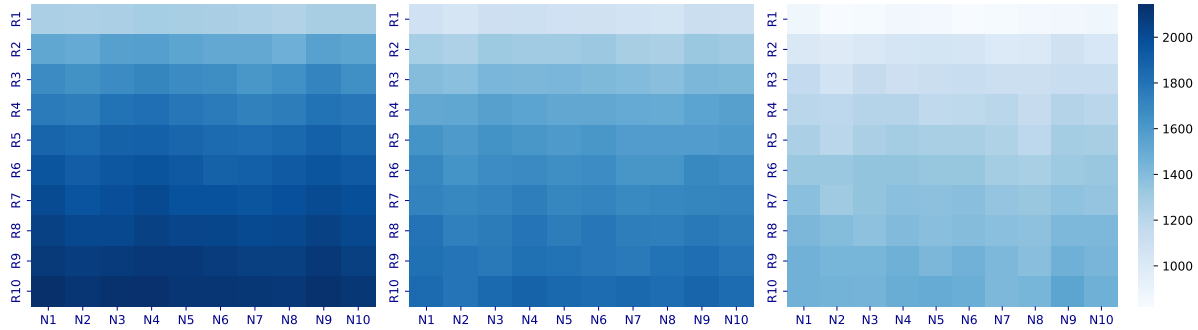


Figure 6.9: Heatmap of AR of PC MIA from Node 1 in fully connected network across CIFAR-10no, CIFAR-10aug, and CIFAR-10extend

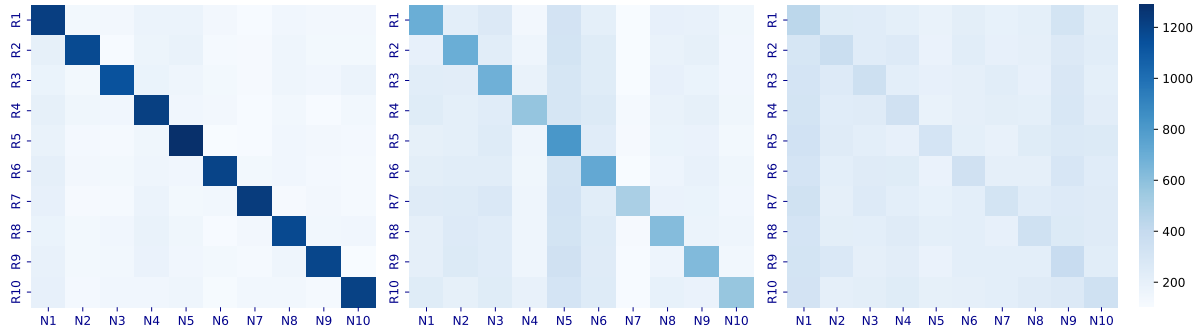


Figure 6.10: Heatmap of AR of SIA in the Final Round (Round 10) from Different Nodes across CIFAR-10no, CIFAR-10aug, and CIFAR-10extend Dataset.

6.3 Comparison Analysis

6.3.1 ML vs FL

First, overall, compared to the performance of various MIAs in machine learning environments, the performance of all MIAs in FL has been greatly reduced. Regardless of the

topology, the attack precision has basically dropped to nearly 50%, which means that the effect of this attack is no better than random guessing. At the same time, the drop in attack recall is even more significant, and the recall value of many MIAs has even dropped to 20% to 30%, which means that it has lost any reasoning function.

At the same time, compared with the performance in machine learning, the performance of MIA in FL is also extremely random. For example, in machine learning, as the number of model training epochs increases, the overall performance of MIA will increase. However, in FL, it can be found that the performance of many MIAs does not seem to be equal to the number of rounds. Even after the last round of aggregation, in theory, the models of each node should gradually converge and become homogeneous, but at this time, the performance of many MIAs will not show a significant increase compared to the previous rounds. This makes the method of controlling MIA more difficult.

The reasons why MIA failed in FL are mainly the following:

- **Reduction of overall model overfitting level**
  As discussed above, an important factor in the success of MIA is that the model has poor generalization ability and a large overfitting level. However, in federated learning, as different participants gradually aggregate each other's model parameters, the poor generalization ability of the original model training is reduced invisibly, which is also one of the advantages of federated learning. Although there is currently no unified calculation method for the overfitting value of the overall model in each round of federated learning, considering that in aggregation, the model can continue to be exposed to information from different data distributions, thereby enhancing the model's predictive ability in dealing with unknown data.

- **Difficulty to simulate the behavior of the federated model**
  Among the many MIAs tested, whether it is the MIA involving shadow model training or the MIA determining the threshold by generating similar random images, the attacker always tries to simulate the unknown part that the attacker wants to speculate by using the data set at hand where the in and out relationships are known. This is exactly what was discussed in Chapter5 that the essential purpose of the shadow model is to mimic the behavior of the target model, and the random data is generated with the assumption that it closely approximates the distribution of non-member values.

  But in the context of federated learning, the assumptions that these attacks work no longer hold true. When implementing a shadow model attack, the attacker can generally only train an ordinary machine learning model. Even if it keeps other conditions consistent, this model will have very different performances when facing the same data points compared to the model after each round of aggregation, resulting in the final trained binary classifier attack model being unable to correctly distinguish the relationship between member sets and non-member sets. This is why in the Table, the MIA effects involving training shadow models are particularly poor. This situation is very similar to the situation of Adversary 3 in the Table, that is, the performance of the target model cannot be correctly simulated, resulting in the final attack not being able to target this situation well. The same principle also

applies to the use of random data. Because in the unique environment of federated learning, a simple random image with each pixel obeying a standard distribution can no longer describe the distribution of non-member sets.

- **Complex Data Distribution among Participants**
  In actual federated learning, it is normal for the data distribution of each participant to be very different. In this case, for an individual with very small local data or very extreme data labels, it is very likely to obtain a model that does not fully converge at the end of federated learning, especially if it is located in a remote position in the entire network. In this case, it is quite difficult for an attacker to use the information obtained to infer the situation of the training set members in the entire federation.

But at the same time, although the MIA effects of most tests have deteriorated significantly, there is still one MIA that poses substantial risk of privacy leakage in the FL environment. It is PC MIA, which is inferred by whether the model prediction is correct. Its easy-to-implement feature makes it more adaptable to changes in the environment in FL. Unlike attacks such as shadow models that can be misled by incorrect attack information, its characteristics based on model prediction results make it more sensitive to changes in models of different participants. This can be seen from the examples in Figure 6.9 to Figure 6.11.

6.3.2 Different Topologies of FL

Although Table 6.6 shows the difference in the average values of all attacks implemented by different topologies, considering that the absolute values are already at a low level, it does not seem reasonable to evaluate the difference at this time. Therefore, looking at the participant composition of the training members inferred by MIA from a micro perspective is a better angle to analyze the differences between different topologies.

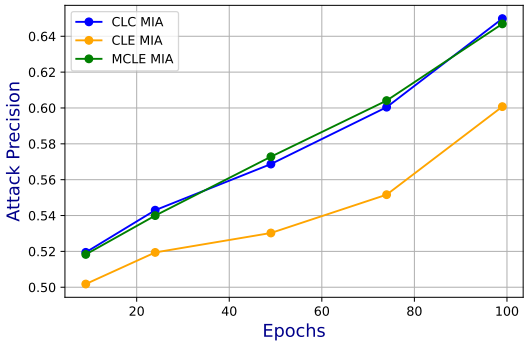
Figure 6.9 uses the form of heat change to depict the different changes in the proportion of each member in the federation inferred by node 1 in different rounds of PC MIA. First of all, from a vertical perspective, no matter which type of CIFAR-10 dataset is used, the overall trend of PC MIA is constantly improving. This is consistent with the nature of federated learning. As the degree of model convergence increases, the information inferred by the attack continues to increase. On the other hand, from a horizontal perspective, no matter which round node 1 is in, the proportion of each member's training data inferred is basically the same. It does not obtain more information for any specific node. This actually corresponds to the property of fully connected topology in which it is located. In this case, the model of each participant after each round of aggregation is the same, so that the reasoning ability of each node is basically identical.

From the Star network shown in Figure 6.10, it can be clearly seen that different attackers have very different reasoning abilities for different nodes. Node 1 is the center of the star graph, and it aggregates with all the remaining nodes in each round. Relatively speaking, it always has the strongest ability to reason about its own training data set, while the level of reasoning about other attachment nodes is basically the same, maintaining a lower

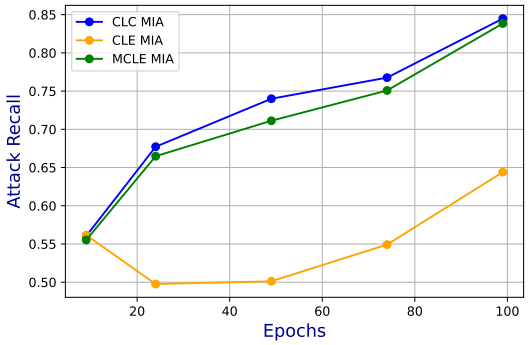
level. This may be because as a center point, its own model information is always used multiple times in aggregation, making it unable to break away from the influence of its own data. In addition, the performance of nodes 2, 3, and 4 is very similar. They always have strong reasoning abilities for themselves and their only neighbor, the center node 1, and know very little about other participants who are not directly connected. It can be clearly seen from Figure 6.10 that the topology of the star graph greatly affects the difference in MIA effects between different participants.

Figure 6.11 depicts the difference in inference results of different nodes in the ring graph. First, from the situation of these four nodes, it can be seen that the number of reasoning decreases from the self to the adjacent then to distant points, presenting an obvious ladder-like form. In other words, the distance between nodes in the ring graph greatly affects the reasoning ability of each participant to others. This is actually quite reasonable. Because each participant will aggregate with the neighbours directly connected to it in each round, in this case, the model information of the node close to it must have a greater impact on itself, while the information of the distant individuals needs to spend extra time to be brought to itself step by step through other nodes. This logic precisely reflects the change of the ladder state potential. In addition, unlike the star graph, the ring graph itself is a symmetrical graph, so there is no one that has a higher reasoning ability than any other nodes. This conclusion reflects that the topological structure of the network leads to different reasoning ability by affecting the aggregation behavior of each participant.

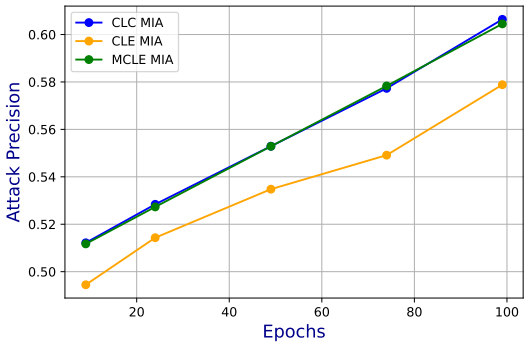
Finally, just as data augmentation reduces the effect of MIA in machine learning, in federated learning, data augmentation masks the network topology tendency discussed above. That is, for a participant, the membership composition it infers gradually tends to be uniform. This phenomenon can be clearly seen from the heatmaps between different data sets from Figure 6.9 to Figure 6.11. Obviously, from left to right, not only the number of members inferred as a whole is decreasing, but also the inferred targets no longer have obvious directionality. This result is also related to the conclusion that data augmentation can reduce the overall overfitting level of the model. In federated learning, after using data augmentation, the model influence of each participant gradually converges, so that MIA no longer has additional network topology reasoning functions as mentioned above. This also shows that the means of data augmentation can still protect the privacy leakage of MIA in a relatively profound way in federated learning.



