information

Article

Comparative Analysis of Membership Inference Attacks in Federated and Centralized Learning †

Ali Abbasi Tadi [1,*], Saroj Dayal [1], Dima Alhadidi [1] and Noman Mohammed [2]

[1] School of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada; sdayal@uwindsor.ca (S.D.); dima.alhadidi@uwindsor.ca (D.A.)
[2] Department of Computer Science, University of Manitoba, Winnipeg, MB R3T 2N2, Canada; noman.mohammed@umanitoba.ca
* Correspondence: abbasit@uwindsor.ca
† This paper is an extended version of our paper published in International Database Engineered Applications Symposium Conference, Heraklion, Crete, Greece, 5–7 May 2023. Entitled 'Comparative Analysis of Membership Inference Attacks in Federated Learning'.

Abstract: The vulnerability of machine learning models to membership inference attacks, which aim to determine whether a specific record belongs to the training dataset, is explored in this paper. Federated learning allows multiple parties to independently train a model without sharing or centralizing their data, offering privacy advantages. However, when private datasets are used in federated learning and model access is granted, the risk of membership inference attacks emerges, potentially compromising sensitive data. To address this, effective defenses in a federated learning environment must be developed without compromising the utility of the target model. This study empirically investigates and compares membership inference attack methodologies in both federated and centralized learning environments, utilizing diverse optimizers and assessing attacks with and without defenses on image and tabular datasets. The findings demonstrate that a combination of knowledge distillation and conventional mitigation techniques (such as Gaussian dropout, Gaussian noise, and activity regularization) significantly mitigates the risk of information leakage in both federated and centralized settings.

Keywords: federated learning; membership inference attack; privacy; machine learning

Citation: Abbasi Tadi, A.; Dayal, S.; Alhadidi, D.; Mohammed, N. Comparative Analysis of Membership Inference Attacks in Federated and Centralized Learning. *Information* **2023**, *14*, 620. https://doi.org/10.3390/info14110620

Academic Editor: Peter Revesz

Received: 30 September 2023
Revised: 17 November 2023
Accepted: 18 November 2023
Published: 19 November 2023

1. Introduction

Machine learning (ML) is gaining popularity thanks to the increasing availability of extensive datasets and technological advancements [1,2]. Centralized learning (CL) techniques become impractical in the context of abundant private data as they mandate transmitting and processing data through a central server. Google's federated learning (FL) has emerged as a distributed machine learning paradigm since its inception in 2017 [3]. In FL, a central server supports participants in the training model by exchanging trained models or gradients of training data without revealing raw or sensitive information either to the central server or other participants. The application of FL is crucial, particularly in processing sensitive and personal data, such as in healthcare, where ML is increasingly prevalent, especially in compliance with GDPR [4] and HIPAA [5] regulations. Despite its advancements, FL is susceptible to membership inference attacks (MIA), a method employed to gain insights into training data. Although FL primarily aims for privacy protection, attackers can infer specific data by intercepting FL updates transmitted between training parties and the central server [6,7]. For instance, if an attacker is aware that patient data are part of the model's training set, they could deduce the patient's current health status [8]. Prior research has explored membership inference attacks (MIA) in a centralized environment where data are owned by a single data owner. It is imperative to extend this investigation to MIA in federated learning (FL). This article undertakes an analysis

of various MIA techniques initially proposed in the centralized learning (CL) environment [9–11]. The examination encompasses their applicability in the FL environment and evaluates the effectiveness of countermeasures to mitigate these attacks in both FL and CL environments. An earlier version of this work has already been published [12], focusing solely on MIA in the FL environment. In that iteration, we scrutinized nine mitigation techniques [9,10,13–19] against MIA attacks and showed that knowledge distillation [19] performs better in reducing the attack recall while keeping accuracy as high as possible. We also conducted some experiments to observe the effects of three various optimizers, Stochastic Gradient Descent (SGD) [20], Root Mean Squared Propagation (RMSProp) [21], and Adaptive Gradient (Adagrad) [22], in deep learning on MIA recall and FL model accuracy. We found no difference between these optimizers on MIA recall. In this paper, we investigated two more optimizers and three more countermeasures in both CL and FL environments, and we compared the results. To the best of our knowledge, this study is the first comprehensive study that investigates the MIA in both CL and FL environments and applies twelve mitigation techniques against MIA with five various optimizers for the target model. Our contributions in this paper are summarized below.

- We conducted a comprehensive study of the effectiveness of the membership inference attack in the FL and CL environments considering different attack techniques, optimizers, datasets, and countermeasures. Existing related work focuses on the CL environment and the effectiveness of one single countermeasure. In this paper, we investigated the FL environment, compared it with the CL environment, and studied the effectiveness of combining two mitigation techniques together.
- We compared the effectiveness of four well-known membership inference attacks [9–11] in the CL and FL environments considering different mitigation techniques: dropout [16], Monte Carlo dropout [13], batch normalization [14], Gaussian noise [23], Gaussian dropout [16], activity regularization [24], masking [17], and knowledge distillation [19].
- We compared the accuracy of models in the CL and the FL environments using five optimizers: SGD, RMSProp, Adagrad, incorporation of Nesterov momentum into Adam (Nadam) [25], and Adaptive Learning Rate method (Adadelta) [26] using four real datasets, MNIST [27], Fashion-MNIST (FMNIST) [28], CIFAR-10 [29], and Purchase [30]. We found that using the Adadelta optimizer alone, for image datasets, can mitigate the MIA significantly while preserving the accuracy of the model.
- We established a trade-off relationship between model accuracy and attack recall. Our investigation revealed that employing knowledge distillation in conjunction with either Gaussian noise, Gaussian dropout, or activity regularization yields the most favorable balance between model accuracy and attack recall across both image and tabular datasets.

The remainder of this article is organized as follows. In the Section 2, we presented the related work. In Section 3, we explained the different attacks on a model for membership inference. Countermeasures are detailed in Section 4. The setup and the results of the experiments are described and analyzed in Section 5. Finally, we conclude our article in Section 6.

2. Related Work

This section summarizes the related work focusing on the MIA in CL and FL (Table 1).

Table 1. Related work summary.

Authors	CL or FL	Attack	Defense
Shokri et al. [9]	CL	✓	✓
Salem et al. [10]	CL	✓	✓
Nasr et al. [31]	CL, FL	✓	×
Liu et al. [11]	CL	✓	×
Carlini et al. [2]	CL	✓	×
Conti et al. [32]	CL	✓	✓
Zheng et al. [33]	CL	×	✓
Shejwalkar et al. [34]	CL	×	✓
Lee et al. [35]	FL	×	✓
Su et al. [36]	FL	×	✓
Xie et al. [37]	FL	×	✓

2.1. MIA against CL

Shokri et al. [9] performed the first MIA on ML models to identify the presence of a data sample in the training set of the ML model with black-box access. Shokri et al. [9] created a target model, shadow models, and attack models, and they made two main assumptions. First, the attacker must create multiple shadow models, each with the same structure as the target model. Second, the dataset used to train shadow models comes from the same distribution as the target model's training data. Subsequently, Salem et al. [10] widened the scope of the MIA of Shokri et al. [9]. They showed that the MIA is possible without having any prior assumption of the target model dataset or having multiple shadow models. Nasr et al. [31] showed that more reasonable attack scenarios are possible in both FL and CL environments. They designed a white-box attack on the target model in FL and CL by assuming different adversary prior knowledge. Lan Liu et al. [11] studied perturbations in feature space and found that the sensitivity of trained data to a fully trained machine learning model is lower than that of untrained data. Lan Liu et al. [11] calculated sensitivity by comparing the sensitivity values of different data samples using a Jacobian matrix, which measures the relationship between the target's predictions and the feature value of the target sample.

Numerous attacks in the existing literature draw inspiration from Shokri's research [9]. Carlini et al. [2] introduced a novel attack called the Likelihood Ratio Attack (LiRA), which amalgamates concepts from various research papers. They advocate for a shift in the evaluation metric for MIA by recommending the use of the true positive rate (recall) while maintaining a very low false alarm rate. Their findings reveal that, when measured by recall, many attacks prove to be less effective than previously believed. In our study, we adopt the use of recall, rather than accuracy, as the measure of MIA attack effectiveness.

2.2. MIA against FL

Nasr et al. [31] showed that MIA seriously compromises the privacy of FL participants even when the universal model achieves high prediction accuracy. A common defense against such attacks is the differential privacy (DP) [38] approach, which manipulates each update with some random noise. However, it suffers from a significant loss of FL classification accuracy. Bai et al. [39] proposed a homomorphic-cryptography-based privacy enhancement mechanism impacting MIA. They used homomorphic cryptography to encrypt the collaborators' parameters and added a parameter selection method to the FL system aggregator to select specific participant updates with a given probability. Another FL MIA defense technique is the digestive neural network (DNN) [35], which modifies inputs and skews updates, maximizing FL classification accuracy and minimizing inference attack accuracy. Wang et al. [36] proposed a new privacy mechanism called the Federated Regularization Learning Model to prevent information leakage in FL. Xie et al. [37] proposed an adversarial noise generation method that was added to the attack features of the attack model on MIA against FL.

3. Attack Techniques for Membership Inference

In this section, we summarize the different methods of MIA [9–11] that we applied in this paper. The summary of the considered membership attacks is shown in Table 2. We employed four well-known attacks in this paper, and each of them has its own characteristics.

Table 2. Comparison of the considered attacks.

Attack Type	Shadow Model		Target's Model		
	No. Shadow Models	Target Model Structure	Training Data Distribution	Prediction Sensitivity	
Attack 1 [9]	10	✓	✓	-	
Attack 2 [10]	1	-	✓	-	
Attack 3 [10]	1	-	-	-	
Attack 4 [11]	-	-	-	✓	

3.1. Shokri et al.'s MIA

MIA can be formulated [40] as follows:

$$M_{Attack}(K_{M_{Target}}(x, y)) \rightarrow 0, 1 \quad (1)$$

Given a data sample(x, y) and additional knowledge $K_{M_{Target}}$ about the target model M_{Target}, the attacker typically tries to create an attack model M_{Attack} to eventually return either 0 or 1, where 0 indicates the sample is not a member of the training set and 1 indicates the sample is a member of the training set. The additional knowledge can be the distribution of the target data and the type of the target model. Figure 1 summarizes the general idea of the first MIA on ML models proposed by Shokri et al. [9].

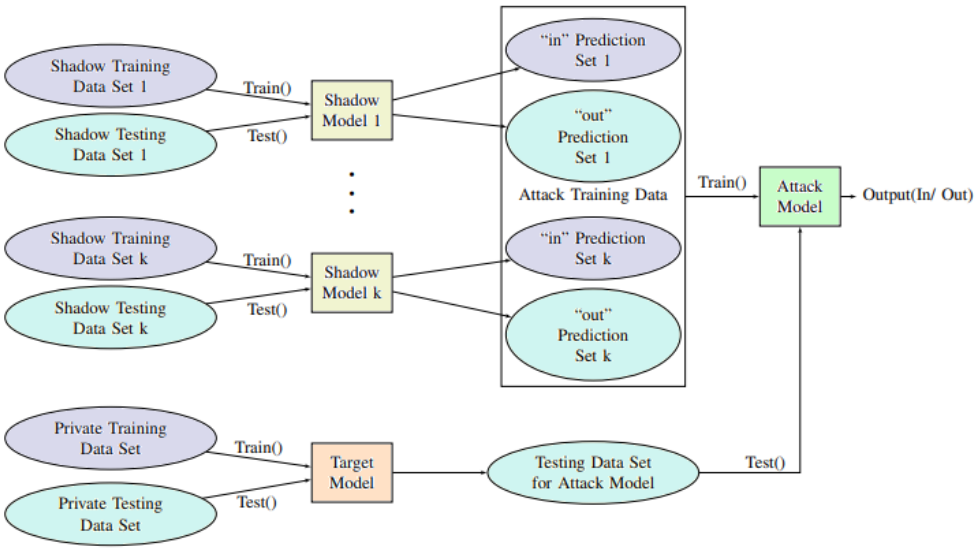

Figure 1. Overview of MIA on ML models [9].

The target model takes a data sample as input and generates the probability prediction vector after training. Suppose D^{Train}_{Target} is the private training dataset of the target model M_{Target}, where (x_i, y_i) are the labeled data records. In this labeled dataset, (x_i) represents the input to the target model, while (y_i) represents the class label of x_i in the set $1, 2, ..., C_{Target}$. The output of the target model M_{Target} is a vector of probabilities of size C_{Target}, where the elements range from 0 to 1 and they sum to 1. Multiple shadow models are created by the attacker to mimic the behavior of the target model and to generate the data needed to train the attack model. The attacker creates several (n) shadow models $M^i_{Shadow}()$, where each shadow model i is trained on the dataset D^i_{Shadow}. The attacker first splits its dataset D^i_{Shadow} into two sets, D^{iTrain}_{Shadow} and D^{iTest}_{Shadow}, such that $D^{iTrain}_{Shadow} \cap D^{iTrial}_{Shadow} = \phi$. Then, the

attacker trains each shadow model M^{i}_{Shadow} with the training set D^{iTrain}_{Shadow} and tests the same model with D^{iTest}_{Shadow} test dataset. The attack model is a collection of models, one for each output class of target data. D^{Train}_{Attack} is the attack model's training dataset, which contains labeled data records (x_i, y_i) and the probability vector generated by the shadow model for each data sample x_i. The label for x_i in the attack model is either "in" if x_i is used to train the shadow model or "out" if x_i is used to test the shadow model. This attack is named Attack 1 in our experiments.

3.2. Salem et al.'s MIA

Early demonstrations by Shokri et al. [9] on the feasibility of MIA are based on many assumptions, e.g. the use of multiple shadow models, knowledge of the structure of the target model, and the availability of a dataset from the same distribution as the training data of the target model. Salem et al [10] diminished all these key assumptions, showing that the MIA is generally applicable at low cost and carries greater risk than previously thought [10]. They provided two MIA attacks: I) with the knowledge of dataset distribution, model architecture, and only one shadow model, and II) with no knowledge about dataset distribution and model architecture. The former attack is named Attack 2 and the latter one is named Attack 3 in Table 2.

3.3. Prediction Sensitivity MIA

The idea behind this attack is that training data from a fully trained ML model generally have lower prediction sensitivity than untrained data (i.e., test data). The overview of this attack [11] is shown in Figure 2. The only allowed interaction between the attacker and the target model M is to query M with a sample x and then obtain the prediction result. The target model M maps the n-dimensional vector $x \in \mathbb{R}^n$ to the output m-dimensional $y \in \mathbb{R}^m$. The Jacobian matrix of M is a matrix $m \times n$ whose element in the ith row and jth column is $J_{ij} = \frac{\partial y_i}{\partial x_j} (i \in [1, 2, \ldots, n]$ and $j \in [1, 2, \ldots, m])$:

$$J(x; M) = \left[\frac{\partial M(x)}{\partial x_1} \ldots \frac{\partial M(x)}{\partial x_n} \right] = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{2}$$

where $y = M(x)$. The input sample is $x = [x_1, x_2, \ldots, x_n]$, and the corresponding prediction is $y = [y_1, y_2, \ldots, y_m]$. $\frac{\partial y_i}{\partial x_j}$ is the relationship between the change in the input record's i-th feature value and the change in the prediction probability that this sample belongs to j-th class.

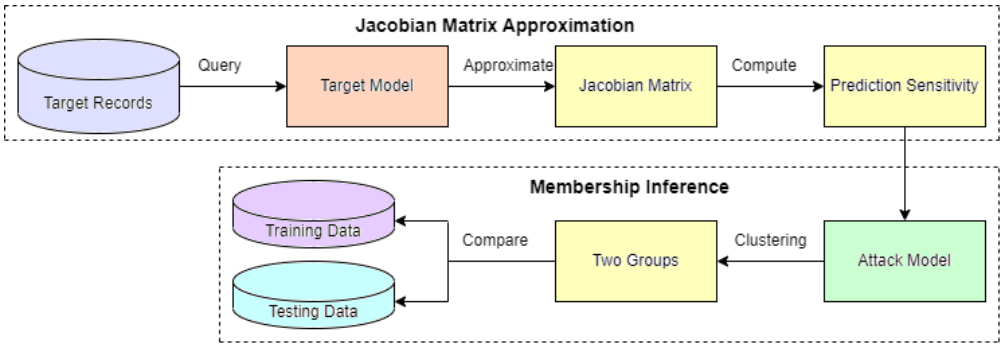

Figure 2. Overview of MIA using Jacobian matrix and prediction sensitivity [11].

The Jacobian matrix comprises a series of first-order partial derivatives. The derivatives can be approximated by calculating the numerical differentiation with the following equation:

$$\frac{\partial y_j}{\partial x_i} \approx \frac{M(x + \epsilon) - M(x - \epsilon)}{2\epsilon}, \tag{3}$$

where ϵ is a small value added to the input sample's i-th feature value. Add ϵ to the i-th feature value of the target sample x_t, whose membership property to know provides two modified samples to query the target model and derive the partial derivatives of the i-th feature for the target model: $\frac{\partial M(x)}{\partial x_i} = \left[\frac{\partial y_1}{\partial x_i}, \frac{\partial y_2}{\partial x_i} \ldots, \frac{\partial y_m}{\partial x_i}\right]$. Then, for each feature in x, this process is repeated to combine the partial derivatives into the Jacobian matrix. For simplicity, the approximation of the Jacobian matrix is defined as $J(x; M)$. The L-2 norm of $J(x; M)$ represents the prediction sensitivity for the target sample, as described by Novak et al. [41]. For a $m \times n$ matrix A, the L-2 norm of A can be computed as follows:

$$||A||_2 = (\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2)^{\frac{1}{2}} \tag{4}$$

where i and j are the row and column number of the matrix element a_{ij}, respectively. There is a difference in prediction sensitivity between samples from the training set and samples from the testing set. Once prediction sensitivity is calculated, an unsupervised clustering method (like k-means) partitions a set of target records (prediction sensitivity values) into two subsets. The cluster with the lowest mean sensitivity compared to the members of the M's training set is chosen. Then, during the inference stage, the samples are clustered into three or more groups and ordered by an average norm. Finally, the groups with lower average norms are predicted from the target model's training set, whereas others are not.

4. Defense Mechanisms

Attackers take advantage of the fact that ML models behave differently during the prediction with new data than with training data to differentiate members from nonmembers. This property is associated with the degree of overfitting, which is measured by the generalization gap. The generalization gap is the difference between the accuracy of the model between training and testing time. When overfitting is high, the model is more vulnerable to MIA. Therefore, whatever method is used to reduce overfitting is also profitable for MIA reduction. We applied the following methods to see how they mitigate the MIA.

- **Dropout (D):** It prevents overfitting by randomly deleting units in the neural network and allows for an approximately efficient combination of many different neural network architectures [16]. This was suggested by Salem et al. [10] and implemented as an MIA mitigation technique in ML models in a centralized framework.
- **Monte Carlo Dropout (MCD)**: It is proposed by Gal et al. [13]. It captures the uncertainty of the model. Various networks (where several neurons have been randomly disabled) can be visualized as Monte Carlo samples from the space of all available models. This provides a mathematical basis for the model to infer its uncertainty, often improving its performance. This work allows dropout to be applied to the neural network during model inference [42]. Therefore, instead of making one prediction, multiple predictions are made, one for each model (already prepared with random disabled neurons), and their distributions are averaged. Then, the average is considered as the final prediction.
- **Batch Normalization (BN)**: This is a technique that improves accuracy by normalizing activations in the middle layers of deep neural networks [14]. Normalization is used as a defense in label-only MIA, and the results show that both regularization and normalization can slightly decrease the average accuracy of the attack [32].
- **Gaussian Noise (GN)**: This is the most practical perturbation-based model for describing the nonlinear effects caused by additive Gaussian noise [23]. GN is used to ignore adversarial attacks [15].
- **Gaussian Dropout (GD)**: It is the integration of Gaussian noise with the random probability of nodes. Unlike standard dropout, nodes are not entirely deleted. Instead of ignoring neurons, they are subject to Gaussian noise. From Srivatsava's experiments [16], it appears that using the Gaussian dropout reduced computation time

because the weights did not have to be scaled each time to match the skipped nodes, as in the standard dropout.

- **Activity Regularization (AR)**: It is a technique used to encourage the model to have specific properties regarding the activations (outputs) of neurons in the network during training. The purpose of activity regularization is to prevent overfitting and encourage certain desirable characteristics in the network's behavior. The L1 regularizer and the L2 regularizer are two regularization techniques [24]. L1 regularization penalizes the sum of the absolute values of the weights, while L2 regularization penalizes the sum of the squares of the weights. Shokri et al. [9] used a conventional L2 regularizer as a defense technique to overcome MIA in ML neural network models.
- **Masking (M)**: It tells the sequence processing layers that some steps are missing from the input and should be ignored during data processing [17]. If all input tensor values in that timestep are equal to the mask value, the timestep is masked (ignored) in all subsequent layers of that timestep.
- **Differential Privacy (DP)**: Differentially Private Stochastic Gradient Descent (DPSGD) is a differentially private version of the Stochastic Gradient Descent (SGD) algorithm that happens during model training [18] and incorporates gradient updates with some additive Gaussian noise to provide differential privacy. DP [43–45] is a solid standard to ensure the privacy of distributed datasets.
- **Knowledge Distillation (KD)**: It distills and transfers knowledge from one deep neural network (DNN) to another DNN [19,46]. According to many MIA mitigation articles, KD outperforms the cutting edge approaches [33,34] in terms of MIA mitigation, while other FL articles support that it facilitates effective communication [47–49] to maintain the heterogeneity of the collaborating parties.
- **Combination of KD with AR (AR–KD)**: In our early experiments [12], we noticed that, in most test cases, KD lowers the recall while preserving the model accuracy. In this work, we are combining AR as a mitigation technique with KD. To the best of our knowledge, this is the first work that combines AR and KD and evaluates its results both in CL and FL.
- **Combination of KD with GN (GN–KD)**: Like AR, we are also combining GN and KD to see how they affect the attack recall and model accuracy. This paper is also the first paper that combines GN and KD and evaluates the performance of this combination in CL and FL environments.
- **Combination of KD with GD (GD–KD)**: We also combine KD and GD to see their effects on attack recall and model accuracy using five various optimizers on image datasets. To our knowledge, there is no work that combines these two methods to evaluate how they behave against MIA. Therefore, this is the first paper that combines these methods and analyses them in both CL and FL environments.

5. Performance Analysis

In this section, a summary of the experimental setup and results is provided. We performed our experiments on a 2.30 GHz 12th Gen Intel(R) Core(TM) i7-12700H processor with 16.00 GB RAM on the x64-based Windows 11 OS. We used open-source frameworks and standard libraries, such as Keras and Tensorflow in Python. The code of this work is available at [50].

5.1. Experimental Setup

In the following, we detail the experimental setup.

5.1.1. Datasets

The datasets of our experiments are CIFAR-10 [29], MNIST [27], FMNIST [28], and Purchase [30]. These datasets are the benchmark to validate the MIA, and they are the same as those used in recent related work [51]. CIFAR-10, MNIST, and FMNIST are image datasets in which, by normalizing, we fit the image pixel data in the range [0,1], which helps

to train the model more accurately. Purchase is a tabular dataset that has 600 dimensions and 100 labels. We used one-hot encoding of this dataset to be able to feed it into the neural network [51]. Each dataset is split into 30,000 for training and 10,000 for testing. For training in the FL environment, the training dataset is uniformly divided between three FL participants to train the local models based on the FedAvg [3] algorithm separately and update the central server to reach a global optimal model.

5.1.2. Model Architecture

The models are based on the Keras sequential function and a linear stack of neural network layers. In these models, we first defined the flattened input layer, followed by three dense layers. The MNIST and FMNIST input sizes are 28×28, while the CIFAR-10 input sizes are 32×32. The Purchase dataset input size is considered 600 since it has 600 features. We added all countermeasure layers in between the dense layers. As knowledge distillation is an architectural mitigation technique, we ran a separate experiment to see its performance. We specified an output size of 10 as the labels for each class in the MNIST, FMNIST, and CIFAR-10 datasets range between 0 and 9. Also, we set the output size of 100 for the Purchase dataset as the labels for this dataset range between 0 and 99. In addition, we set the activation function for the output layer to softmax to make the outputs sum to 1.

5.1.3. Training Setup

For training, we used SGD, RMSProp, Adagrad, Nadam, and Adadelta optimizers, with a learning rate equal to 0.01. The loss function for all the optimizers is the categorical cross-entropy. We have a batch size of 32 and epochs of 10 for each participant during training. We reproduced the FL process, including local participant training and FedAvg aggregation. The scheme of data flow is illustrated in Figure 3.

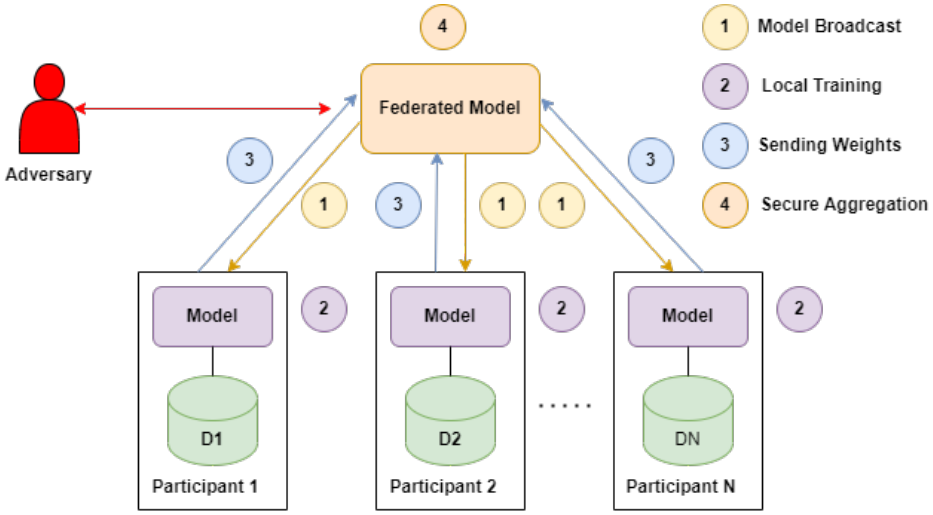

Figure 3. Overview of the FL system.

5.1.4. Evaluation Metrics

We focus on test accuracy as an evaluation metric for the FL model and recall as an evaluation metric for successful attacks in the FL setting. The recall (true positive rate) represents the fraction of the members of the training dataset that are correctly inferred as members by the attacker.

5.1.5. Comparison Methods

We investigated the performance of four attacks, as mentioned in Table 2. Attack 1 employs multiple shadow models mimicking both the structure and the data distribution of the target model. Attack 2 applies a single shadow model. The structure of the model is

different. However, the training data distribution imitates the target model. Unlike Attack 1 and Attack 2, in Attack 3, both the structure of the model and the training data distribution differ from the target model. Finally, Attack 4 applies the Jacobian matrix paradigm, which brings us an entirely different membership inference attack using the target model.

5.2. Experimental Results

In this section, we compared FL and CL. We also experimentally analyzed the effect of the MIA and the effect of the mitigation techniques in both environments, considering image and tabular datasets.

5.2.1. CL vs. FL

Many studies thoroughly compared the CL and FL approaches [52,53]. FL is concluded as a network-efficient alternative to CL [54]. In our comparison of the two approaches, as shown in Figures 4–7, CL outperformed FL regarding accuracy in most cases, which is expected. In Figure 7, the accuracy in CL is considerably lower than the accuracy in FL for GN, GD, and AR. This is justified by the nature of the tabular dataset, which seems to be overfitted using Adadelta and Nadam optimizers in the CL environment, and overfitting is removed when we apply these optimizers in the FL environment. The accuracy values are also tabulated in Tables 3 and 4 for CL and FL environments, respectively. In all figures and tables, the WC is the value for the model accuracy (or attack recall) without having any countermeasure included in the model. Figures 8–11 illustrate attack recall in our experiments. An interesting aspect to note is related to the Adadelta optimizer in image datasets. If we examine Adadelta's performance in image datasets in Figures 4–6, we can observe that there is minimal loss in accuracy when using this optimizer. However, our experiments depicted in Figures 8–10 indicate that, even when we do not implement any countermeasure (WC) to mitigate membership inference attacks (MIA), Adadelta is capable of functioning as a countermeasure without significantly compromising utility. It is evident that utilizing Adadelta alone results in a substantial reduction in the recall of the MIA attack. However, for tabular datasets, Adadelta is not performing significantly differently from other optimizers, as shown in Figure 11. In all the tables in this paper, the value in parentheses shows the difference between that countermeasure and its corresponding value in the without countermeasure (WC) column. WC shows the values when we do not use any countermeasure.

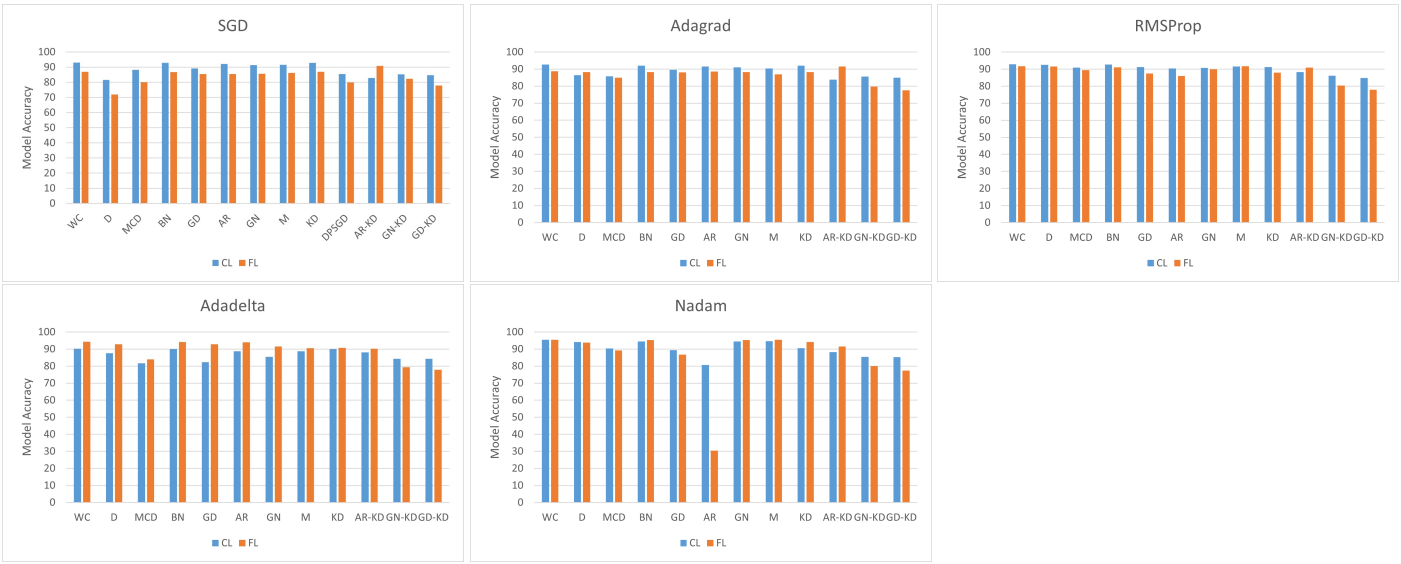

Figure 4. Comparison of model accuracy of CL and FL using various optimizers and countermeasures—MNIST dataset.