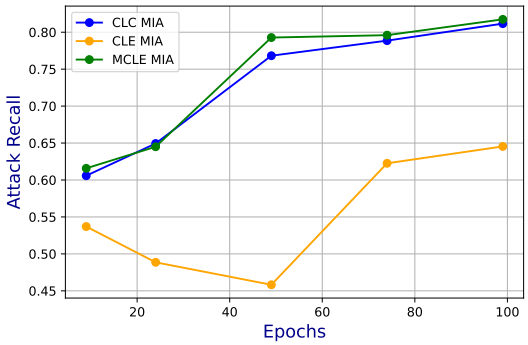
(a) AP with n = 2500



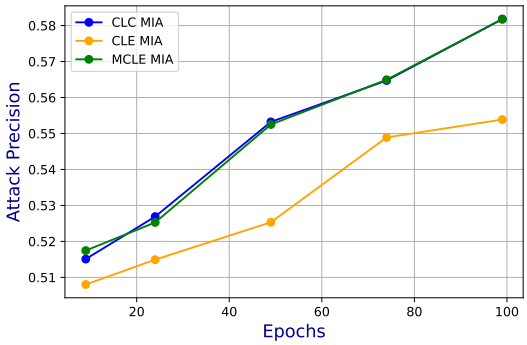
(b) AR with n = 2500



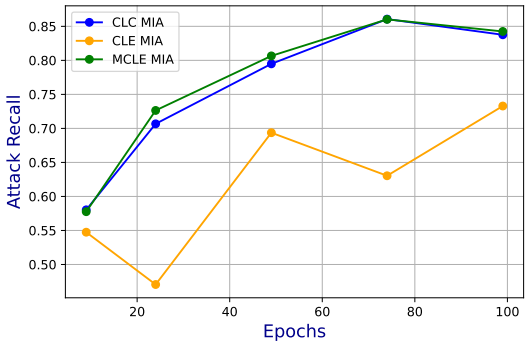
(c) AP with n = 5000



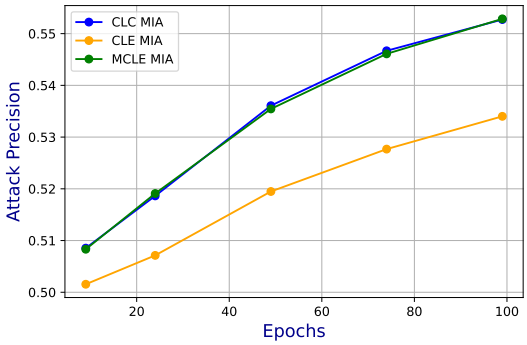
(d) AR with n = 5000



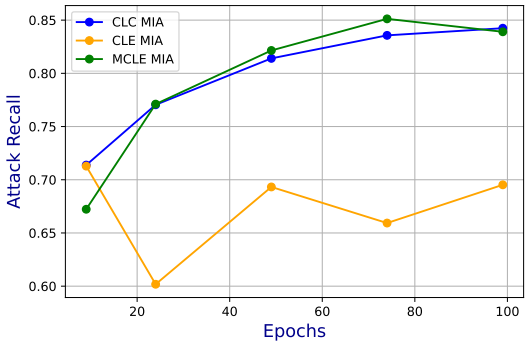
(e) AP with n = 12500



(f) AR with n = 12500

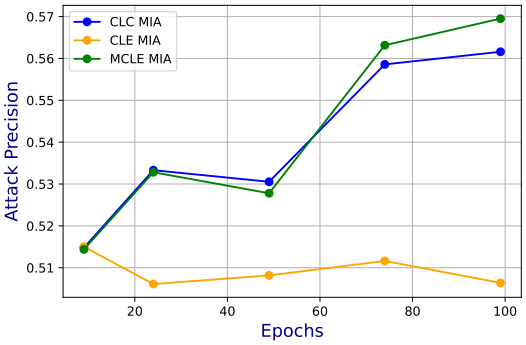


(g) AP with n = 25000

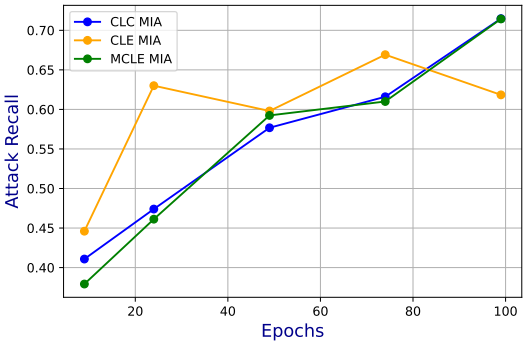


(h) AR with n = 25000

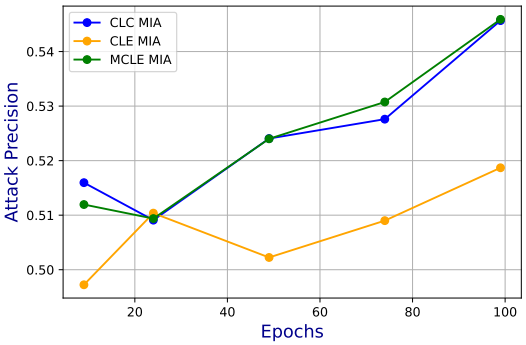
Figure 6.5: Performance of CLA, CLE, and MCLE MIA for the CIFAR-10aug Dataset based on Adversary 1 Assumption.



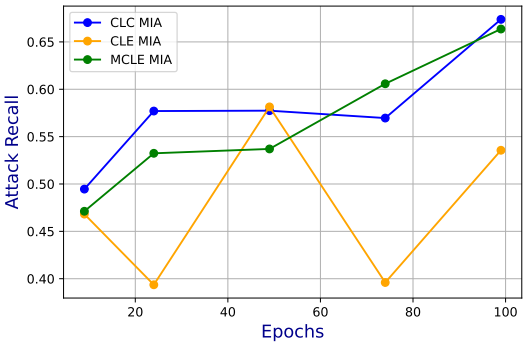
(a) AP with n = 2500



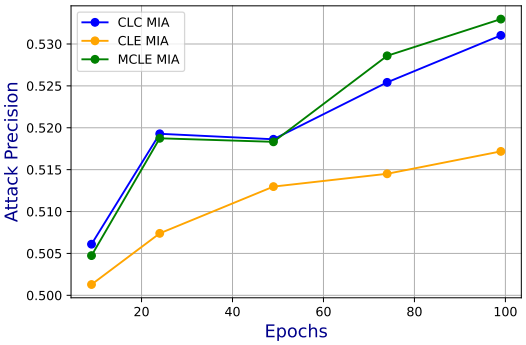
(b) AR with n = 2500



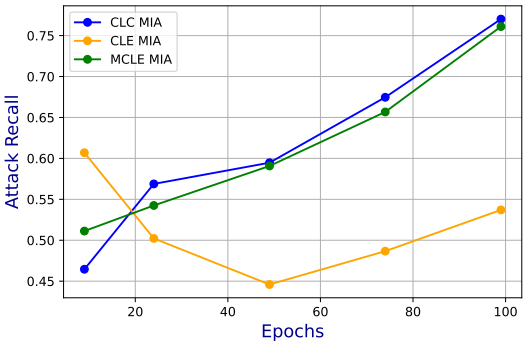
(c) AP with n = 5000



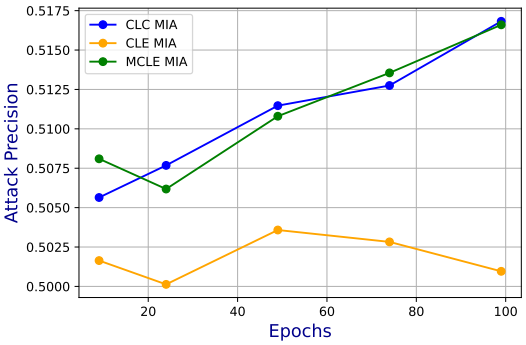
(d) AR with n = 5000



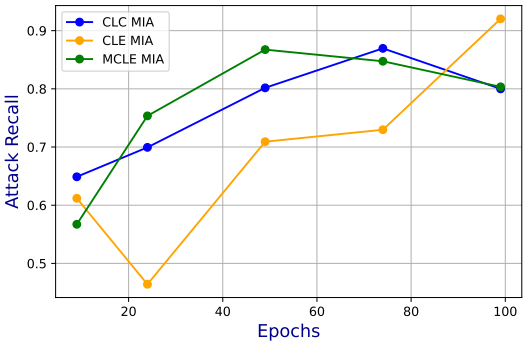
(e) AP with n = 12500



(f) AR with n = 12500

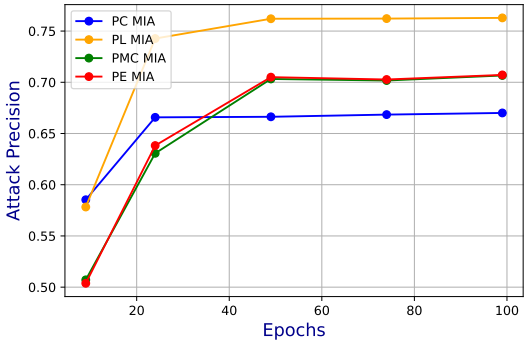


(g) AP with n = 25000

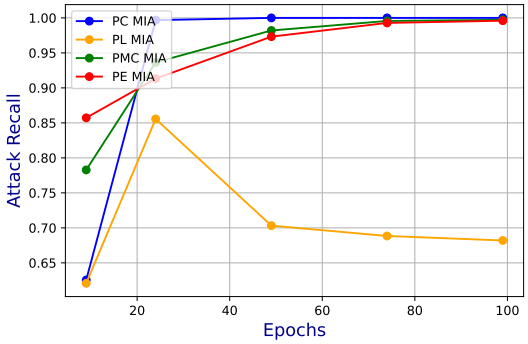


(h) AR with n = 25000

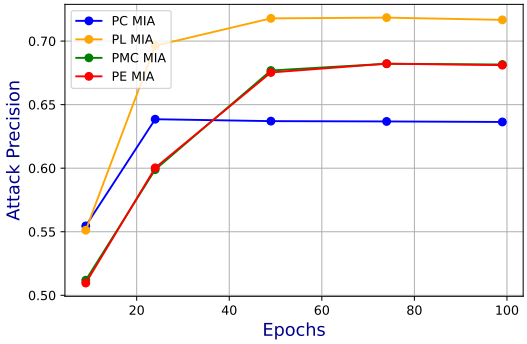
Figure 6.6: Performance of CLA, CLE, and MCLE MIA for the CIFAR-10extend Dataset based on Adversary 1 Assumption.



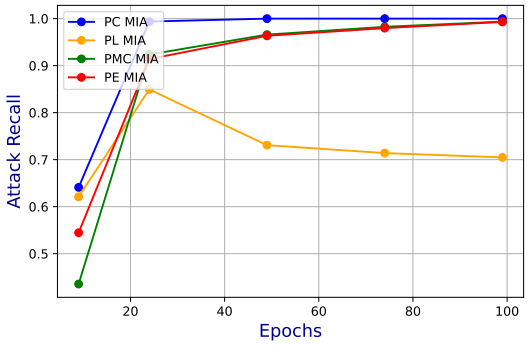
(a) AP with n = 2500



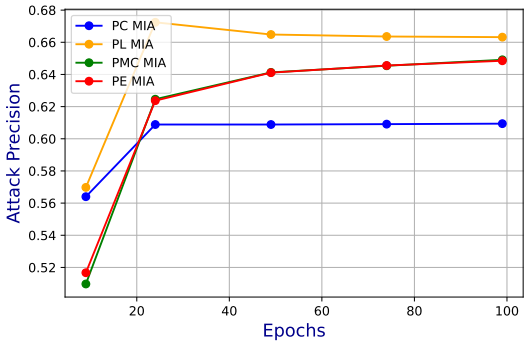
(b) AR with n = 2500



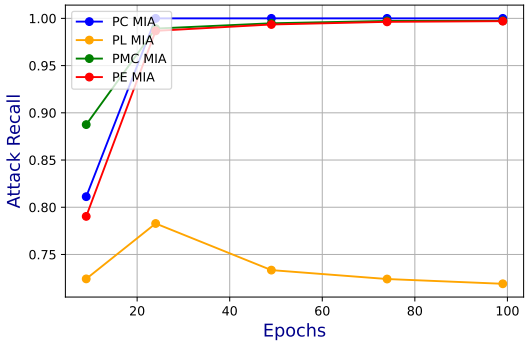
(c) AP with n = 5000



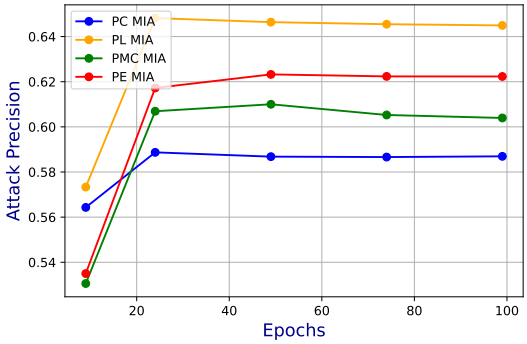
(d) AR with n = 5000



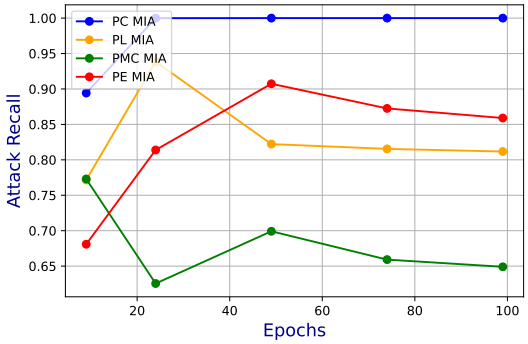
(e) AP with n = 12500



(f) AR with n = 12500

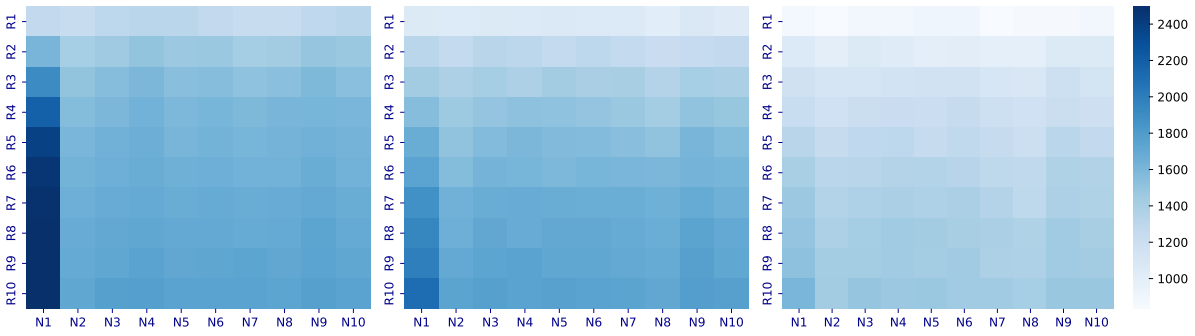


(g) AP with n = 25000

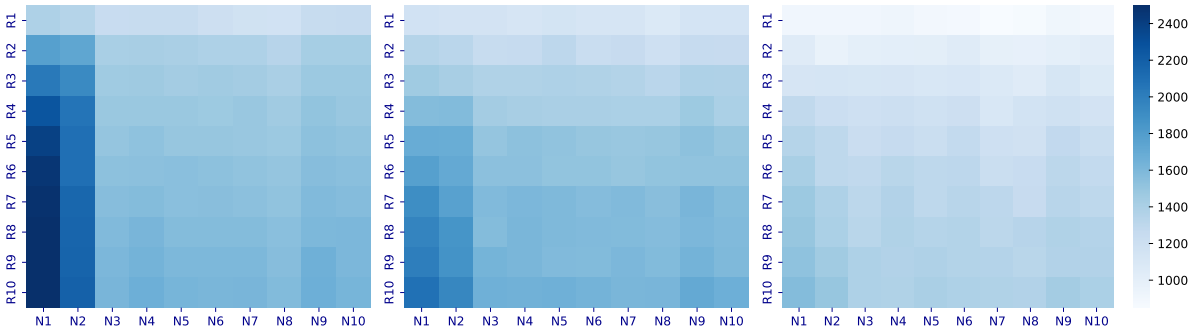


(h) AR with n = 25000

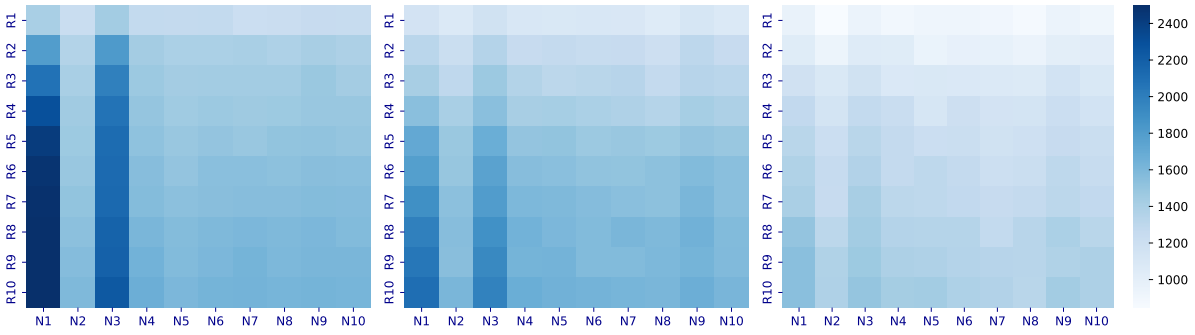
Figure 6.7: Performance of PC, PL, PMC, and PE MIA for the CIFAR-10no Dataset.



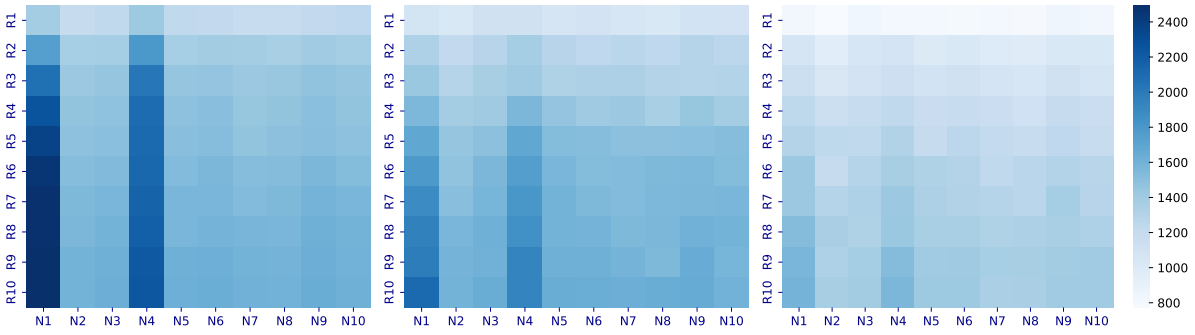
(a) Node 1 Performances



(b) Node 2 Performances

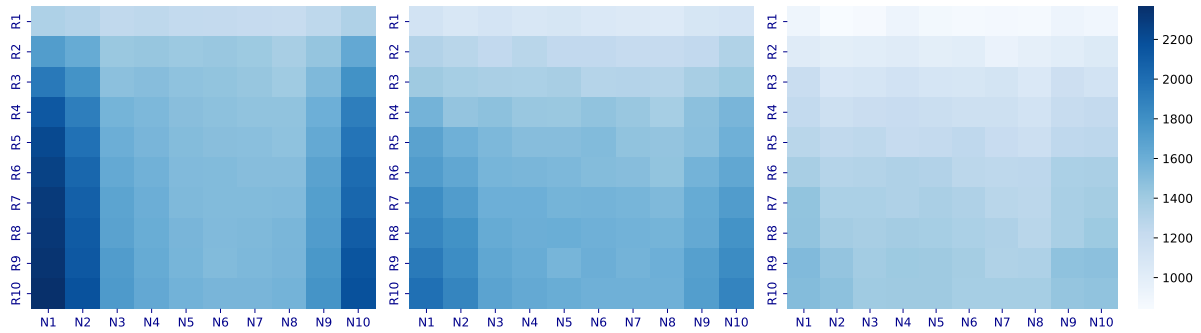


(c) Node 3 Performances

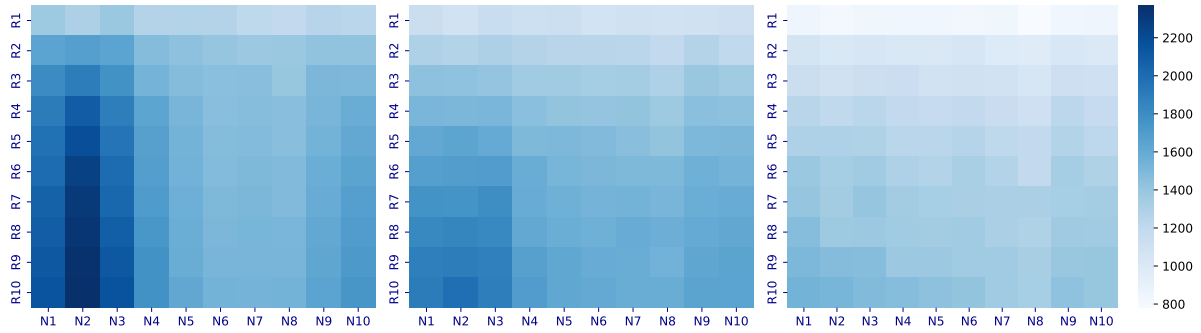


(d) Node 4 Performances

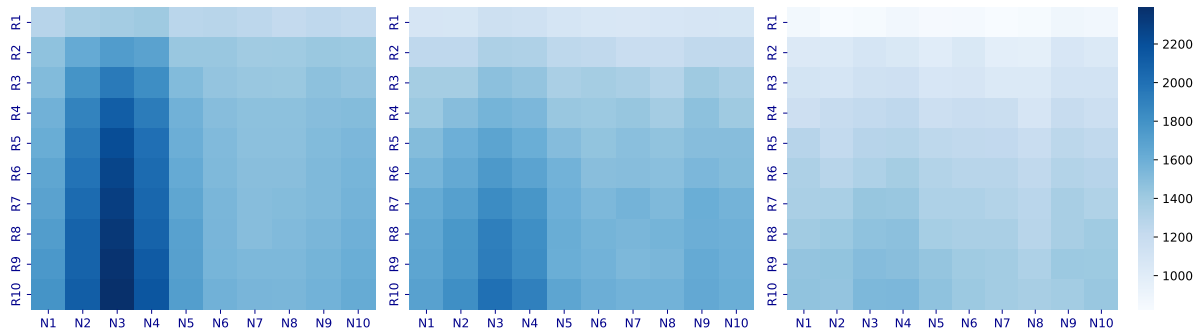
Figure 6.11: Heatmap of AR of PC MIA from Different Nodes in Star network across CIFAR-10no, CIFAR-10aug, and CIFAR-10extend Dataset.