Table 3. CL model accuracy.

Datasets	Optimizers	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
MNIST	SGD	93.1	81.6(−11.5)	88.3(−4.8)	92.8(−0.3)	89.1(−4)	92.1(−1)	91.4(−1.7)	91.6(−1.5)	**92.8(−0.3)**	85.5(−7.6)	82.8(−10.3)	85.2(−7.9)	84.8(−8.3)
	Adagrad	92.6	86.5(−6.1)	85.7(−6.9)	**92.1(−0.5)**	89.6(−3)	91.6(−1)	91.1(−1.5)	90.3(−2.3)	92(−0.6)	-	83.8(−8.8)	85.6(−7)	84.9(−7.7)
	RMSProp	92.8	92.5(−0.3)	90.9(−1.9)	**92.7(−0.1)**	91.3(−1.5)	90.4(−2.4)	90.7(−2.1)	91.5(−1.3)	91.2(−1.6)	-	88.3(−4.5)	86.1(−6.7)	84.8(−8)
	Nadam	**95.5**	94.1(−1.4)	90.3(−5.2)	94.4(−1.1)	89.4(−6.1)	80.7(−14.8)	94.5(−1)	**94.7(−0.8)**	90.6(−4.9)	-	88.3(−7.2)	85.4(−10.1)	85.2(−10.3)
	Adadelta	90.2	87.5(−2.7)	81.6(−8.6)	**90.1(−0.1)**	82.3(−7.9)	88.8(−1.4)	85.4(−4.8)	88.8(−1.4)	90.1(−0.1)	-	88(−2.2)	84.3(−5.9)	84.3(−5.9)
FMNIST	SGD	**88.6**	84.7(−3.9)	84.1(−4.5)	86.4(−2.2)	85.2(−3.3)	87.3(−1.3)	85.8(−2.8)	**88.1(−0.5)**	86.9(−1.7)	84.2(−4.4)	82.2(−6.4)	74.7(−13.9)	73.2(−15.4)
	Adagrad	85.8	78.6(−7.2)	75.9(−9.9)	84.6(−1.2)	81.9(−3.9)	83.8(−2)	79.1(−6.7)	**88.7(+2.9)**	82.3(−3.5)	-	83.1(−2.7)	75.3(−10.5)	73.8(−12)
	RMSProp	83.6	79.5(−4.1)	78.5(−5.1)	82.9(−0.7)	78.7(−4.9)	82.6(−1)	81.7(−1.9)	**83.1(−0.5)**	81.9(−1.7)	-	81.5(−2.1)	75.5(−8.1)	73.2(−10.4)
	Nadam	85.9	82.5(−3.4)	78.8(−7.1)	**83.4(−2.5)**	74.4(−11.5)	66.2(−19.7)	83(−2.9)	83.2(−2.7)	82.5(−3.4)	-	82.4(−3.5)	75.6(−10.3)	72.6(−13.3)
	Adadelta	83.8	77.6(−6.2)	75.1(−8.7)	**82.7(−1.1)**	73(−10.8)	77.4(−6.4)	77.8(−6)	81.8(−2)	82.3(−1.5)	-	82(−1.8)	74.2(−9.6)	71.3(−12.5)
CIFAR-10	SGD	85.7	74.2(−11.5)	74.9(−10.8)	82.8(−2.9)	83.1(−2.6)	**84.3(−1.4)**	78.6(−7.1)	82.5(−3.2)	83.9(−1.8)	74.6(−11.1)	82.6(−3.1)	75.1(−10.6)	74.2(−11.5)
	Adagrad	**88.4**	75.7(−12.7)	72.6(−15.8)	85.8(−2.6)	83.6(−4.8)	**87.2(−1.2)**	81.9(−6.5)	82.5(−5.9)	85.3(−3.1)	-	83.9(−4.5)	76.3(−12.1)	75.1(−13.3)
	RMSProp	86.3	81.6(−4.7)	76.4(−9.9)	**85.7(−0.6)**	84.9(−1.4)	84.2(−2.1)	82.1(−4.2)	83.6(−2.7)	81.5(−4.8)	-	80.5(−5.8)	77(−9.3)	75.9(−10.4)
	Nadam	75.3	73.4(−1.9)	73.6(−1.7)	**81.6(6.3)**	80.6(5.3)	72.3(7)	78.2(2.9)	80.8(5.5)	81.1(5.8)	-	79.8(4.5)	76.2(0.9)	75.6(0.3)
	Adadelta	78.2	71.2(−7)	70.5(−7.7)	83.1(4.9)	82.1(3.9)	75.5(7.3)	79.4(1.2)	81.3(3.1)	**84.5(6.3)**	-	82.1(3.9)	75.9(−2.3)	73.8(−4.4)
Purchase	SGD	79.3	72(−7.3)	79.2(−0.1)	70.5(−8.8)	57(−22.3)	3.8(−75.5)	76.8(−2.5)	79.8(0.5)	79.2(−0.1)	70.3(−9)	**82.6(3.3)**	75.1(−4.2)	74.2(−5.1)
	Adagrad	**82.6**	76.2(−6.4)	83.1(0.5)	69.2(−13.4)	64.7(−17.9)	4.4(−78.2)	80.7(−1.9)	82.9(0.3)	78.8(−3.8)	-	**83.9(1.3)**	76.3(−6.3)	75.1(−7.5)
	RMSProp	56.4	24.1(−32.3)	51.5(−4.9)	67.4(11)	8.5(−47.9)	5.2(−51.2)	51.7(−4.7)	52.6(−3.8)	77.8(21.4)	-	**80.5(24.1)**	77(20.6)	75.9(19.5)
	Nadam	65.1	41.4(−23.7)	66.6(1.5)	68.9(3.8)	13.2(−51.9)	8.2(−56.9)	60.9(−4.2)	67.5(2.4)	79.4(14.3)	-	**79.8(14.7)**	76.2(11.1)	75.6(10.5)
	Adadelta	28.1	15.5(−12.6)	29.1(1)	24.6(−3.5)	3.1(−25)	2.5(−25.6)	14.3(−13.8)	29.1(1)	80.3(52.2)	-	**82.1(54)**	75.9(47.8)	73.8(45.7)

Table 4. FL model accuracy.

Datasets	Optimizers	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
MNIST	SGD	87	72(−15)	80(−7)	86.7(−0.3)	85.5(−1.5)	85.4(−1.6)	85.6(−1.4)	86.2(−0.8)	86.9(−0.1)	79.9(−7.1)	**90.9(3.9)**	82.3(−4.7)	77.9(−9.1)
	Adagrad	88.7	88.2(−0.5)	84.9(−3.8)	88.3(−0.4)	88.1(−0.6)	88.5(−0.2)	88.3(−0.4)	87(−1.7)	88.2(−0.5)	-	**91.6(2.9)**	79.7(−9)	77.5(−11.2)
	RMSProp	91.7	91.6(−0.1)	89.5(−2.2)	91.1(−0.6)	87.4(−4.3)	86(−5.7)	90(−1.7)	**91.7(0)**	87.9(−3.8)	-	90.9(−0.8)	80.4(−11.3)	78(−13.7)
	Nadam	**95.5**	93.9(−1.6)	89.3(−6.2)	95.3(−0.2)	86.7(−8.8)	30.4(−65.1)	95.3(−0.2)	**95.4(−0.1)**	94.1(−1.4)	-	91.5(−4)	80.1(−15.4)	77.4(−18.1)
	Adadelta	94.3	92.8(−1.5)	83.9(−10.4)	**94.1(−0.2)**	92.8(−1.5)	94(−0.3)	91.6(−2.7)	90.5(−3.8)	90.7(−3.6)	-	90.2(−4.1)	79.3(−15)	77.9(−16.4)
FMNIST	SGD	81.3	79.8(−1.5)	76.4(−4.9)	81(−0.3)	75.7(−5.6)	80.5(−0.8)	74.9(−6.4)	80.3(−1)	80.9(−0.4)	77.6(−3.7)	**83.5(2.2)**	73.3(−8)	69.4(−11.9)
	Adagrad	82.6	82(−0.6)	78.9(−3.7)	81.6(−1)	79.9(−2.7)	**82.2(−0.4)**	78.4(−4.2)	80.6(−2)	80.7(−1.9)	-	**83.5(0.9)**	73.2(−9.4)	69.6(−13)
	RMSProp	**91.7**	76.8(−14.9)	74.4(−17.3)	71(−20.7)	72.3(−19.4)	68.2(−23.5)	55(−36.7)	76.1(−15.6)	75.8(−15.9)	-	**83.1(−8.6)**	73.3(−18.4)	69.6(−22.1)
	Nadam	84.6	82.2(−2.4)	78.6(−6)	83(−1.6)	69.3(−15.3)	52.5(−32.1)	73.8(−10.8)	83.2(−1.4)	**83.9(−0.7)**	-	83.3(−1.3)	73.3(−11.3)	69(−15.6)
	Adadelta	84.1	81.1(−3)	75.8(−8.3)	81.1(−3)	72(−12.1)	81.6(−2.5)	75.4(−8.7)	82.8(−1.3)	83(−1.1)	-	**83(−1.1)**	72.8(−11.3)	67.8(−16.3)
CIFAR-10	SGD	79.5	73(−6.5)	65.9(−13.6)	79.3(−0.2)	73.2(−6.3)	74.6(−4.9)	74.9(−4.6)	73.1(−6.4)	79.3(−0.2)	75.7(−3.8)	**82.8(3.3)**	75.3(−4.2)	68.5(−11)
	Adagrad	76.3	67(−9.3)	54(−22.3)	76.2(−0.1)	72.9(−3.4)	71.4(−4.9)	71.1(−5.2)	69.9(−6.4)	75.7(−0.6)	-	**79.7(3.4)**	70.2(−6.1)	66.7(−9.6)
	RMSProp	72.8	61.2(−11.6)	53.6(−19.2)	**72.4(−0.4)**	70.9(−1.9)	72.2(−0.6)	71(−1.8)	68.6(−4.2)	72.1(−0.7)	-	71.3(−1.5)	64.8(−8)	61.1(−11.7)
	Nadam	**80.3**	75.6(−4.7)	70.9(−9.4)	**79.8(−0.5)**	72.3(−8)	74.6(−5.7)	75.2(−5.1)	76.8(−3.5)	77.6(−2.7)	-	79.4(−0.9)	70.8(−9.5)	70.3(−10)
	Adadelta	78.6	77.8(−4.8)	72.6(−10)	78.2(−4.4)	70.1(−12.5)	71.3(−11.3)	77.8(−4.8)	75.9(−6.7)	76.3(−6.3)	-	**81.5(−1.1)**	65.8(−16.8)	66.8(−15.8)
Purchase	SGD	78.9	77.8(−1.1)	78.5(−0.4)	78.3(−0.6)	79(0.1)	**79.6(0.7)**	78.3(−0.6)	79.3(0.4)	78.8(−0.1)	76.5(−2.4)	78(−0.9)	75.5(−3.4)	43.9(−35)
	Adagrad	**81.3**	80.2(−1.1)	80.4(−0.9)	80.4(−0.9)	80(−1.3)	**81.4(0.1)**	80.2(−1.1)	80.1(−1.2)	77.6(−3.7)	-	79(−2.3)	77.4(−3.9)	43.8(−37.5)
	RMSProp	21.6	20.6(−1)	19.4(−2.2)	22.7(1.1)	23.8(2.2)	21.8(0.2)	23(1.4)	23.9(2.3)	76.6(55)	-	**78.8(57.2)**	76.9(55.3)	46.8(25.2)
	Nadam	30.1	28.7(−1.4)	32(1.9)	29.9(−0.2)	29.3(−0.8)	28.6(−1.5)	27.9(−2.2)	25.5(−4.6)	**79.3(49.2)**	-	78.8(48.7)	74.1(44)	45.3(15.2)
	Adadelta	30.8	29.6(−1.2)	32.8(2)	34.4(3.6)	33.8(3)	32.2(1.4)	32.6(1.8)	31.4(0.6)	**78.6(47.8)**	-	77.8(47)	77(46.2)	42.9(12.1)

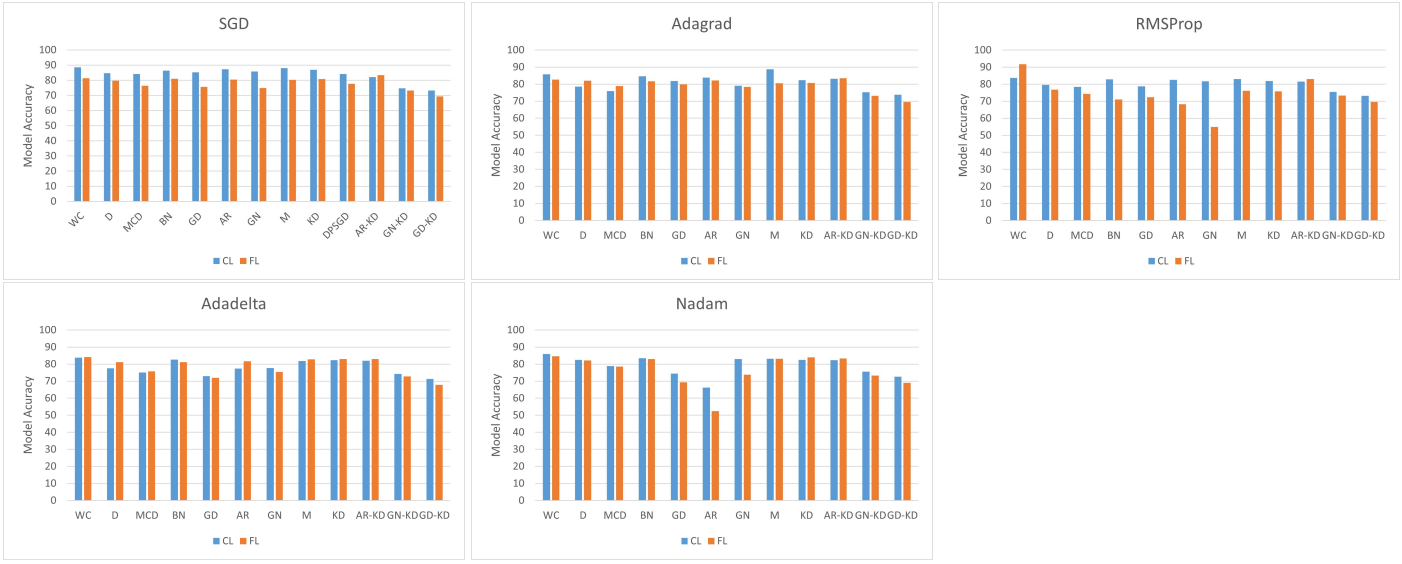

Figure 5. Comparison of model accuracy of CL and FL using various optimizers and countermeasures—FMNIST dataset.

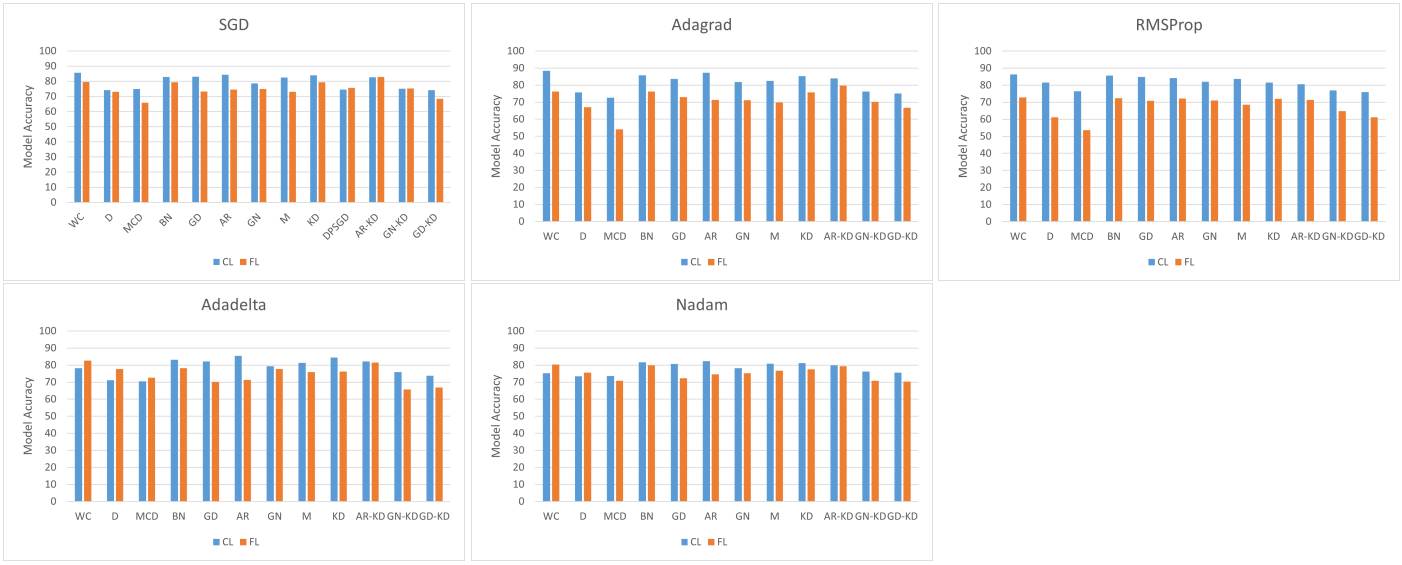

Figure 6. Comparison of model accuracy of CL and FL using various optimizers and countermeasures—CIFAR-10 dataset.

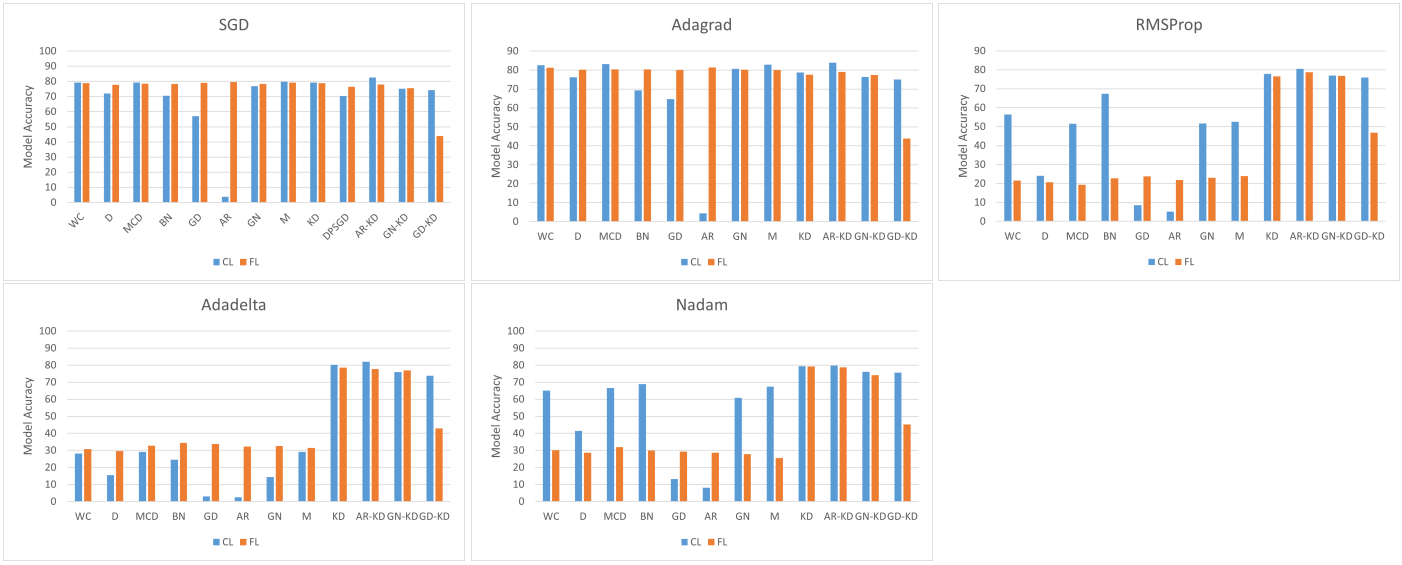

Figure 7. Comparison of model accuracy of CL and FL using various optimizers and countermeasures—Purchase dataset.

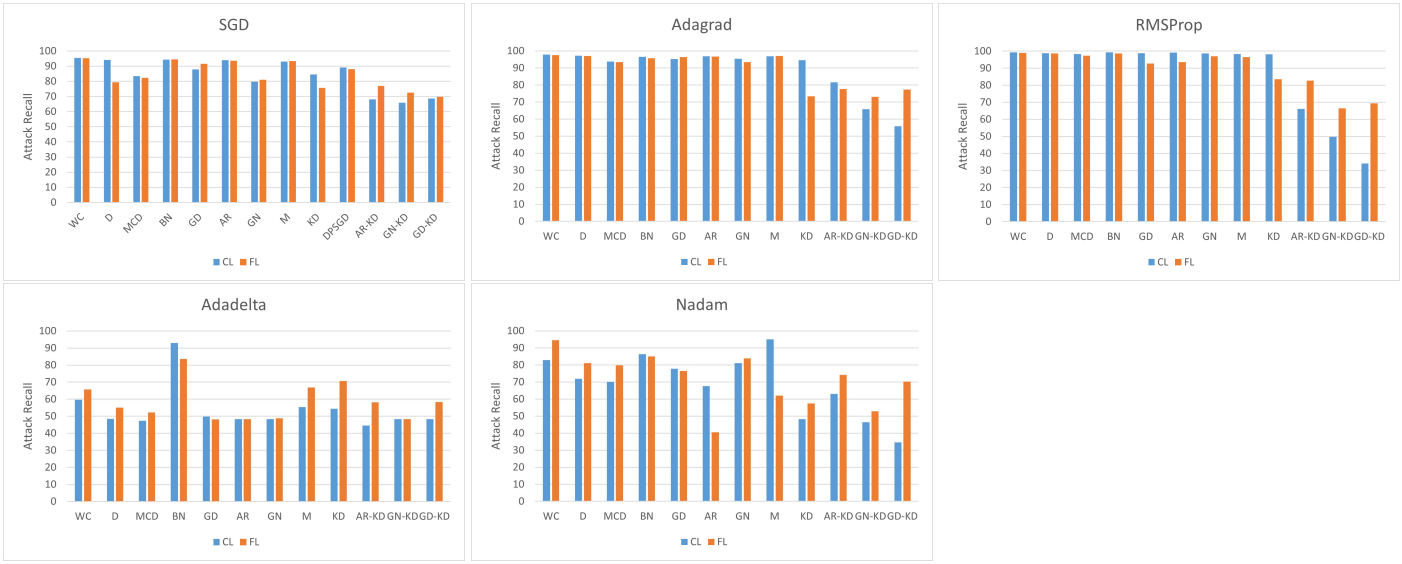

Figure 8. Comparison of Attack 1 recall on CL and FL using various optimizers and countermeasures—MNIST dataset.

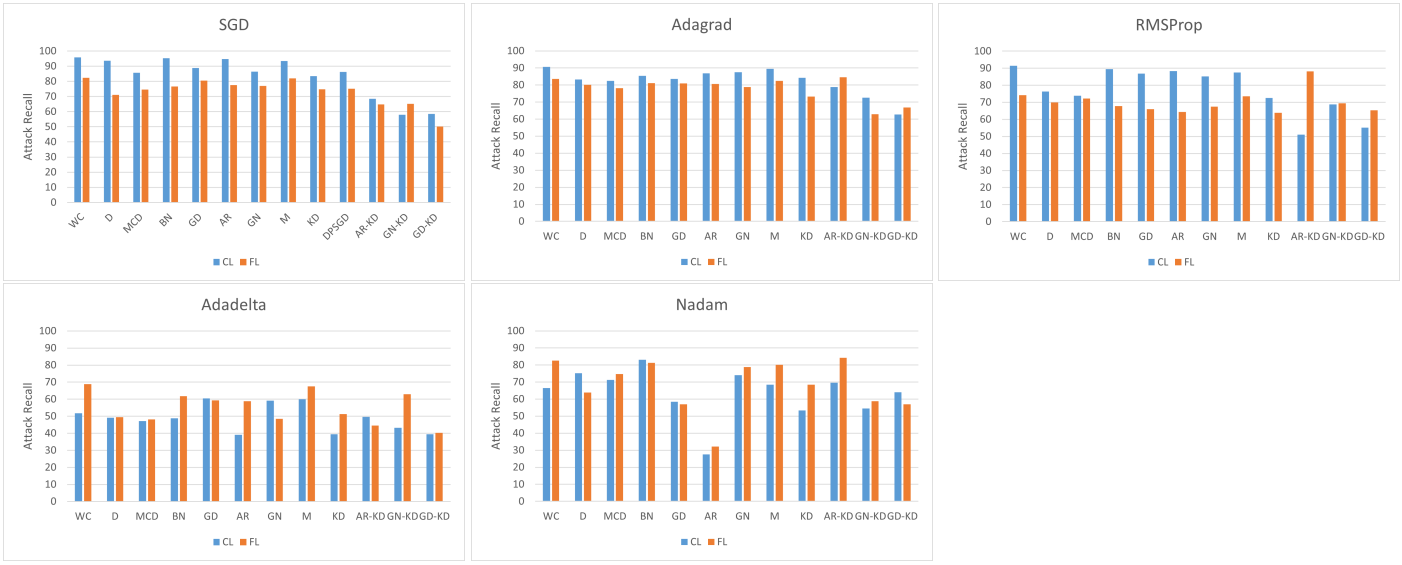

Figure 9. Comparison of Attack 1 Recall on CL and FL using various optimizers and countermeasures—FMNIST dataset.

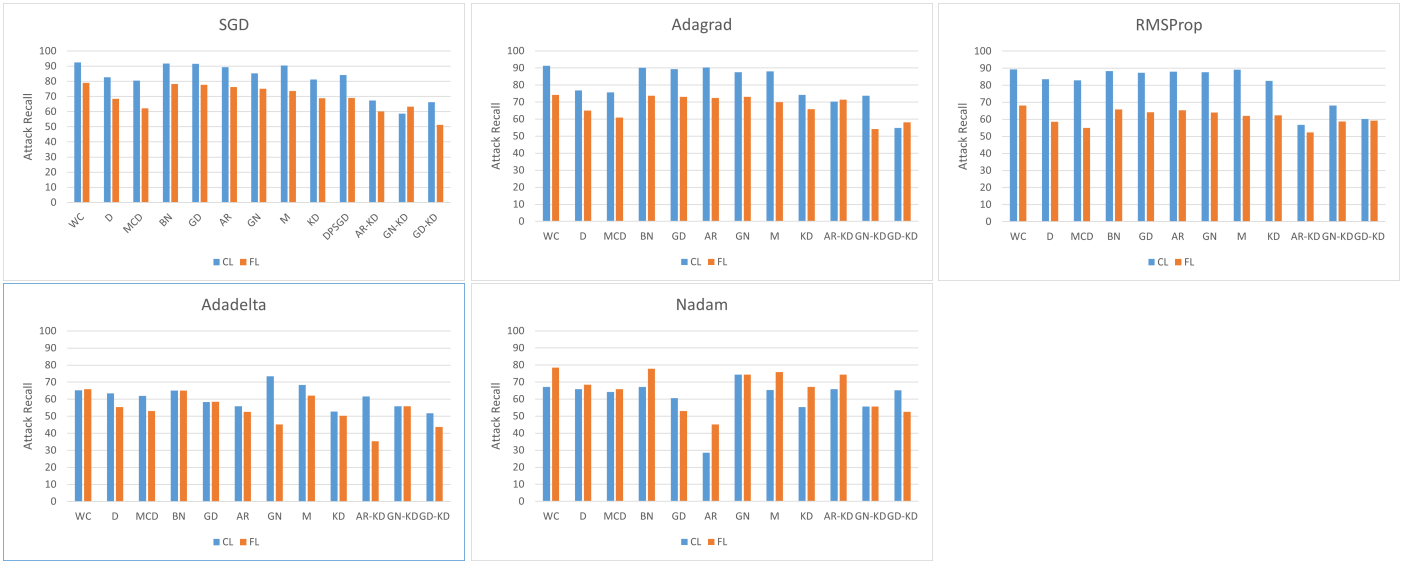

Figure 10. Comparison of Attack 1 recall on CL and FL using various optimizers and countermeasures—CIFAR-10 dataset.