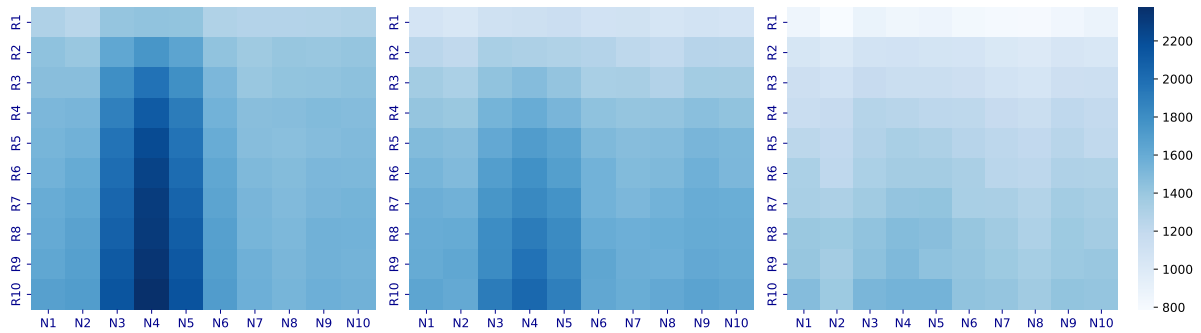
(a) Node 1 Performances



(b) Node 2 Performances



(c) Node 3 Performances



(d) Node 4 Performances

Figure 6.12: Heatmap of AR of PC MIA from Different Nodes in Ring network across CIFAR-10no, CIFAR-10aug, and CIFAR-10extend Dataset.

Chapter 7

Summary and Conclusions

This work evaluates in detail the attack performance of different types of MIA in machine learning and federated learning environments, especially decentralized federated learning. On the one hand, it makes up for the analysis of MIA effects in the context of federated learning; on the other hand, it provides a unique insight into the unique topology of DFL, linking the inference results of MIA with the structure of network topology.

For MIA in machine learning, this work demonstrates that as the model training rounds increase and the training data set is minimal, the attack performance of most MIA will improve, because this means that the overfitting level of the target model is increasing. In addition to evaluating the original standard data set, the work also considers that the data-enhanced preprocessing of the data often reduces the effect of MIA, thus acting as a privacy breach defense effect.

In Federation Learning, it was found that most MIA attacks suffer a significant decrease in effectiveness in FL. This phenomenon is attributed to two factors: (1) Federated learning reduces the overfitting level of the overall model through the continuous aggregation of models among multiple nodes, thus making MIA unable to accurately judge the differences in and out samples. (2) On the other hand, the setting of the federation learning environment interferes with the attacker's judgment of the MIA hypothesis and thus fails to make corresponding simulations, such as training a shadow model with similar effects, which makes the final attack effect very weak. In addition, the ability of different nodes in the federation to reason about the training data of other points in the entire network is also explored. This is often related to the topology of the entire model. In symmetric graphs such as fully connected and circular graphs, the reasoning ability of each node is basically the same, while in star graphs, the central node has significantly stronger reasoning ability than other nodes attached to it. This also provides a possible realistic basis for topological inference attacks that may exist in federated learning.

At the same time, there are some shortcomings in this work, such as fewer studies and the latest estimates of MIA specifically for federated learning. These emerging MIA can theoretically overcome many difficulties in attacking in federation learning and thus achieve a better attack level. In future work, further tests aiming on the attack performance of these MIA could be implemented in DFL and explore whether they still reflect a strong topological logic tendency.

Bibliography

[1] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey", *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[2] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, *et al.*, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges", *IEEE Communications Surveys & Tutorials*, 2023.

[3] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions", *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[4] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: Concepts, challenges, and future directions", *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3501–3509, 2021.

[5] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks", *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 1, pp. 1–30, 2018.

[6] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases", in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 644–655.

[7] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures", in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[8] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference", in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 148–162.

[9] S. Dayal, D. Alhadidi, A. Abbasi Tadi, and N. Mohammed, "Comparative analysis of membership inference attacks in federated learning", in *Proceedings of the 27th International Database Engineered Applications Symposium*, 2023, pp. 185–192.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models", in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18. DOI: 10.1109/SP.2017.41.

[11] 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", *arXiv preprint arXiv:1806.01246*, 2018.

[12] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning", in *2019 IEEE symposium on security and privacy (SP)*, IEEE, 2019, pp. 739–753.

[13] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting", in *2018 IEEE 31st computer security foundations symposium (CSF)*, IEEE, 2018, pp. 268–282.

[14] L. Liu, Y. Wang, G. Liu, K. Peng, and C. Wang, "Membership inference attacks against machine learning models via prediction sensitivity", *IEEE Transactions on Dependable and Secure Computing*, 2022.

[15] P. Irolla and G. Châtel, "Demystifying the membership inference attack", in *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*, IEEE, 2019, pp. 1–7.

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

[17] T. Nguyen, P. Lai, K. Tran, N. Phan, and M. T. Thai, "Active membership inference attack under local differential privacy in federated learning", *arXiv preprint arXiv:2302.12685*, 2023.

[18] S. Banerjee, S. Roy, S. F. Ahamed, *et al.*, "Mia-bad: An approach for enhancing membership inference attack and its mitigation with federated learning", *arXiv preprint arXiv:2312.00051*, 2023.

[19] Y. Gu, Y. Bai, and S. Xu, "Cs-mia: Membership inference attack based on prediction confidence series in federated learning", *Journal of Information Security and Applications*, vol. 67, p. 103 201, 2022.

[20] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning", in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1102–1107. DOI: 10.1109/ICDM51629.2021.00129.

[21] C. Dwork, "Differential privacy", in *International colloquium on automata, languages, and programming*, Springer, 2006, pp. 1–12.

[22] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey", *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[23] R. J. Little, "Modeling the drop-out mechanism in repeated-measures studies", *Journal of the american statistical association*, vol. 90, no. 431, pp. 1112–1121, 1995.

[24] E. T. M. Beltrán, Á. L. P. Gómez, C. Feng, *et al.*, "Fedstellar: A platform for decentralized federated learning", *Expert Systems with Applications*, vol. 242, p. 122 861, 2024.

[25] Y. LeCun and C. Cortes, *MNIST handwritten digit database*, http://yann.lecun.com/exdb/mnist/, Accessed: 2016-01-14, 2010.

[26] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms", *arXiv:1708.07747 [cs, stat]*, Sep. 2017, Accessed: 2023-07-11. DOI: 10.48550/arXiv.1708.07747.

[27] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images", 2009.

[28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

Abbreviations

ML	Machine Learning
FL	Federated Learning
CFL	Centralized Federated Learning
DFL	Decentralized Federated Learning
IID	Independent and Identically Distributed
MIA	Membership Inference Attack
AP	Attack Precision
AR	Attack Recall
DP	Differential Privacy

List of Figures

List of Tables

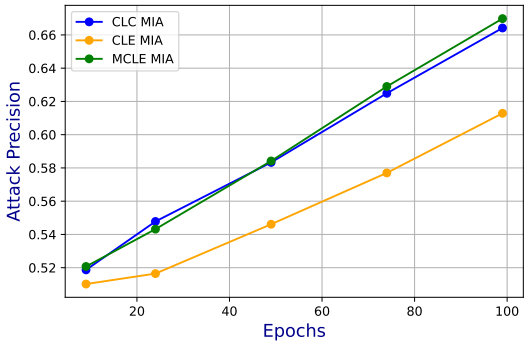
Appendix

Dataset	Size of Training Data	Epochs	AP	AR	F1-Score	Overfitting Level	AUC
CIFAR10	n= 2500	10	0.513	0.630	0.565	-0.04%	0.513
		25	0.523	0.354	0.422	5.20%	0.520
		50	0.531	0.663	0.590	14.30%	0.550
		75	0.548	0.767	0.639	23.96%	0.598
		100	0.597	0.836	0.697	33.18%	0.679
	n=5000	10	0.511	0.374	0.432	-1.68%	0.514
		25	0.513	0.629	0.565	1.28%	0.525
		50	0.516	0.741	0.609	10.35%	0.531
		75	0.553	0.690	0.614	16.72%	0.584
		100	0.583	0.696	0.635	24.05%	0.634
	n = 12500	10	0.504	0.659	0.571	-0.80%	0.507
		25	0.509	0.737	0.602	3.24%	0.519
		50	0.531	0.770	0.629	10.41%	0.556
		75	0.552	0.805	0.655	15.93%	0.598
		100	0.561	0.842	0.673	19.69%	0.620
	n = 25000	10	0.497	0.496	0.496	-1.67%	0.495
		25	0.494	0.453	0.473	3.10%	0.495
		50	0.515	0.853	0.642	8.36%	0.527
		75	0.532	0.821	0.646	11.37%	0.563
		100	0.536	0.850	0.658	13.22%	0.574
CIFAR10 extend	n= 2500	10	0.493	0.298	0.371	-1.58%	0.501
		25	0.503	0.412	0.453	3.28%	0.498
		50	0.503	0.412	0.453	3.28%	0.510
		75	0.517	0.646	0.574	8.56%	0.523
		100	0.520	0.580	0.548	9.40%	0.532
	n=5000	10	0.508	0.715	0.594	-1.90%	0.515
		25	0.515	0.329	0.401	-2.44%	0.509
		50	0.510	0.455	0.481	-1.38%	0.510
		75	0.519	0.551	0.534	2.12%	0.525
		100	0.528	0.461	0.492	5.34%	0.534
	n = 12500	10	0.505	0.588	0.543	-4.54%	0.509
		25	0.507	0.303	0.380	-2.56%	0.505
		50	0.510	0.559	0.533	-0.56%	0.509
		75	0.508	0.686	0.584	2.05%	0.516
		100	0.518	0.698	0.595	1.62%	0.525
	n = 25000	10	0.497	0.460	0.478	-2.38%	0.494
		25	0.491	0.350	0.409	-3.32%	0.491
		50	0.484	0.293	0.365	-1.75%	0.488
		75	0.480	0.332	0.393	-0.32%	0.483
		100	0.487	0.356	0.411	0.35%	0.490

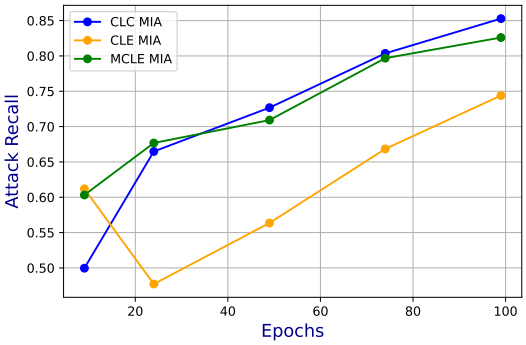
Table 7.1: Performance of SM MIA for CIFAR-10 Dataset with the First and the Second Data Augmentation based on Adversary 2 Assumption.

Dataset	Size of Training Data	Epochs	AP	AR	F1-Score	Overfitting Level	AUC
CIFAR10	n= 2500	10	0.520	0.390	0.445	-0.04%	0.513
		25	0.507	0.100	0.168	5.20%	0.521
		50	0.584	0.128	0.209	14.30%	0.565
		75	0.611	0.281	0.385	23.96%	0.618
		100	0.651	0.460	0.539	33.18%	0.674
	n=5000	10	0.509	0.316	0.390	-1.68%	0.508
		25	0.520	0.159	0.243	1.28%	0.519
		50	0.543	0.115	0.190	10.35%	0.539
		75	0.581	0.260	0.360	16.72%	0.574
		100	0.609	0.415	0.493	24.05%	0.634
	n = 12500	10	0.517	0.295	0.376	-0.80%	0.511
		25	0.529	0.136	0.216	3.24%	0.521
		50	0.539	0.253	0.344	10.41%	0.549
		75	0.559	0.341	0.424	15.93%	0.582
		100	0.579	0.462	0.514	19.69%	0.608
	n = 25000	10	0.505	0.429	0.464	-1.67%	0.505
		25	0.515	0.328	0.400	3.10%	0.515
		50	0.524	0.200	0.290	8.36%	0.530
		75	0.532	0.276	0.363	11.37%	0.544
		100	0.543	0.383	0.449	13.22%	0.559
CIFAR10 extend	n= 2500	10	0.514	0.082	0.141	-1.58%	0.504
		25	0.481	0.016	0.030	3.28%	0.513
		50	0.500	0.001	0.002	3.28%	0.507
		75	0.468	0.020	0.039	8.56%	0.522
		100	0.565	0.042	0.078	9.40%	0.539
	n=5000	10	0.521	0.181	0.268	-1.90%	0.504
		25	0.546	0.029	0.054	-2.44%	0.511
		50	0.557	0.011	0.021	-1.38%	0.509
		75	0.536	0.029	0.054	2.12%	0.513
		100	0.559	0.039	0.072	5.34%	0.527
	n = 12500	10	0.504	0.128	0.204	-4.54%	0.504
		25	0.527	0.008	0.016	-2.56%	0.509
		50	0.516	0.024	0.047	-0.56%	0.509
		75	0.525	0.048	0.088	2.05%	0.508
		100	0.539	0.081	0.140	1.62%	0.516
	n = 25000	10	0.506	0.249	0.334	-2.38%	0.500
		25	0.506	0.114	0.187	-3.32%	0.505
		50	0.520	0.043	0.080	-1.75%	0.506
		75	0.524	0.061	0.109	-0.32%	0.508
		100	0.521	0.081	0.140	0.35%	0.511

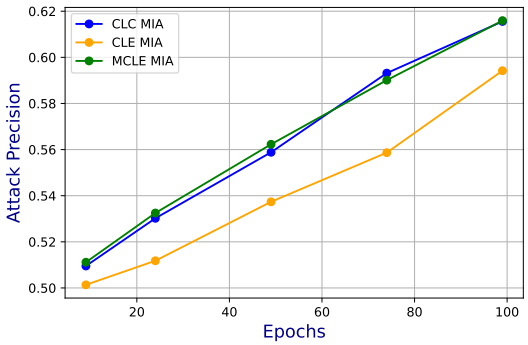
Table 7.2: Performance of SM MIA for CIFAR-10 Dataset with the First and the Second Data Augmentation based on Adversary 3 Assumption.



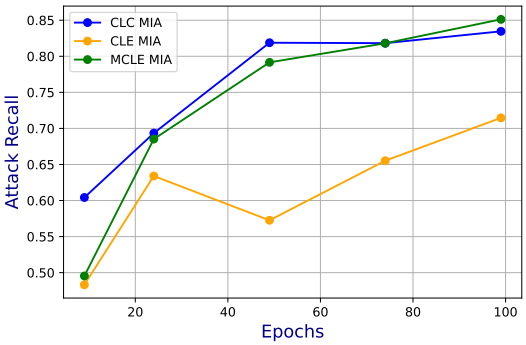
(a) AP with n = 2500



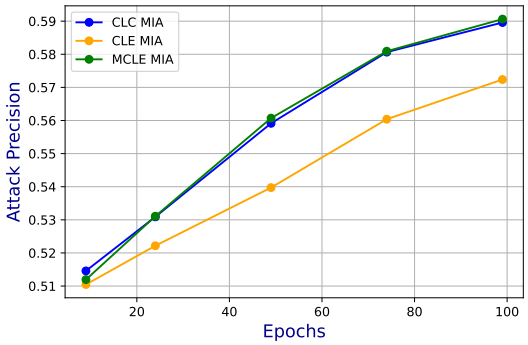
(b) AR with n = 2500



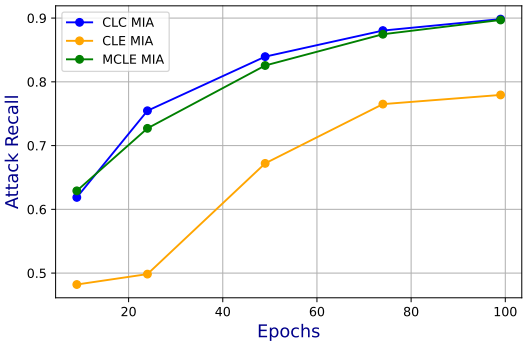
(c) AP with n = 5000



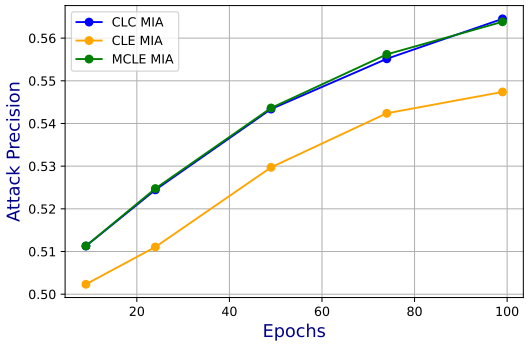
(d) AR with n = 5000



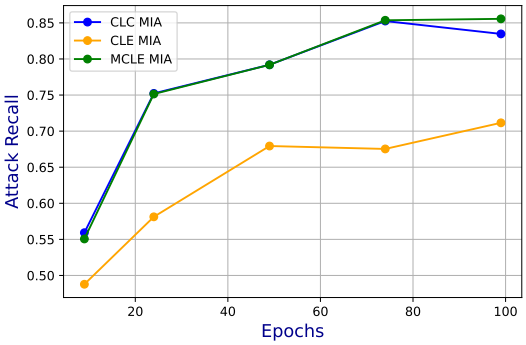
(e) AP with n = 12500



(f) AR with n = 12500

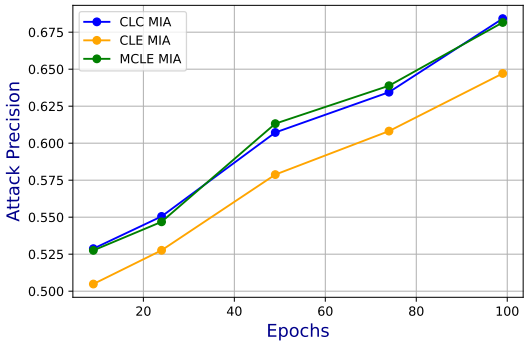


(g) AP with n = 25000

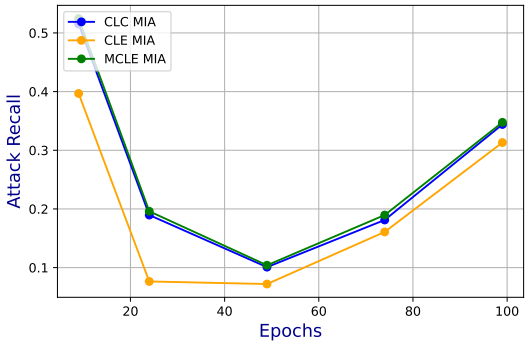


(h) AR with n = 25000

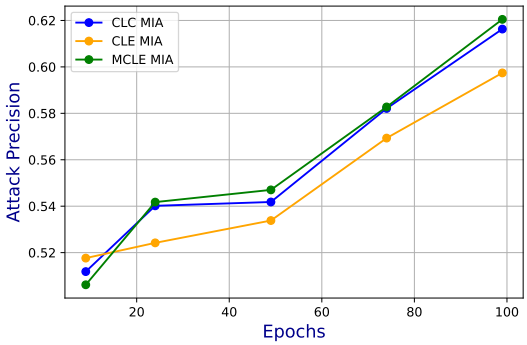
Figure 7.1: Performance of CLA, CLE, and MCLE MIA for the CIFAR-10aug Dataset based on Adversary 2 Assumption.