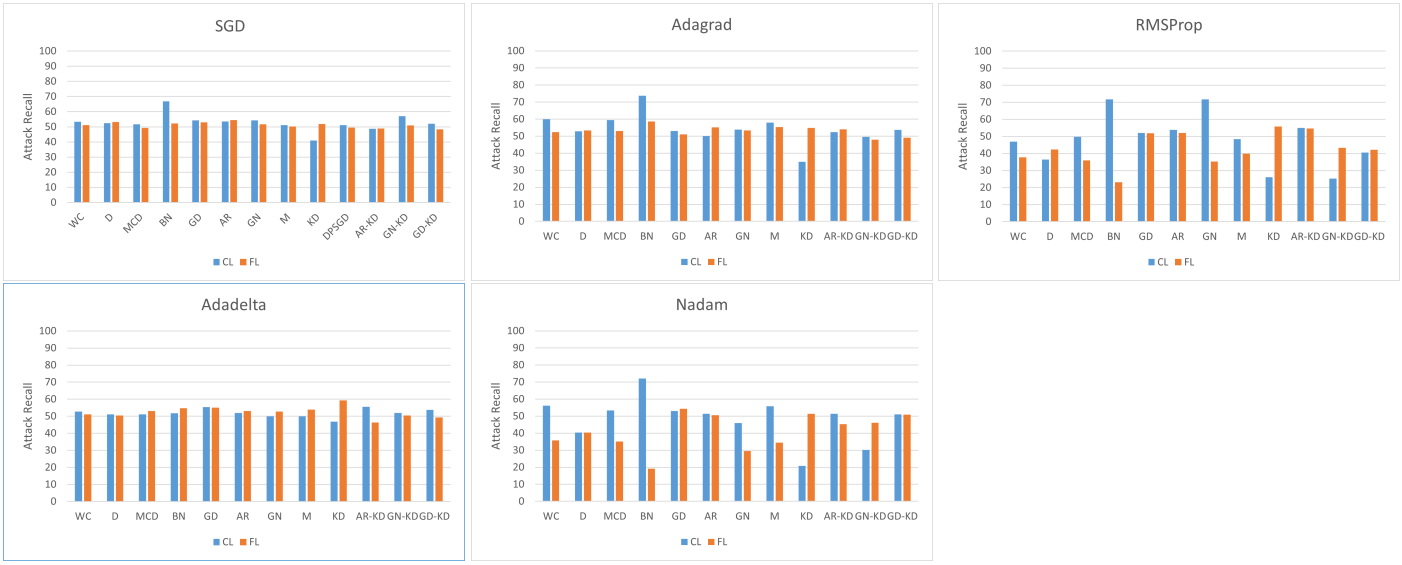

Figure 11. Comparison of Attack 1 recall on CL and FL using various optimizers and countermeasures—Purchase dataset.

Generally, the recall of Attack 1 is almost the same, if not less, in FL compared to the recall in CL considering different mitigation techniques. Figure 12 illustrates five various optimizers' effects as well as various countermeasures' effects on the FL model accuracy, where the y-axis provides the test accuracy of the FL model. As DP-SGD is specialized for SGD optimizer, we applied DP only on SGD optimizer and not with other optimizers. The first group in all the plots is WC, which represents the baseline without countermeasures. We have provided the full details of our experiments in CL and FL environments in Tables 3 and 4, respectively.

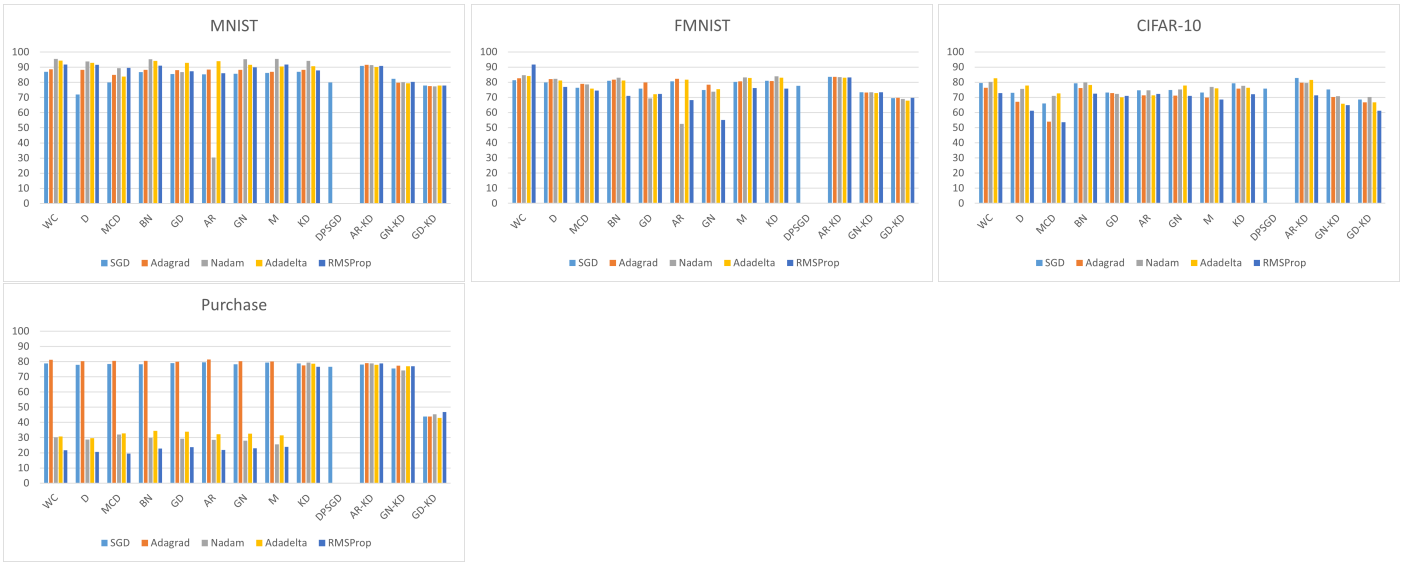

Figure 12. A comparison of FL model accuracy with five various optimizers, with and without countermeasures—MNIST, FMNIST, CIFAR-10, and Purchase datasets.

- **CL model accuracy without countermeasure:** As per Table 3, the highest CL model accuracy results for Nadam, SGD, Adagrad, and Adagrad on the MNIST, FMNIST, CIFAR-10, and Purchase datasets, respectively. In contrast, Nadam on the CIFAR-10, Adadelta on MNIST, FMNIST, and Purchase yield the lowest accuracy. Generally speaking, depending on the dataset, the optimizer, and the batch size used in each round of training, the values for the model accuracy change.

- **CL model accuracy with countermeasures**: As per Table 4, the combination that yields the highest CL model accuracy for MNIST after applying countermeasures belongs to Nadam with M. When we apply M as the countermeasure and Nadam as the optimizer, the accuracy of the model slightly decreases compared to the case when we use no countermeasure (WC). Subsequently, this is followed by an increase in the attack recall when using Nadam with M, as per Table 5. In general, Nadam with M slightly decreases model accuracy and significantly increases attack recall for the MNIST dataset, while Adadelta with MCD provides the lowest model accuracy. For the FMNIST dataset, when we use Adagrad with M, we have even higher accuracy than no countermeasure. However, the attack recall is subsequently high, as shown in Table 5. In CIFAR-10, AR and BN hold the highest accuracy, while MCD has the lowest accuracy. In the Purchase dataset, AR–KD yields the highest accuracy for all optimizers, even better than without countermeasures. This happens while attack recall in the Purchase dataset, as per Table 5, is reduced for SGD and Adagrad.
- **FL model accuracy without countermeasure**: As shown in Table 4, the highest FL model accuracy belongs to Nadam, RMSProp, Adadelta, and Adagrad on the MNIST, FMNIST, CIFAR-10, and Purchase datasets, respectively, whereas RMSProp on the CIFAR-10 and Purchase datasets as well as SGD on MNIST and FMNIST yield the lowest accuracy. In general, FL model accuracy is the lowest for Purchase and the highest for MNIST. This is justified by the nature of the datasets and the distribution of their features, which make each data record more distinguishable from the others. The reason why some optimizers are performing very well for specific datasets in the CL environment and not performing well for the same dataset in the FL environment is that these optimizers are sensitive to the FedAvg algorithm, where we average the total weights that are computed locally by the clients to generate the global model.
- **FL model accuracy with countermeasures**: As per Table 4, BN has no significant effect on the CIFAR-10 model accuracy. For CIFAR-10, the highest accuracy belongs to AR–KD when using the SGD optimizer, and the lowest accuracy belongs to MCD when using the Adagrad optimizer. For MNIST and FMNIST, the countermeasure that maintains the maximum accuracy varies between different optimizers. For instance, in FMNIST, the mitigation technique that keeps the model accuracy at its maximum value is AR–KD for four optimizers: SGD, Adagrad, RMSProp, and Adadelta. Also, for Nadam, KD yields the highest accuracy in the FL environment. For FMNIST, the lowest accuracy belongs to AR when using the Nadam optimizer. For MNIST, the best accuracy goes for AR–KD when using SGD and Adagrad, whereas M provides the highest accuracy in RMSProp and Nadam. Also, BN provides the highest accuracy when using Adadelta. The lowest accuracy for MNIST belongs to GD–KD when using the Nadam optimizer. The highest accuracy for the Purchase dataset belongs to AR when using SGD and Adagrad, as well as KD when using Nadam and Adadelta.

Table 5. CL attack recall.

Datasets	Optimizers	Attacks	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
MNIST	SGD	Attack-1	95.40	94.2(−1.2)	83.4(−12)	94.4(−1)	87.9(−7.5)	94(−1.4)	79.8(−15.6)	93.1(−2.3)	84.6(−10.8)	89.2(−6.2)	68.1(−27.3)	**65.8(−29.6)**	68.6(−26.8)
		Attack-2	94.90	93.7(−1.2)	83.2(−11.7)	94.8(−0.1)	86.8(−8.1)	93.6(−1.3)	93.2(−1.7)	93.9(−1)	82.4(−12.5)	88.3(−6.6)	65.4(−29.5)	**63.2(−31.7)**	65.2(−29.7)
		Attack-3	90.70	85.3(−5.4)	74.5(−16.2)	88.7(−2)	82.9(−7.8)	83.2(−7.5)	88.6(−2.1)	89.3(−1.4)	80.5(−10.2)	82.4(−8.3)	63.7(−27)	**60.4(−30.3)**	63.4(−27.3)
		Attack-4	87.10	32(−55.1)	34.6(−52.5)	28.7(−58.4)	24.5(−62.2)	35.3(−51.8)	37.4(−49.7)	24.6(−62.5)	**22.6(−64.5)**	26.8(−60.3)	24.7(−62.4)	32.5(−54.6)	28(−59.1)
	Adagrad	Attack-1	97.80	97.2(−0.6)	93.8(−4)	96.6(−1.2)	95.2(−2.6)	96.9(−0.9)	95.4(−2.4)	96.9(−0.9)	94.6(−3.2)	-	81.6(−16.2)	65.9(−31.9)	**55.9(−41.9)**
		Attack-2	97.70	92.4(−5.3)	93.5(−4.2)	95.7(−2)	95.1(−2.6)	96.4(−1.3)	93.6(−4.1)	95.9(−1.8)	76.1(−21.6)	-	66.2(−31.5)	64.1(−33.6)	**55.3(−42.4)**
		Attack-3	91.30	87.4(−3.9)	76.6(−14.7)	89.5(−1.8)	86.2(−5.1)	86.3(−5)	81.3(−10)	85.4(−5.9)	74.9(−16.4)	-	63.4(−27.9)	58.9(−32.4)	**52.1(−39.2)**
		Attack-4	86.90	33.6(−53.3)	32.1(−54.8)	35.2(−51.7)	34.1(−52.8)	39.7(−47.2)	31.8(−55.1)	35.3(−51.6)	31.4(−55.5)	-	**20(−66.9)**	96(9.1)	84(−2.9)
	RMSProp	Attack-1	99.40	98.9(−0.5)	98.3(−1.1)	99.3(−0.1)	98.8(−0.6)	99.1(−0.3)	98.6(−0.8)	98.4(−1)	98.2(−1.2)	-	66.2(−33.2)	49.7(−49.7)	**34.2(−65.2)**
		Attack-2	99.20	98.6(−0.6)	97.3(−1.9)	98.7(−0.5)	97.2(−2)	98.3(−0.9)	97.7(−1.5)	97(−2.2)	96.4(−2.8)	-	64.3(−34.9)	45.5(−53.7)	**33.1(−66.1)**
		Attack-3	98.60	96.3(−2.3)	94.1(−4.5)	98.2(−0.4)	93.1(−5.5)	95.6(−3)	94.3(−4.3)	94.8(−3.8)	92.5(−6.1)	-	58.9(−39.7)	43.8(−54.8)	**32.4(−66.2)**
		Attack-4	89.90	23(−66.9)	38.7(−51.2)	39.5(−50.4)	37.3(−52.6)	39.4(−50.4)	34.6(−55.3)	32.8(−57.1)	31.3(−58.6)	-	24(−65.9)	**16(−73.9)**	20(−69.9)
	Nadam	Attack-1	82.90	71.9(−11)	70.1(−12.8)	86.3(3.4)	77.8(−5.1)	67.7(−15.2)	81.1(−1.8)	95(12.1)	48.3(−34.6)	-	63.1(−19.8)	46.5(−36.4)	**34.6(−48.3)**
		Attack-2	80.70	70.3(−10.4)	68.5(−12.2)	85.6(4.9)	76.4(−4.3)	66.4(−14.3)	79.5(−1.2)	92.9(12.2)	45.5(−35.2)	-	61.2(−19.5)	44.6(−36.1)	**32.1(−48.6)**
		Attack-3	78.50	71.5(−7)	69.4(−9.1)	77.5(−1)	65.1(−13.4)	58.9(−19.6)	73.6(−4.9)	78.2(−0.3)	44.7(−33.8)	-	59.8(−18.7)	39.5(−39)	**29.8(−48.7)**
		Attack-4	80.30	69(−11.3)	67.9(−12.4)	78(−2.3)	32(−48.3)	28(−52.3)	84.4(4.1)	43.3(−37)	52(−28.3)	-	30(−50.3)	24.2(−56.1)	**12(−68.3)**
	Adadelta	Attack-1	59.70	48.4(−11.3)	47.3(−12.4)	93(33.3)	49.8(−9.9)	48.3(−11.4)	48.3(−11.4)	55.3(−4.4)	54.4(−5.3)	-	**44.6(−15.1)**	48.3(−11.4)	48.3(−11.4)
		Attack-2	58.80	47.3(−11.5)	46.7(−12.1)	91.8(33)	45.5(−13.3)	47.8(−11)	44.2(−14.6)	52.8(−6)	53.1(−5.7)	-	**43.1(−15.7)**	45.6(−13.2)	46.9(−11.9)
		Attack-3	56.50	43.6(−12.9)	46.8(−9.7)	92.2(35.7)	49.3(−7.2)	45.6(−10.9)	**41.3(−15.2)**	51.9(−4.6)	52.2(−4.3)	-	42.6(−13.9)	43.5(−13)	45.8(−10.7)
		Attack-4	44.00	36(−8)	34.8(−9.2)	76.2(32.2)	**8(−36)**	20(−24)	20(−24)	12(−32)	32.8(−11.2)	-	16(−28)	24(−20)	20(−24)
FMNIST	SGD	Attack-1	95.8	93.7(−2.1)	85.6(−10.2)	95.3(−0.5)	88.9(−6.9)	94.7(−1.1)	86.5(−9.3)	93.4(−2.4)	83.5(−12.3)	86.3(−9.5)	68.4(−27.4)	**57.9(−37.9)**	58.4(−37.4)
		Attack-2	93.6	86.4(−7.2)	83.2(−10.4)	93.1(−0.5)	85.9(−7.7)	92.8(−0.8)	86.1(−7.5)	92.5(−1.1)	82.9(−10.7)	85.7(−7.9)	65.2(−28.4)	**55.6(−38)**	56.2(−37.4)
		Attack-3	90.2	82.2(−8)	81.6(−8.6)	89.6(−0.6)	84.9(−5.3)	89.1(−1.1)	83.7(−6.5)	88.5(−1.7)	81.9(−8.3)	83.1(−7.1)	61.3(−28.9)	**51.2(−39)**	54.2(−36)
		Attack-4	82.1	27(−55.1)	25.8(−56.3)	33.6(−48.5)	38.4(−43.7)	42.8(−39.3)	29.5(−52.6)	37.5(−44.6)	21.7(−60.4)	27.9(−54.2)	35.9(−46.2)	22.8(−59.3)	**8(−74.1)**
	Adagrad	Attack-1	90.6	83.2(−7.4)	82.4(−8.2)	85.4(−5.2)	83.6(−7)	86.9(−3.7)	87.6(−3)	89.5(−1.1)	84.2(−6.2)	-	78.8(−11.8)	72.6(−18)	**62.7(−27.9)**
		Attack-2	87.2	82.6(−4.6)	81.6(−5.6)	84.3(−2.9)	81.9(−5.3)	85.3(−1.9)	86.8(−0.4)	85.7(−1.5)	81.8(−5.4)	-	76.5(−10.7)	68.5(−18.7)	**58.6(−28.6)**
		Attack-3	85.7	82.4(−3.3)	81.1(−4.6)	82.8(−2.9)	78.4(−7.3)	82.9(−2.8)	82.7(−3)	83.6(−2.1)	79.3(−6.4)	-	75.1(−10.6)	65.3(−20.4)	**55.9(−29.8)**
		Attack-4	80.9	46.2(−34.7)	26.9(−54)	36.7(−44.2)	26.4(−54.5)	35.4(−45.5)	32(−48.9)	43.2(−37.7)	31.2(−49.7)	-	88(7.1)	68(−12.9)	**24(−56.9)**
	RMSProp	Attack-1	91.5	76.4(−15.1)	73.8(−17.7)	89.4(−2.1)	86.9(−4.6)	88.4(−3.1)	85.2(−6.1)	87.5(−4)	72.6(−18.9)	-	**51(−40.5)**	68.7(−22.8)	55.1(−36.4)
		Attack-2	89.2	73.6(−15.6)	72.7(−16.5)	86.3(−2.9)	82.8(−6.4)	87.2(−2)	83.1(−6.1)	85.3(−3.9)	71.8(−17.4)	-	**47.3(−41.9)**	65.3(−23.9)	52.6(−36.6)
		Attack-3	85.3	71.9(−13.4)	70.6(−14.7)	84.3(−1)	81.4(−3.9)	82.8(−2.5)	80.1(−5.2)	81.6(−3.7)	69.3(−16)	-	**45.9(−39.4)**	58.8(−26.5)	50.8(−34.5)
		Attack-4	70.8	20.1(−50.7)	20.5(−50.3)	35.9(−34.9)	26.2(−44.6)	34.8(−36)	28.6(−42.2)	29.6(−41.2)	26.1(−44.7)	-	**24(−46.8)**	92(21.2)	28(−42.8)
	Nadam	Attack-1	66.5	75.2(8.7)	71.2(4.7)	83(16.5)	58.4(−8.1)	**27.5(−39)**	74.1(7.6)	68.4(1.9)	53.3(−13.2)	-	69.6(3.1)	54.5(−12)	64.1(−2.4)
		Attack-2	65.7	73.5(7.8)	70.4(4.7)	81.3(15.6)	55.3(−10.4)	**25.6(−40.1)**	73.1(7.4)	65.2(−0.5)	51.6(−14.1)	-	65.8(0.1)	52.6(−13.1)	63.2(−2.5)
		Attack-3	60.3	69.8(9.5)	65.9(5.6)	79.7(19.4)	51.9(−8.4)	**26.6(−33.7)**	70.4(10.1)	62.8(2.5)	49.8(−10.5)	-	64.2(3.9)	50.8(−9.5)	61.5(1.2)
		Attack-4	28	66(38)	58.7(30.7)	14(−14)	88(60)	12(−16)	92(64)	**4(−24)**	44(16)	-	76.2(48.2)	32(4)	84(56)
	Adadelta	Attack-1	51.8	49.1(−2.7)	47.2(−4.6)	48.8(−3)	60.5(8.7)	**39.2(−12.6)**	59.2(7.4)	60(8.2)	39.4(−12.4)	-	49.7(−2.1)	43.3(−8.5)	39.5(−12.3)
		Attack-2	50.7	47.1(−3.6)	45.3(−5.4)	50.3(−0.4)	59.6(8.9)	37.4(−13.3)	58.1(7.4)	58.2(7.5)	37.6(−13.1)	-	45.6(−5.1)	42.1(−8.6)	**35.8(−14.9)**
		Attack-3	49.6	46.8(−2.8)	44.4(−5.2)	49.5(−0.1)	55.3(5.7)	35.5(−14.1)	57.6(8)	57.3(7.7)	35.9(−13.7)	-	43.8(−5.8)	39.6(−10)	**34.5(−15.1)**
		Attack-4	92	88(−4)	78.3(−13.7)	92(0)	80(−12)	16(−76)	84(−8)	12(−80)	20(−72)	-	7(−85)	66.3(−25.7)	**39.9(−52.1)**