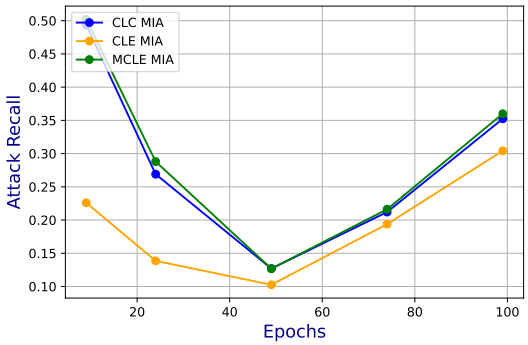
(a) AP with n = 2500



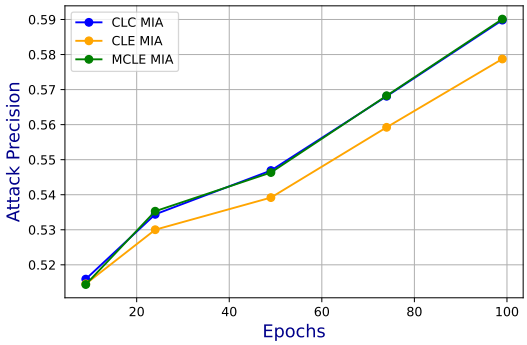
(b) AR with n = 2500



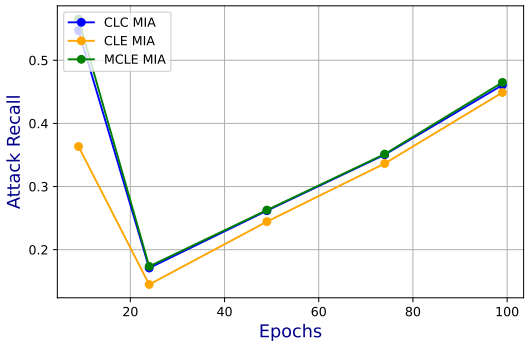
(c) AP with n = 5000



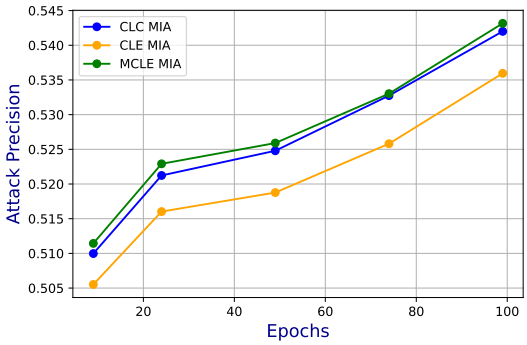
(d) AR with n = 5000



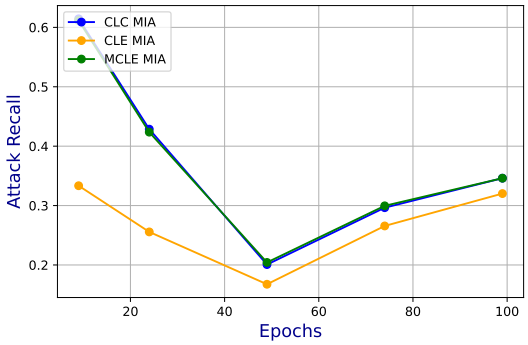
(e) AP with n = 12500



(f) AR with n = 12500

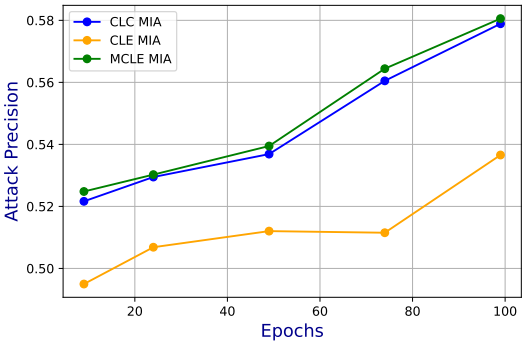


(g) AP with n = 25000

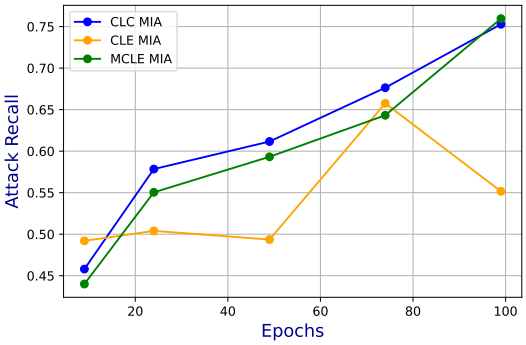


(h) AR with n = 25000

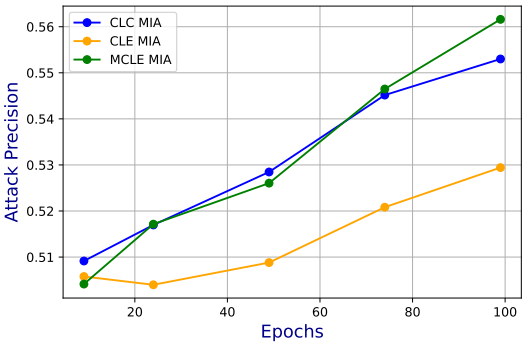
Figure 7.2: Performance of CLA, CLE and MCLE MIA for CIFAR-10aug Dataset based on Adversary 3 Assumption.



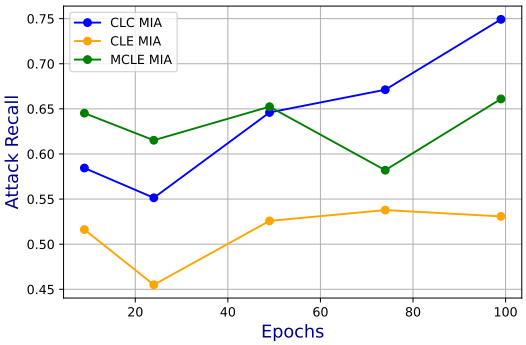
(a) AP with n = 2500



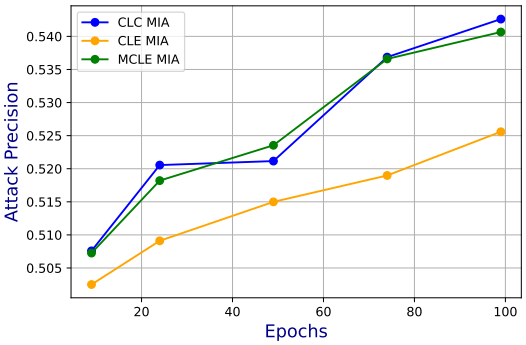
(b) AR with n = 2500



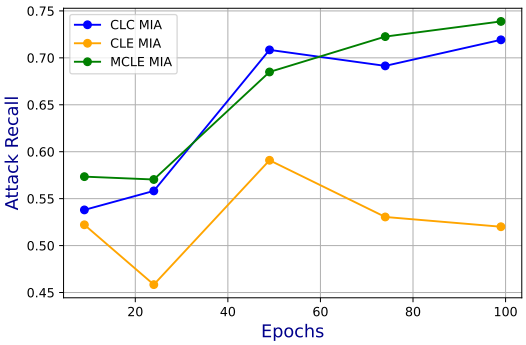
(c) AP with n = 5000



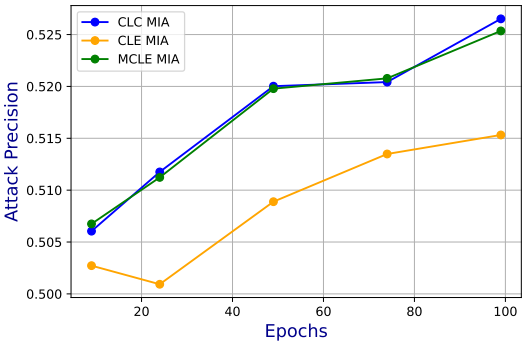
(d) AR with n = 5000



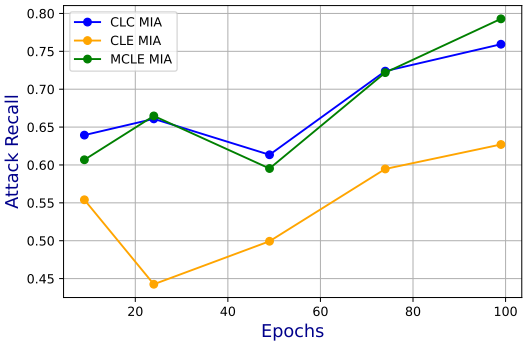
(e) AP with n = 12500



(f) AR with n = 12500

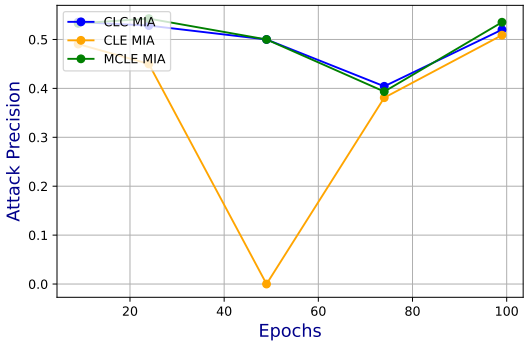


(g) AP with n = 25000

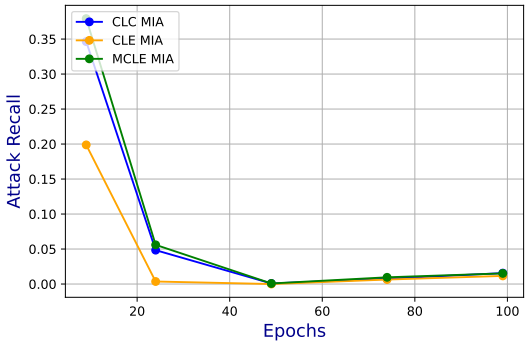


(h) AR with n = 25000

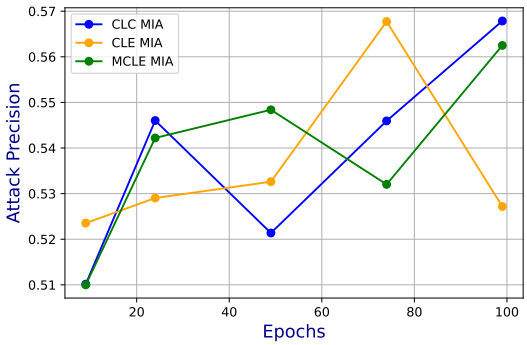
Figure 7.3: Performance of CLA, CLE and MCLE MIA for CIFAR-10extend Dataset based on Adversary 2 Assumption.



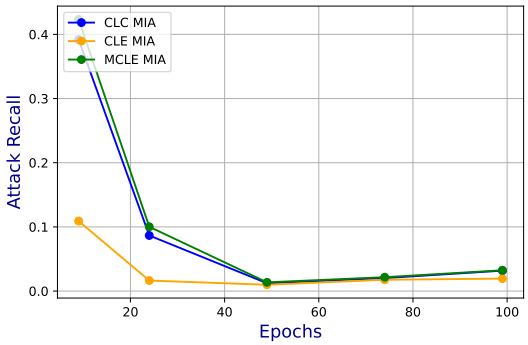
(a) AP with n = 2500



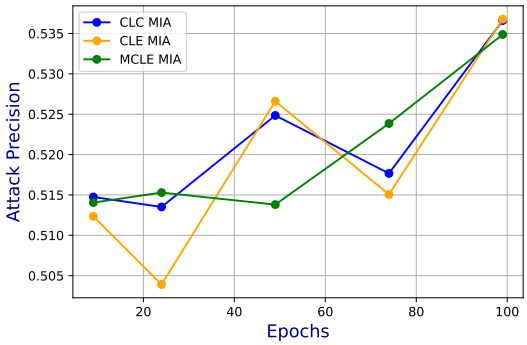
(b) AR with n = 2500



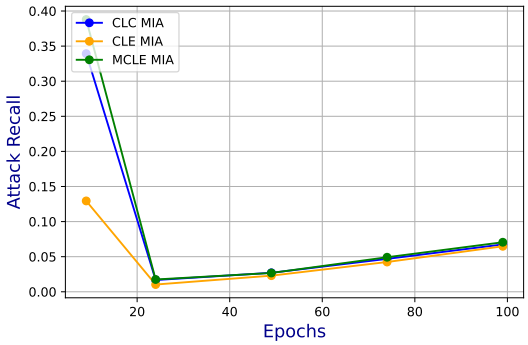
(c) AP with n = 5000



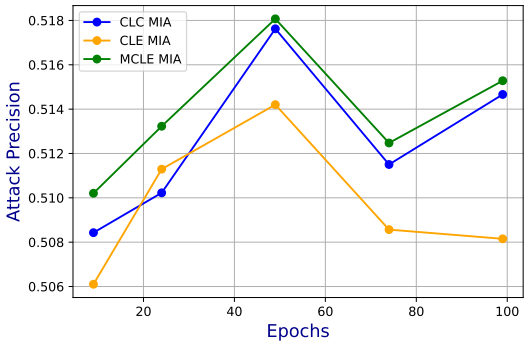
(d) AR with n = 5000



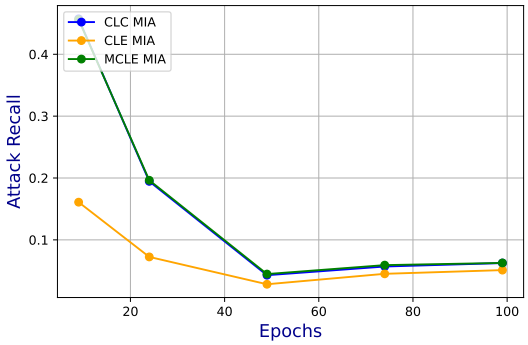
(e) AP with n = 12500



(f) AR with n = 12500

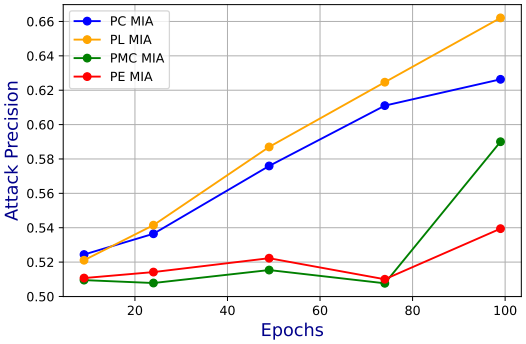


(g) AP with n = 25000

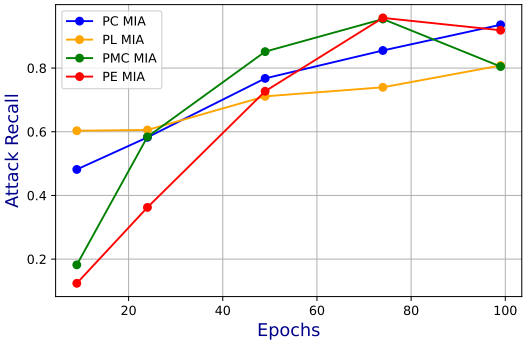


(h) AR with n = 25000

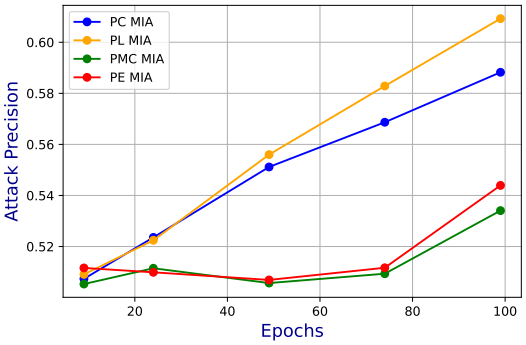
Figure 7.4: Performance of CLA, CLE and MCLE MIA for CIFAR-10extend Dataset based on Adversary 3 Assumption.



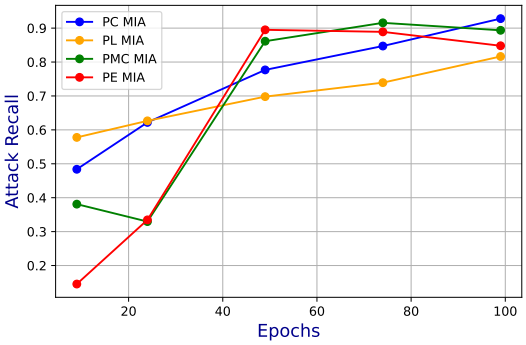
(a) AP with n = 2500



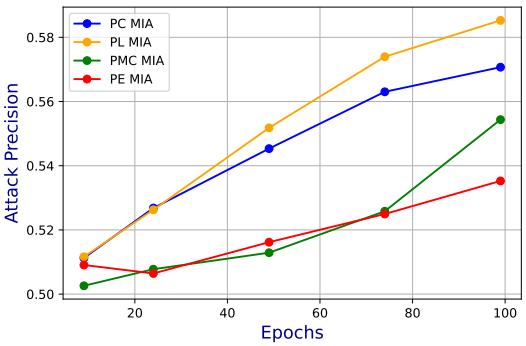
(b) AR with n = 2500



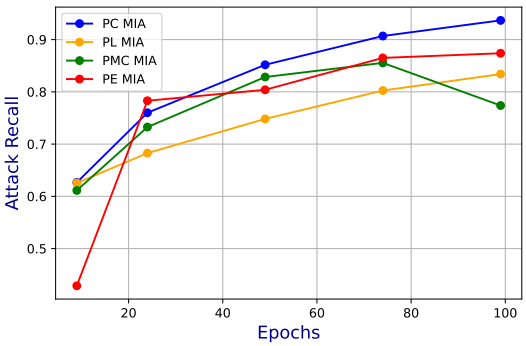
(c) AP with n = 5000



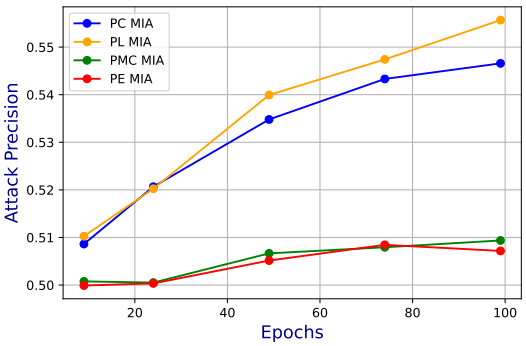
(d) AR with n = 5000



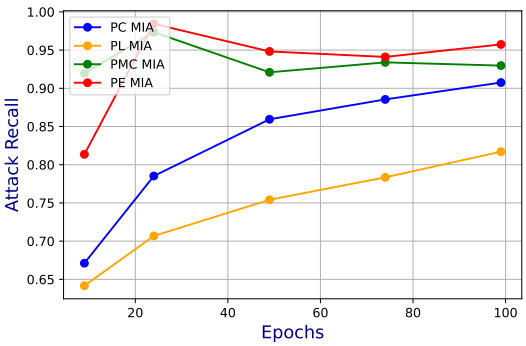
(e) AP with n = 12500



(f) AR with n = 12500

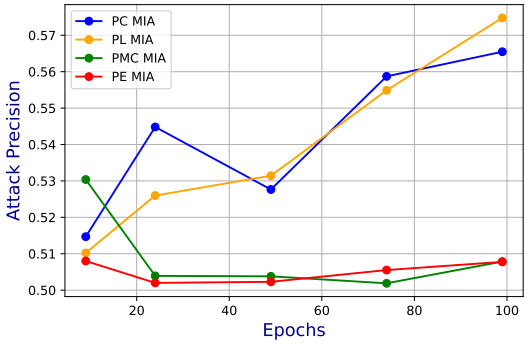


(g) AP with n = 25000

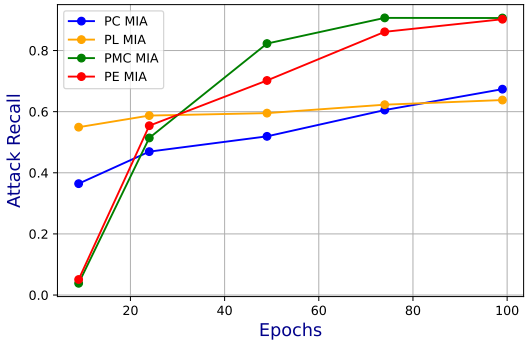


(h) AR with n = 25000

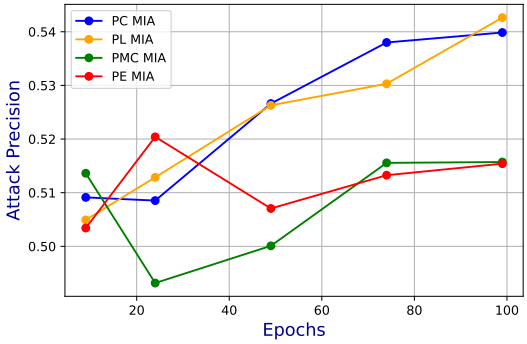
Figure 7.5: Performance of PC, PL, PMC, and PE MIA for the CIFAR-10aug Dataset.