Table 5. *Cont.*

Datasets	Optimizers	Attacks	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
CIFAR-10	SGD	Attack-1	92.6	82.8(−9.8)	80.5(−12.1)	91.7(−0.9)	91.5(−1.1)	89.4(−3.2)	85.3(−7.3)	90.4(−2.2)	81.3(−11.3)	84.2(−8.4)	67.3(−25.3)	**58.6(−34)**	66.3(−26.3)
		Attack-2	90.4	80.2(−10.2)	79.9(−10.5)	89.2(−1.2)	89.6(−0.8)	84.2(−6.2)	83.7(−6.7)	86.9(−3.5)	79.4(−11)	81.8(−8.6)	63.2(−27.2)	**55.4(−35)**	64.8(−25.6)
		Attack-3	84.7	77.9(−6.8)	75.1(−9.6)	82.8(−1.9)	82.6(−2.1)	81.7(−3)	82.3(−2.4)	82.8(−1.9)	75.2(−9.5)	76.4(−8.3)	62.1(−22.6)	**53.7(−31)**	60.1(−24.6)
		Attack-4	78.4	36.8(−41.6)	35.6(−42.8)	42.7(−35.7)	40.9(−37.5)	40.5(−37.9)	39.1(−39.3)	40.6(−37.8)	33.9(−44.5)	35.2(−43.2)	**25.6(−52.8)**	38.4(−40)	26(−52.4)
	Adagrad	Attack-1	91.3	76.8(−14.5)	75.7(−15.6)	90.2(−1.1)	89.4(−1.9)	90.4(−0.9)	87.6(−3.7)	88.1(−3.3)	74.2(−17.1)	-	70.2(−21.1)	73.8	**54.8(−36.5)**
		Attack-2	89.4	73.6(−15.8)	72.4(−17)	88.3(−1.1)	87.9(−1.5)	85.3(−4.1)	84.1(−5.3)	83.8(−5.6)	72.8(−16.6)	-	68.3(−21.1)	69.7(−19.7)	**51.9(−37.5)**
		Attack-3	83.6	68.9(−14.7)	65.7(−17.9)	83.1(−0.5)	80.7(−2.9)	81.5(−2.1)	81.6(−2)	82.4(−1.2)	66.9(−16.7)	-	65.4(−18.2)	68.3(−15.3)	**50.1(−33.5)**
		Attack-4	72.5	28.6(−43.9)	25.3(−47.2)	36.4(−36.1)	34.9(−37.6)	37.7(−34.8)	30.3(−42.2)	33.4(−39.1)	25.9(−46.6)	-	71.3(−1.2)	53.9(−18.6)	**24(−48.5)**
	RMSProp	Attack-1	89.3	83.6(−5.7)	82.9(−6.4)	88.4(−0.9)	87.3(−2)	88(−1.3)	87.6(−1.7)	89.1(−0.2)	82.5(−6.8)	-	**56.8(−32.5)**	68.2(−21.1)	60.3(−29)
		Attack-2	84.2	78.4(−5.8)	75.9(−8.3)	83.9(−0.3)	82.4(−1.8)	81.9(−2.3)	82.8(−1.4)	82.7(−1.5)	76.8(−7.4)	-	60.9(−23.3)	65.8(−18.4)	**58.9(−25.3)**
		Attack-3	81.6	74.9(−6.7)	72.9(−8.7)	80.6(−1)	78.5(−3.1)	77.3(−4.3)	78.2(−3.4)	80.1(−1.5)	71.4(−10.2)	-	61.2(−20.4)	67.3(−14.3)	**54.6(−27)**
		Attack-4	68.5	24.7(−43.8)	22.6(−45.9)	32.9(−35.6)	30.5(−38)	31.8(−36.7)	29.2(−39.3)	31.3(−37.2)	23.4(−45.1)	-	48.8(−19.7)	60.8(−7.7)	**28.8(−39.7)**
	Nadam	Attack-1	67.2	65.8(−1.4)	64.2(−3)	67.1(−0.1)	60.6(−6.6)	**28.6(−38.6)**	74.3(7.1)	65.3(−1.9)	55.3(−11.9)	-	65.8(−1.4)	55.6(−11.6)	65.1(−2.1)
		Attack-2	65.8	63.3(−2.5)	62.1(−3.7)	64.2(−1.6)	58.3(−7.5)	**25.7(−40.1)**	73.6(7.8)	61.4(−4.4)	51.2(−14.6)	-	62.3(−3.5)	54.3(−11.5)	63.8(−2)
		Attack-3	63.2	61.1(−2.1)	60.7(−2.5)	62.1(−1.1)	57.8(−5.4)	**24.5(−38.7)**	74.8(11.6)	63.1(−0.1)	50.8(−12.4)	-	60.9(−2.3)	52.8(−10.4)	60.9(−2.3)
		Attack-4	68.3	48.3(−20)	45.6(−22.7)	67.5(−0.8)	61.2(−7.1)	29.6(−38.7)	88.5(20.2)	72.1(3.8)	45.3(−23)	-	44.2(−24.1)	58.9(−9.4)	**18(−50.3)**
	Adadelta	Attack-1	65.3	63.5(−1.8)	61.9(−3.4)	65.1(−0.2)	58.3(−7)	55.8(−9.5)	73.5(8.2)	68.3(3)	52.8(−12.5)	-	61.7(−3.6)	55.8(−9.5)	**51.8(−13.5)**
		Attack-2	64.2	61.3(−2.9)	60.2(−4)	64.1(−0.1)	57.6(−6.6)	53.9(−10.3)	73.6(9.4)	65.1(0.9)	53.4(−10.8)	-	60.5(−3.7)	54.9(−9.3)	**49.3(−14.9)**
		Attack-3	63.1	59.8(−3.3)	57.8(−5.3)	62.3(−0.8)	55.8(−7.3)	52.1(−11)	74.5(11.4)	66.3(3.2)	51.1(−12)	-	58.8(−4.3)	53.1(−10)	**45.2(−17.9)**
		Attack-4	58	53(−5)	51.4(−6.6)	38(−20)	35(−23)	48.6(−9.4)	70.8(12.8)	28.8(−29.2)	**24(−34)**	-	48(−10)	32(−26)	40.8(−17.2)
Purchase	SGD	Attack-1	53.3	52.5(−0.8)	51.6(−1.7)	66.9(13.6)	54.2(0.9)	53.5(0.2)	54.2(0.9)	51.1(−2.2)	**41(−12.3)**	51.2(−2.1)	48.7(−4.6)	57(3.7)	52(−1.3)
		Attack-2	52.2	51.2(−1)	50.6(−1.6)	66.1(13.9)	53.8(1.6)	52.5(0.3)	53.3(1.1)	50.1(−2.1)	**40.5(−11.7)**	50.1(−2.1)	47.5(−4.7)	56.5(4.3)	51.8(−0.4)
		Attack-3	51.8	50.1(−1.7)	50.1(−1.7)	65.8(14)	52.2(0.4)	51.2(−0.6)	52.1(0.3)	49.8(−2)	**39.2(−12.6)**	49.9(−1.9)	46.9(−4.9)	55.3(3.5)	50.9(−0.9)
		Attack-4	88	12(−76)	66.8(−21.2)	70.3(−17.7)	32.6(−55.4)	54.3(−33.7)	16(−72)	**12(−76)**	20(−68)	66(−22)	26(−62)	28(−60)	24(−64)
	Adagrad	Attack-1	59.9	52.9(−7)	59.4(−0.5)	73.7(13.8)	53.1(−6.8)	50.1(−9.8)	53.9(−6)	57.9(−2)	**35(−24.9)**	-	52.3(−7.6)	49.6(−10.3)	53.7(−6.2)
		Attack-2	58.4	51.3(−7.1)	58.8(0.4)	73.1(14.7)	52.8(−5.6)	49.4(−9)	53.1(−5.3)	56.7(−1.7)	**32.5(−25.9)**	-	51.9(−6.5)	48.5(−9.9)	52.9(−5.5)
		Attack-3	57.2	50.2(−7)	58.1(0.9)	72.8(15.6)	52(−5.2)	48.9(−8.3)	52.8(−4.4)	55.9(−1.3)	**31.8(−25.4)**	-	51.1(−6.1)	48.9(−8.3)	52.1(−5.1)
		Attack-4	16	16(0)	24(8)	16(0)	28(12)	14(−2)	12(−4)	24(8)	26(10)	-	20(4)	**8(−8)**	25(9)
	RMSProp	Attack-1	46.9	36.4(−10.5)	49.8(2.9)	71.8(24.9)	52.1(5.2)	53.9(7)	71.8(24.9)	48.4(1.5)	26(−20.9)	-	54.9(8)	**25.2(−21.7)**	40.5(−6.4)
		Attack-2	45.3	36.1(−9.2)	48.9(3.6)	70.5(25.2)	51.2(5.9)	52.9(7.6)	70.2(24.9)	47.9(2.6)	25.4(−19.9)	-	53.8(8.5)	**24.7(−20.6)**	39.7(−5.6)
		Attack-3	44.3	35.2(−9.1)	48.1(3.8)	69.9(25.6)	51.1(6.8)	52.1(7.8)	68.9(24.6)	47.1(2.8)	24.9(−19.4)	-	53.1(8.8)	**23.9(−20.4)**	39.1(−5.2)
		Attack-4	92	24(−68)	32(−60)	12(−80)	16(−76)	92(0)	12(−80)	24(−68)	76(−16)	-	20(−72)	**8(−84)**	36(−56)
	Nadam	Attack-1	56.2	40.4(−15.8)	53.4(−2.8)	72.1(15.9)	53.1(−3.1)	51.3(−4.9)	46(−10.2)	55.8(−0.4)	**20.8(−35.4)**	-	51.3(−4.9)	30.2(−26)	51(−5.2)
		Attack-2	55.3	39.8(−15.5)	52.2(−3.1)	71.8(16.5)	52.3(−3)	50.8(−4.5)	45.7(−9.6)	55.1(−0.2)	**19.9(−35.4)**	-	50.9(−4.4)	29.2(−26.1)	50.8(−4.5)
		Attack-3	55.1	39.2(−15.9)	51.3(−3.8)	70.9(15.8)	51.3(−3.8)	49.2(−5.9)	44.3(−10.8)	55.8(0.7)	**18.4(−36.7)**	-	50.1(−5)	28.7(−26.4)	49.9(−5.2)
		Attack-4	84	92(8)	24(−60)	16(−68)	24(−60)	76(−8)	11(−73)	8(−76)	28(−56)	-	**8(−76)**	18(−66)	12(−72)
	Adadelta	Attack-1	52.7	51.1(−1.6)	51.1(−1.6)	51.8(−0.9)	55.3(2.6)	51.9(−0.8)	50(−2.7)	50(−2.7)	**46.8(−5.9)**	-	55.6(2.9)	52(−0.7)	53.7(1)
		Attack-2	51.3	50.2(−1.1)	49.8(−1.5)	50.3(−1)	52.7(1.4)	49.8(−1.5)	48.7(−2.6)	49.2(−2.1)	**45.2(−6.1)**	-	53.2(1.9)	51.2(−0.1)	51.3(0)
		Attack-3	49.8	48.3(−1.5)	49.5(−0.3)	48.2(−1.6)	50.8(1)	49.2(−0.6)	47.6(−2.2)	48.6(−1.2)	**43.2(−6.6)**	-	50.1(0.3)	49.8(0)	50.1(0.3)
		Attack-4	84	**4(−80)**	72(−12)	8(−76)	24(−60)	64(−20)	28(−56)	62(−22)	8(−76)	-	18(−66)	16(−68)	16(−68)

5.2.2. Attack Recall

Reducing the attacks' recall is the best sign that implies MIA mitigation. Figure 13 illustrates the results of the four aforementioned attacks applying five optimizers, with(out) countermeasures on four datasets, MNIST, FMNIST, CIFAR-10, and Purchase, respectively. The y-axis represents the recall of the attack. The attack recall in CL is tabulated in Table 5, whereas the attack recall in FL is tabulated in Table 6 on various datasets and optimizers.