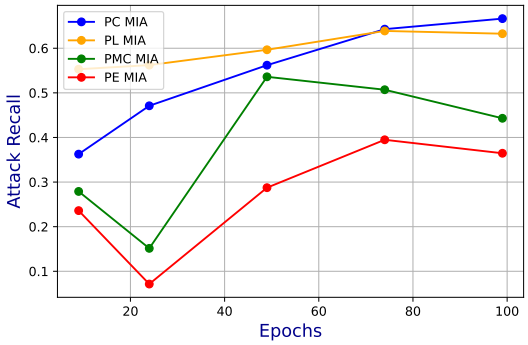
(a) AP with n = 2500



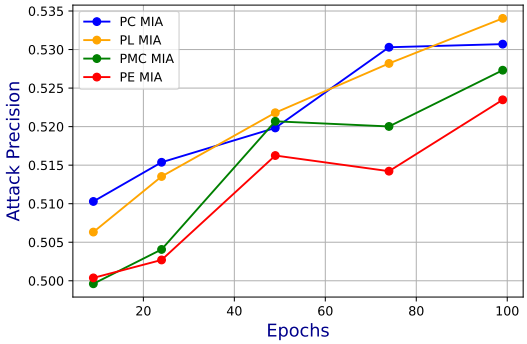
(b) AR with n = 2500



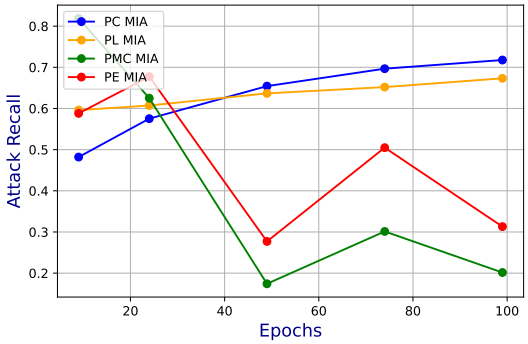
(c) AP with n = 5000



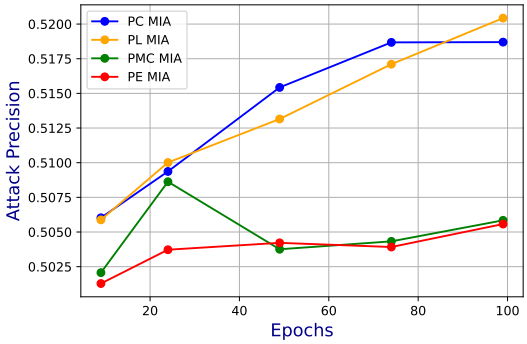
(d) AR with n = 5000



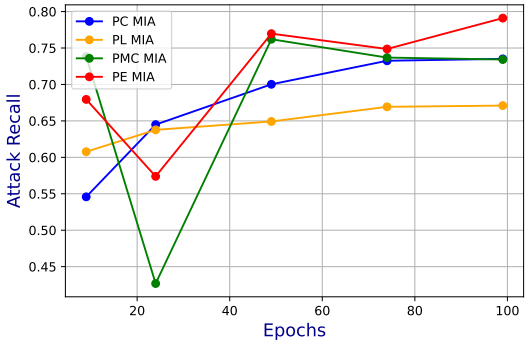
(e) AP with n = 12500



(f) AR with n = 12500

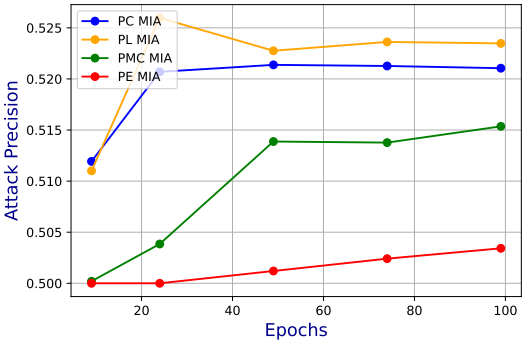


(g) AP with n = 25000

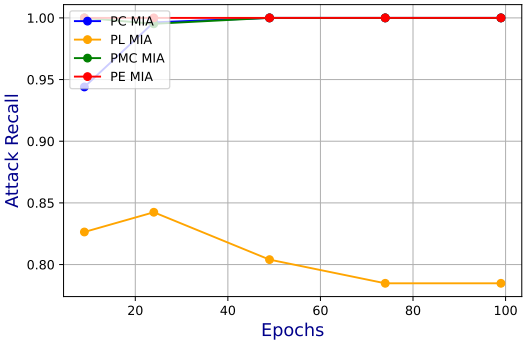


(h) AR with n = 25000

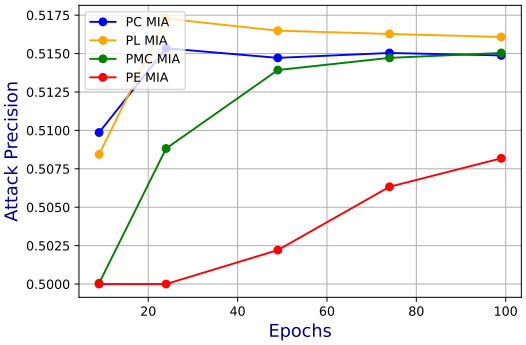
Figure 7.6: Performance of PC, PL, PMC, and PE MIA for the CIFAR-10extend Dataset.



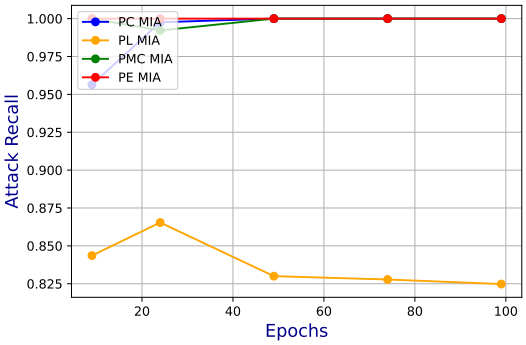
(a) AP with n = 2500



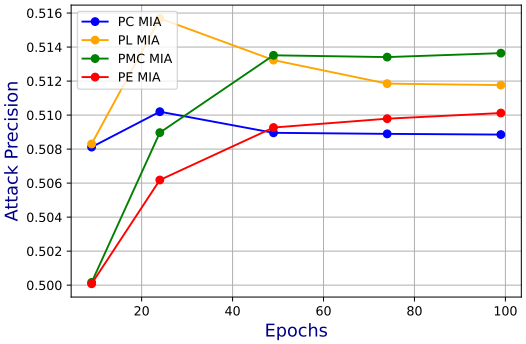
(b) AR with n = 2500



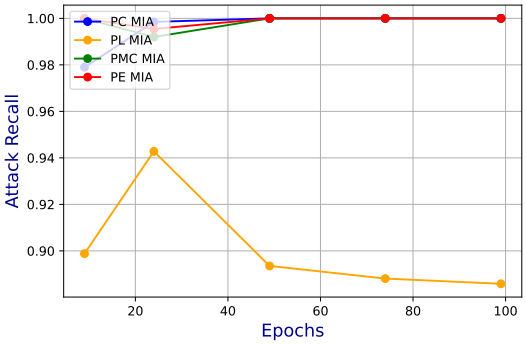
(c) AP with n = 5000



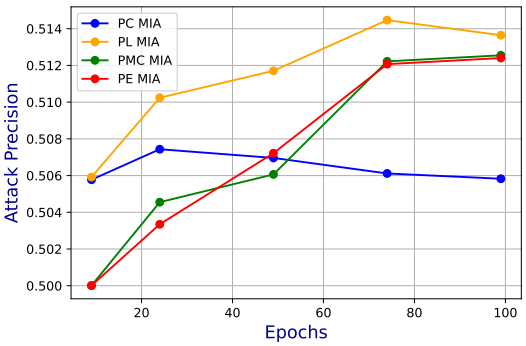
(d) AR with n = 5000



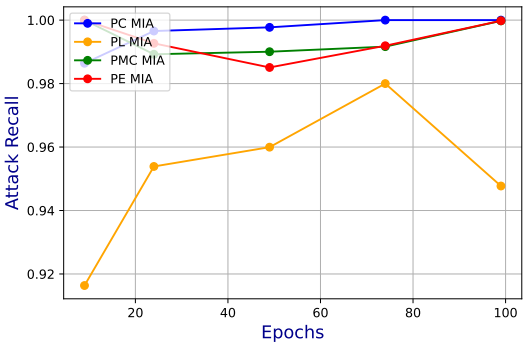
(e) AP with n = 12500



(f) AR with n = 12500

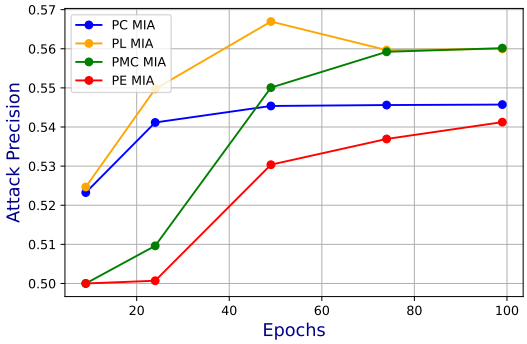


(g) AP with n = 25000

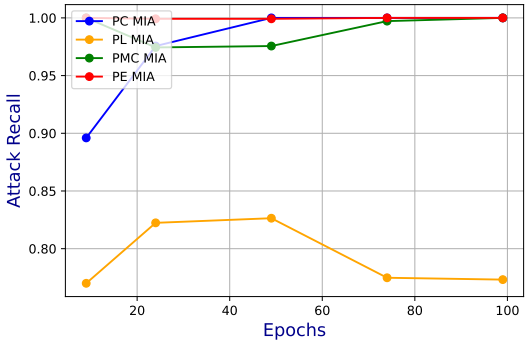


(h) AR with n = 25000

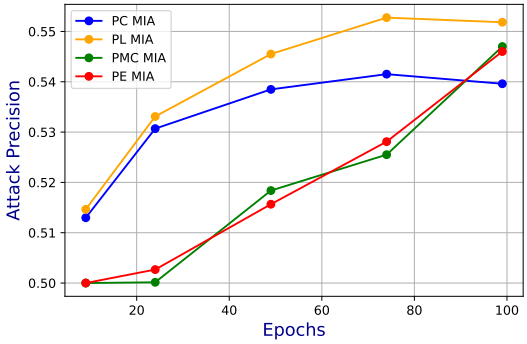
Figure 7.7: Performance of PC, PL, PMC, and PE MIA for the Mnist Dataset.



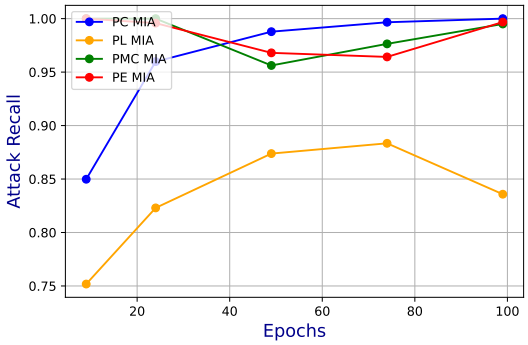
(a) AP with n = 2500



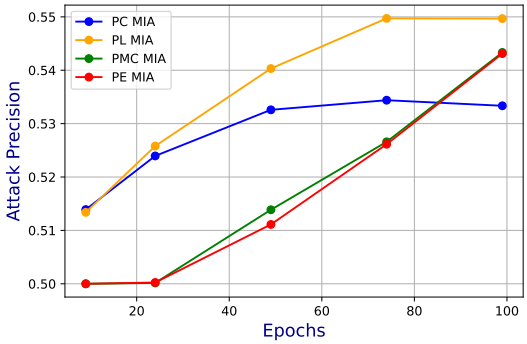
(b) AR with n = 2500



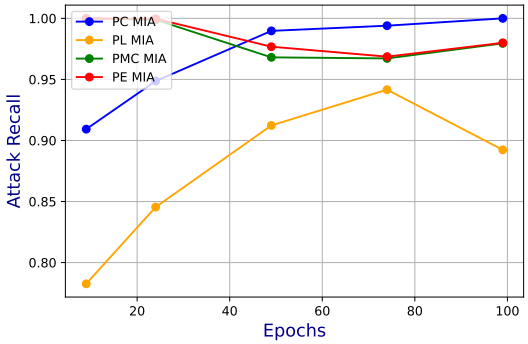
(c) AP with n = 5000



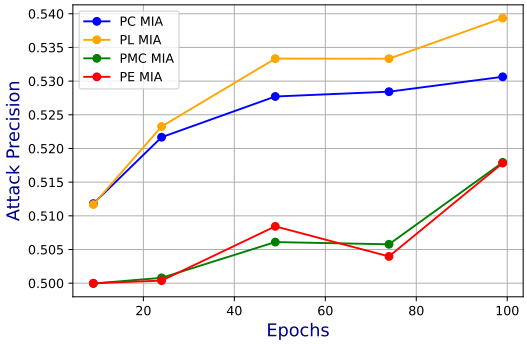
(d) AR with n = 5000



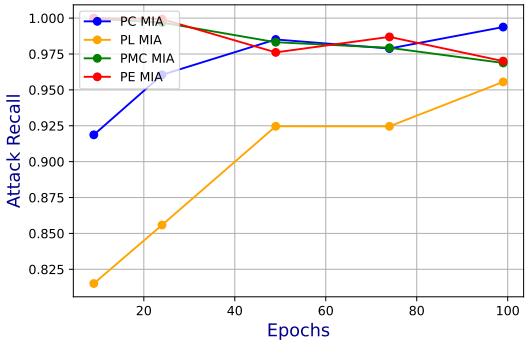
(e) AP with n = 12500



(f) AR with n = 12500



(g) AP with n = 25000



(h) AR with n = 25000

Figure 7.8: Performance of PC, PL, PMC, and PE MIA for the FMnist Dataset.