Figure 13. A comparison of the four attacks on the FL environment using five optimizers with and without countermeasures.

Table 6. FL attack recall.

Datasets	Optimizers	Attacks	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
MNIST	SGD	Attack-1	95.2	79.3(−15.9)	82.4(−12.8)	94.5(−0.7)	91.6(−3.6)	93.6(−1.6)	81(−14.2)	93.4(−1.8)	75.7(−19.5)	88(−7.2)	76.9(−18.3)	72.6(−22.6)	**69.8(−25.4)**
		Attack-2	94.5	74.6(−19.9)	82(−12.5)	94.3(−0.2)	91(−3.5)	93.1(−1.4)	78.2(−16.3)	93(−1.5)	72.8(−21.7)	86.7(−7.8)	74.5(−20)	72.5(−22)	**67.2(−27.3)**
		Attack-3	88.2	71.7(−16.5)	68.4(−19.8)	86.2(−2)	82.5(−5.7)	81.1(−7.1)	69.4(−18.8)	85(−3.2)	68.9(−19.3)	81.2(−7)	72.6(−15.6)	71.8(−16.4)	**65.8(−22.4)**
		Attack-4	**86**	24.8(−61.2)	34(−52)	24.4(−61.6)	24.1(−61.9)	34.5(−51.5)	28(−58)	24.3(−61.7)	**18(−68)**	22.1(−63.9)	48.4(−37.6)	40(−46)	35.5(−50.5)
	Adagrad	Attack-1	97.6	97(−0.6)	93.4(−4.2)	95.8(−1.8)	96.4(−1.2)	96.7(−0.9)	93.5(−4.1)	97(−0.6)	73.4(−24.2)	-	77.6(−20)	**73(−24.6)**	77.4(−20.2)
		Attack-2	97.5	89(−8.5)	93.1(−4.4)	94(−3.5)	96.1(−1.4)	96.3(−1.2)	92.1(−5.4)	95(−2.5)	**70.3(−27.2)**	-	75.8(−21.7)	72.3(−25.2)	83.2(−14.3)
		Attack-3	89.2	88.1(−1.1)	74.7(−14.5)	86(−3.2)	86.2(−3)	84.5(−4.7)	79(−10.2)	81.9(−7.3)	**64.2(−25)**	-	73.6(−15.6)	71.2(−18)	81.6(−7.6)
		Attack-4	**82**	**16(−66)**	38(−44)	34.3(47.7)	32(−50)	38.4(−43.6)	20(−62)	16(−66)	17.8(−64.2)	-	**13.9(−68.1)**	28(−54)	24(−58)
	RMSProp	Attack-1	99	98.7(−0.3)	97.4(−1.6)	98.6(−0.4)	92.8(−6.2)	93.5(−5.5)	97(−2)	96.6(−2.4)	**83.6(−15.4)**	-	82.7(−16.3)	**66.5(−32.5)**	69.4(−29.6)
		Attack-2	98.9	98.2(−0.7)	97(−1.9)	98.3(−0.6)	89.4(−9.5)	87.7(−11.3)	91.9(−7)	91.3(−7.6)	78.6(−20.4)	-	75.2(−23.7)	**63.2(−35.7)**	68.7(−30.2)
		Attack-3	93.5	90.8(−2.7)	83.4(−10.1)	92.9(−0.6)	85.3(−8.2)	91(−2.5)	91.7(−1.8)	91.05(−2.4)	72.5(−21)	-	71.1(−22.4)	**61.1(−32.4)**	65.9(−27.6)
		Attack-4	**88**	**70(−18)**	31(−57)	36(−52)	32(−56)	36.2(51.8)	30(−58)	28(−60)	22.6(−65.4)	-	**21.6(−66.4)**	28(−60)	68(−20)
	Nadam	Attack-1	94.5	81.1(−13.4)	79.8(−14.7)	85.1(−9.4)	76.5(−18)	**40.5(−54)**	83.9(−10.6)	62.1(−32.4)	57.4(−37.1)	-	74.2(−20.3)	52.9(−41.6)	70.2(−24.3)
		Attack-2	93.8	79.5(−14.3)	78.2(−15.6)	84.2(−9.6)	75.6(−18.2)	**39.5(−54.3)**	82.3(−11.5)	60.9(−32.9)	56.2(−37.6)	-	72.1(−21.7)	50.7(−43.1)	68.5(−25.3)
		Attack-3	89.5	75.8(−13.7)	68.5(−21)	83.9(−5.6)	73.2(−16.3)	**38.7(−50.8)**	79.8(−9.7)	58.6(−30.9)	55.3(−34.2)	-	70.6(−18.9)	48.2(−41.3)	64.3(−25.2)
		Attack-4	88	49.3(−38.7)	47.5(−40.5)	87.2(−0.8)	80(−8)	16(−72)	56(−32)	52(−36)	54.6(−33.4)	-	33(−55)	**8(−80)**	12(−76)
	Adadelta	Attack-1	65.7	55.1(−10.6)	52.3(−13.4)	83.7(18)	**48.1(−17.6)**	48.3(−17.4)	48.8(−16.9)	66.8(1.1)	70.7(5)	-	58.1(−7.6)	48.3(−17.4)	58.4(−7.3)
		Attack-2	64.3	53.9(−10.4)	50.8(−13.5)	81.8(17.5)	**45.4(−18.9)**	46.7(−17.6)	47.5(−16.8)	65.2(0.9)	68.5(4.2)	-	56.3(−8)	47.6(−16.7)	56.3(−8)
		Attack-3	60.9	50.2(−10.7)	46.7(−14.2)	75.6(14.7)	**41.8(−19.1)**	44.3(−16.6)	45.4(−15.5)	60.8(−0.1)	65.3(4.4)	-	52.3(−8.6)	41.1(−19.8)	52.1(−8.8)
		Attack-4	88	69.3(−18.7)	65.4(−22.6)	**12(−76)**	84(−4)	32(−56)	32(−56)	49(−39)	45.6(−42.4)	-	28(−60)	12(−76)	50.6(−37.4)
FMNIST	SGD	Attack-1	82.4	71(−11.4)	74.6(−7.4)	76.7(−5.7)	80.5(−1.9)	77.5(−4.9)	77(−5.4)	81.9(−0.5)	74.8(−7.6)	75.1(−7.3)	64.7(−17.7)	65.2(−17.2)	**50.1(−32.3)**
		Attack-2	82.1	70.3(−11.8)	69.8(−12.3)	74.4(−7.7)	77.2(−4.9)	79.1(−3)	76.9(−5.2)	81.1(−1)	70.2(−11.9)	71.6(−10.5)	63.2(−18.9)	63.5(−18.6)	**49.2(−32.9)**
		Attack-3	76.8	64.1(−12.7)	63.2(−13.6)	69.8(−7)	71.9(−4.9)	71.1(−5.7)	69.7(−7.1)	68.3(−8.5)	65.6(−11.2)	65.8(−11)	61.3(−15.5)	60.2(−16.6)	**47.9(−28.9)**
		Attack-4	**72**	**9(−63)**	16(−56)	32.8(−39.2)	36.2(−35.8)	44(−28)	36(−36)	32(−40)	26.1(−45.9)	23.8(−48.2)	76.1(4.1)	44(−28)	62.6(−9.4)
	Adagrad	Attack-1	83.6	80.2(−3.4)	78.2(−5.4)	81.1(−2.5)	81(−2.6)	80.6(−2)	78.9(−4.7)	82.5(−1.1)	73.2(−10.4)	-	84.5(0.9)	**62.9(−20.7)**	66.8(−16.8)
		Attack-2	82.2	80(−2.2)	77.5(−4.7)	80.9(−1.3)	81(−1.2)	80.2(−2)	78.1(−4.1)	82(−0.2)	71.2(−11)	-	80.6(−1.6)	**59.8(−22.4)**	64.8(−17.4)
		Attack-3	75.1	73.4(−1.7)	71.3(−3.8)	71.1(−4)	69.9(−5.2)	70(−5.1)	72.1(−3)	74.2(−0.9)	**69.4(−5.7)**	-	78.8(3.7)	**57.3(−17.8)**	63.1(−12)
		Attack-4	80	68(−12)	**24(−56)**	36.4(−43.6)	**24(−56)**	36.8(−43.2)	34(−46)	36(−44)	26.3(−53.7)	-	92(12)	84(4)	59.8(−20.2)
	RMSProp	Attack-1	74.2	69.9(−4.3)	72.3(−1.9)	67.8(−6.4)	66(−8.2)	64.3(−9.9)	67.5(−6.7)	73.6(−0.6)	**63.9(−10.3)**	-	88.1(13.9)	69.4(−4.8)	65.3(−8.9)
		Attack-2	73.9	69.4(−4.5)	71.7(−2.2)	66.7(−7.2)	66.5(−7.4)	63.8(−10.1)	66.1(−7.8)	73(−0.9)	**60.8(−13.1)**	-	83.7(9.8)	67.5(−6.4)	63.7(−10.2)
		Attack-3	68.2	61.3(−6.9)	**55.8(−12.4)**	59(−9.2)	58.3(−9.9)	59.6(−8.6)	59.3(−8.9)	64.6(−3.6)	58.1(−10.1)	-	80.9(12.7)	65.1(−3.1)	62.9(−5.3)
		Attack-4	69	**12(−57)**	16(−53)	38(−31)	14(−55)	32.3(−36.7)	34(−35)	32(−37)	24.7(−44.3)	-	85(16)	76.1(7.1)	41(−28)
	Nadam	Attack-1	82.6	63.9(−18.7)	74.7(−7.9)	81.25(−1.3)	56.9(−25.7)	**32.1(−50.5)**	78.8(−3.8)	80.2(−2.4)	68.4(−14.2)	-	84.2(1.6)	58.8(−23.8)	56.9(−25.7)
		Attack-2	80.7	62.5(−18.2)	73.2(−7.5)	79.4(−1.3)	55.5(−25.2)	**33.2(−47.5)**	75.6(−5.1)	78.6(−2.1)	65.2(−15.5)	-	82.6(1.9)	57.4(−23.3)	55.8(−24.9)
		Attack-3	81.5	58.1(−23.4)	56.8(−24.7)	72.3(−9.2)	52.3(−29.2)	**40.2(−41.3)**	73.1(−8.4)	75.9(−5.6)	54.5(−27)	-	80.9(−0.6)	53.4(−28.1)	54.3(−27.2)
		Attack-4	83.2	79.2(−4)	54(−29.2)	72(−11.2)	80(−3.2)	76(−7.2)	71.6(−11.6)	69.1(−14.1)	**39(−44.2)**	-	76(−7.2)	84(0.8)	60(−23.2)
	Adadelta	Attack-1	68.9	49.4(−19.5)	48.2(−20.7)	61.8(−7.1)	59.4(−9.5)	58.9(−10)	48.5(−20.4)	67.6(−1.3)	51.2(−17.7)	-	44.6(−24.3)	62.9(−6)	**40.3(−28.6)**
		Attack-2	67.8	48.7(−19.1)	47.3(−20.5)	60.2(−7.6)	58.2(−9.6)	56.5(−11.3)	46.8(−21)	67.2(−0.6)	48.5(−19.3)	-	43.1(−24.7)	60.8(−7)	**38.6(−29.2)**
		Attack-3	62.2	47.6(−14.6)	46.5(−15.7)	57.8(−4.4)	59.5(−2.7)	51.2(−11)	42.3(−19.9)	60.8(−1.4)	47.3(−14.9)	-	42.6(−19.6)	62.4(0.2)	**35.4(−26.8)**
		Attack-4	62.8	22(−40.8)	20(−42.8)	55.9(−6.9)	61.6(−1.2)	68.8(6)	62.4(−0.4)	45.3(−17.5)	37.9(−24.9)	-	**16(−46.8)**	84(21.2)	38.2(−24.6)

Table 6. *Cont.*

Datasets	Optimizers	Attacks	WC	D	MCD	BN	GD	AR	GN	M	KD	DP	AR–KD	GN–KD	GD–KD
CIFAR-10	SGD	Attack-1	79	68.5(−10.5)	62.2(−16.8)	78.3(−0.7)	77.8(−1.2)	76.3(−2.7)	75.2(−3.8)	73.6(−5.4)	68.9(−10.1)	69.1(−9.9)	60.2(−18.8)	63.3(−15.7)	**51.2(−27.8)**
		Attack-2	78.6	68.2(−10.4)	61.1(−17.5)	78.1(−0.5)	77.4(−1.2)	74.3(−4.3)	76.7(−1.9)	74.2(−4.4)	63.1(−15.5)	67.6(−11)	58.3(−20.3)	61.7(−16.9)	**48.9(−29.7)**
		Attack-3	74.3	67.4(−6.9)	60.9(−13.4)	73.4(−0.9)	73.2(−1.1)	71.5(−2.8)	71.2(−3.1)	72.8(−1.5)	60.4(−13.9)	69.6(−4.7)	57.4(−16.9)	58.4(−15.9)	**45.3(−29)**
		Attack-4	75.6	31(−44.6)	28(−47.6)	33.9(−41.7)	32.6(−43)	30(−45.6)	29.8(−45.8)	**23.4(−52.2)**	25.8(−49.8)	30.9(−44.7)	48(−27.6)	40(−35.6)	29(−46.6)
	Adagrad	Attack-1	74.2	65(−9.2)	61(−13.2)	73.8(−0.4)	73.1(−1.1)	72.4(−1.8)	73(−1.2)	70(−4.2)	65.8(−8.4)	-	71.4(−2.8)	**54.2(−20)**	58.1(−16.1)
		Attack-2	73.7	64.3(−9.4)	60(−13.7)	73.1(−0.6)	72.6(−1.1)	72.2(−1.5)	72.8(−0.9)	69.5(−4.2)	63.1(−10.6)	-	69.2(−4.5)	**52.4(−21.3)**	56.8(−16.9)
		Attack-3	67.4	56.9(−10.5)	53(−14.4)	62.4(−5)	62.2(−5.2)	60.4(−7)	61.5(−5.9)	58.4(−9)	60.4(−7)	-	67.4(0)	**51.1(−16.3)**	52.4(−15)
		Attack-4	70.1	17(−53.1)	13(−57.1)	28.4(−41.7)	25(−45.1)	27.1(−43)	25.2(−44.9)	**12(−58.1)**	26.2(−43.9)	-	18(−52.1)	32(−38.1)	25(−45.1)
	RMSProp	Attack-1	68.2	58.6(−9.6)	55(−13.2)	65.8(−2.4)	64.2(−4)	65.3(−2.9)	64(−4.2)	62.1(−6.1)	62.3(−5.9)	-	**52.4(−15.8)**	58.7(−9.5)	59.3(−8.9)
		Attack-2	67.9	57.3(−10.6)	53.2(−14.7)	65.1(−2.8)	63.6(−5.3)	61.9(−6)	62.4(−5.5)	60.8(−7.1)	60.6(−7.3)	-	**50.6(−17.3)**	57.1(−10.8)	57.9(−10)
		Attack-3	63.7	56(−7.7)	52.9(−10.8)	62.3(−1.4)	60.8(−2.9)	61.7(−2)	59(−4.7)	59.4(−4.3)	57.2(−6.5)	-	**49.7(−14)**	56.2(−7.5)	55.4(−8.3)
		Attack-4	65.6	23(−42.6)	**12(−53.6)**	34(−31.6)	32.1(−33.5)	32.7(−32.9)	30.3(−35.3)	25.6(−40)	21.9(−43.7)	-	32.4(−33.2)	46.8(−18.8)	35(−30.6)
	Nadam	Attack-1	78.4	68.5(−9.9)	65.8(−12.6)	77.8(−0.6)	53.1(−25.3)	**45.2(−33.2)**	74.3(−4.1)	75.9(−2.5)	67.1(−11.3)	-	74.3(−4.1)	55.6(−22.8)	52.6(−25.8)
		Attack-2	75.2	64.2(−11)	62.3(−12.9)	74.2(−1)	52.8(−22.4)	**43.8(−31.4)**	71.6(−3.6)	74.2(−1)	52.3(−22.9)	-	71.4(−3.8)	52.1(−23.1)	49.4(−25.8)
		Attack-3	73.1	6 4.8(−8.3)	61.7(−11.4)	72.1(−1)	50.3(−22.8)	**41.6(−31.5)**	69.2(−3.9)	72.6(−0.5)	50.9(−22.2)	-	69.8(−3.3)	51.3(−21.8)	48.7(−24.4)
		Attack-4	70.6	65.2(−5.4)	32(−38.6)	70.1(−0.5)	65.4(−5.2)	50.3(−20.3)	60.5(−10.1)	55.6(−15)	**31.2(−39.4)**	-	60.3(−10.3)	74.6(4)	49.1(−21.5)
	Adadelta	Attack-1	65.9	55.3(−10.6)	53.1(−12.8)	65.1(−0.8)	58.5(−7.4)	52.6(−13.3)	45.2(−20.7)	62.1(−3.8)	50.3(−15.6)	-	**35.3(−30.6)**	55.8(−10.1)	43.7(−22.2)
		Attack-2	64.5	52.1(−12.4)	51.2(−13.3)	63.1(−1.4)	56.2(−8.3)	49.5(−15)	43.8(−20.7)	59.8(−4.7)	48.7(−15.8)	-	**33.7(−30.8)**	52.9(−11.6)	42.9(−21.6)
		Attack-3	62.3	50.6(−11.7)	48.5(−13.8)	61.1(−1.2)	53.1(−9.2)	47.6(−14.7)	41.9(−20.4)	57.6(−4.7)	45.4(−16.9)	-	**32.2(−30.1)**	51.6(−10.7)	40.3(−22)
		Attack-4	58	22(−36)	20(−38)	50(−8)	50.3(−7.7)	60(2)	56.2(−1.8)	45.7(−12.3)	18(−40)	-	**18.6(−39.4)**	32.4(−25.6)	36.9(−21.1)
Purchase	SGD	Attack-1	52.2	53.1(1.9)	49.3(−1.9)	52.2(1)	52.9(1.7)	54.5(3.3)	51.6(0.4)	50.1(−1.1)	51.9(0.7)	49.5(−1.7)	48.9(−2.3)	50.9(−0.3)	**48.4(−2.8)**
		Attack-2	50.2	52.2(2)	48.9(−1.3)	51.8(1.6)	52.1(1.9)	53.7(3.5)	50.8(0.6)	49.5(−0.7)	51.1(0.9)	49.1(−1.1)	48.1(−2.1)	50.6(0.4)	**47.9(−2.3)**
		Attack-3	49.3	51.6(2.3)	48.1(−1.2)	51.1(1.8)	51.5(2.2)	52.4(3.1)	50.1(0.8)	49(−0.3)	50.8(1.5)	48.7(−0.6)	47.8(−1.5)	49.9(0.6)	**47.1(−2.2)**
		Attack-4	84	24(−60)	24(−60)	92(8)	22(−62)	80(−4)	12(−72)	36(−48)	**8(−76)**	80(−4)	84(0)	88(4)	20(−64)
	Adagrad	Attack-1	52.4	53.3(0.9)	53.1(0.7)	58.7(6.3)	51.1(−1.3)	55.2(2.8)	53.4(1)	55.3(2.9)	54.9(2.5)	-	54(1.6)	**47.9(−4.5)**	49.1(−3.3)
		Attack-2	51.2	52.8(1.6)	52.8(1.6)	57.8(6.6)	50.8(−0.4)	54.9(3.7)	53.1(1.9)	54.8(3.6)	54.2(3)	-	53.5(2.3)	**47(−4.2)**	48.8(−2.4)
		Attack-3	50.8	52.1(1.3)	52.1(1.3)	57(6.2)	50.2(−0.6)	54.2(3.4)	52.8(2)	54.1(3.3)	53.7(2.9)	-	53.1(2.3)	**46.5(−4.3)**	48.1(−2.7)
		Attack-4	72	80(8)	88(16)	12(−60)	28(−44)	24(−48)	84(12)	24(−48)	80(8)	-	**12(−60)**	84(12)	92(20)
	RMSProp	Attack-1	37.8	42.3(4.5)	36(−1.8)	**23.1(−14.7)**	51.8(14)	52(14.2)	35.2(−2.6)	39.8(2)	55.8(18)	-	54.7(16.9)	43.4(5.6)	42.2(4.4)
		Attack-2	37.1	41.9(4.8)	35.7(−1.4)	**22.8(−14.3)**	51.2(14.1)	51.6(14.5)	34.8(−2.3)	39.1(2)	55.1(18)	-	53.9(16.8)	42.8(5.7)	41.8(4.7)
		Attack-3	36.8	41(4.2)	35(−1.8)	**22.5(−14.3)**	49.7(12.9)	51(14.2)	34.2(−2.6)	38.7(1.9)	54.9(18.1)	-	53.1(16.3)	42.2(5.4)	41.2(4.4)
		Attack-4	96	88(−8)	36(−60)	36(−60)	24(−72)	32(−64)	36(−60)	24(−72)	96(0)	-	**16(−80)**	20(−76)	24(−72)
	Nadam	Attack-1	35.8	40.4(4.6)	35.1(−0.7)	**19.2(−16.6)**	54.4(18.6)	50.5(14.7)	29.6(−6.2)	34.4(−1.4)	51.4(15.6)	-	45.3(9.5)	46.1(10.3)	50.9(15.1)
		Attack-2	34.7	39.8(5.1)	34.7(0)	**18.9(−15.8)**	53.9(19.2)	49.8(15.1)	28.9(−5.8)	34(−0.7)	50.7(16)	-	44.8(10.1)	45.8(11.1)	50.1(15.4)
		Attack-3	34.1	39(4.9)	34(−0.1)	**18.3(−15.8)**	53.1(19)	49.1(15)	28.3(−5.8)	33.8(−0.3)	50(15.9)	-	44.3(10.2)	45.3(11.2)	49.8(15.7)
		Attack-4	40	28(−12)	72(32)	20(−20)	88(48)	80(40)	88(48)	**8(−32)**	72(32)	-	12(−28)	84(44)	16(−24)
	Adadelta	Attack-1	51.1	50.5(−0.6)	53.1(2)	54.7(3.6)	55.1(4)	53.1(2)	52.8(1.7)	53.9(2.8)	59.4(8.3)	-	**46.4(−4.7)**	50.5(−0.6)	49.3(−1.8)
		Attack-2	50.4	49.5(−0.9)	52.9(2.5)	54.1(3.7)	54.8(4.4)	52.8(2.4)	51.9(1.5)	53.2(2.8)	58.7(8.3)	-	**45.9(−4.5)**	50(−0.4)	48.6(−1.8)
		Attack-3	50	49.1(−0.9)	52.1(2.1)	53.3(3.3)	54.3(4.3)	52.1(2.1)	51.1(1.1)	52.8(2.8)	58(8)	-	**45.3(−4.7)**	49.8(−0.2)	48.1(−1.9)
		Attack-4	72	28(−44)	96(24)	24(−48)	96(24)	64(−8)	32(−40)	68(−4)	**4(−68)**	-	8(−64)	12(−60)	24(−48)

- **CL attacks recall without countermeasure:** As shown in Table 5, for the MNIST dataset, the strongest attack is Attack 1 when we apply RMSProp. The recall value of this attack without any countermeasure is 99.4%, which is the highest among other attacks. For Attack 1, only changing the optimizer to Adadelta drops this value to 59.7% without using any countermeasure. Also, the weakest attack goes for Attack 4 when using Adadelta optimization. The recall value of this attack is 44%. For the FMNIST dataset, the strongest attack is Attack 1 with the SGD optimizer and the weakest attack is Attack 4 with the Nadam optimizer. For CIFAR-10, the strongest attack is Attack 1 with SGD optimizer and the weakest is Attack 4 with Adadelta optimizer. For the Purchase dataset, the best attack is Attack 4 with RMSProp optimizer and the worst attack is Attack 4 with Adagrad optimizer.
- **CL attacks recall with countermeasures**: As per Table 5, different mitigation techniques provide various recall values in every attack. We observe that the strongest attack in the case of MNIST, which is Attack 1 with RMSProp, is defended by GD–KD by a reduction of 65.2% of recall value, which is impressive. Using GD–KD is only reducing the model accuracy by 8% according to Table 3. We can conclude that, in the CL environment, GD–KD provides the strongest defense with the lowest model accuracy degradation. This is very important in developing future ML models. For the FMNIST dataset, the strongest attack belongs to Attack 1 when using SGD. This attack in the case of FMNIST is also defended by GD–KD by a reduction of 37.4% in recall value, although the strongest defense for this particular attack and dataset is GN–KD with a 37.9% recall reduction. It is noteworthy that GD–KD and GN–KD drop model accuracy by 15.4% and 13.9%, respectively, as shown in Table 3. The same thing holds true for the CIFAR-10 dataset. The strongest attack is Attack 1 with SGD optimizer for this dataset, and GN–KD is capable of defending this attack by a reduction in attack recall by 34%. Also, in the Purchase dataset, the strongest attack, which is Attack 4 with RMSProp optimizer, is defended by GN–KD and resulted in recall value reduction by 84%. In general, we observe that, in the CL environment, in most of the experiments, the combinations of KD and another countermeasure provides lower attack recall values than other mitigation techniques. This means that these combinations are the best to defend MIA against ML in the CL environment.
- **FL attacks recall without countermeasure**: As shown in Figure 13 and Table 6, for the MNIST dataset, we observe that the highest attack recall (99%) belongs to Attack 1 with RMSProp. This value is significantly reduced to 65.7% by only changing the optimizer to Adadelta. It is impressive to see that changing the optimizer to Adadelta will not drop model accuracy significantly. According to Table 4, using Adadelta reduces FL model accuracy by approximately 1% compared to Nadam. For the FMNIST dataset, Attack 1 with Adagrad provides the highest attack recall value (83.6%). When we change the optimizer to Adadelta, we witness a drop in attack recall to 68.9% without any mitigation technique. The same as Adadelta in MNIST, we are seeing a slight drop in accuracy from 91.7% to 84.1% according to Table 4. For the CIFAR-10 dataset, the highest attack recall is 79% for Attack 1 with SGD optimizer. This value is dropped to 65.9% by only changing the optimizer to Adadelta. Similar to MNIST and FMNIST, this change has not had a significant impact on the accuracy of the FL model. As shown in Table 4, the accuracy of CIFAR-10, when using Adadelta as an optimizer, only drops by roughly 2%. For the Purchase dataset, the best attack is Attack 4 with the RMSProp optimizer with 96% recall value. Also, without applying any countermeasure, the lowest recall value for this dataset belongs to Attack 3 with the Nadam optimizer.
- **FL attacks recall with countermeasures**: As shown in Table 6, it is evident that the various mitigation techniques exhibit varying performance. However, in general, the combinations of KD with either GD, GN, or AR consistently offer improved protection while preserving the model's utility. For MNIST with RMSProp, GN–KD effectively reduces the recall of Attack 1 by 32.5%, which is the most potent attack in our FL MNIST experiments. Remarkably, this reduction is achieved with only an 11%

decrease in FL model accuracy, as indicated in Table 4. In the case of FMNIST, Table 6 reveals that Attack 1 with Adagrad exhibits a high recall value of 83.6%. However, this attack can be mitigated by GN–KD, resulting in a 20.7% reduction in recall. It is worth noting that this defense strategy incurs a modest accuracy drop of 9.7%, as reflected in Table 4. In CIFAR-10, the strongest attack is Attack 1 with SGD, boasting a recall value of 92.6%. GN–KD is capable of reducing this recall to 58.6% while causing a minimal 4.2% drop in FL accuracy, as detailed in Table 4. In the Purchase dataset, the most potent attack, Attack 4, using the RMSProp optimizer, experiences an 80% reduction in effectiveness with a recall value of 96% when AR–KD is applied. Notably, AR–KD not only avoids a decline in accuracy for the Purchase dataset with the RMSProp optimizer but also substantially boosts accuracy by 52%. This improvement is attributed to the capacity of AR–KD to modify the model's architecture, thereby averting overfitting.

5.2.3. Accuracy–Recall Trade-Off

To obtain a clear comparison between the efficiency of the countermeasures, we calculated the ratio of accuracy over recall. The higher the ratio is, the better the trade-off we are achieving. Figure 14 illustrates the accuracy–recall ratio of each countermeasure. As shown in Figure 14, for almost all optimizers, the highest trade-off belongs to one of the combinatory approaches (either AR–KD, GN–KD, or GD–KD). This figure proves that the combinational approaches that we tested provide a better trade-off between the accuracy of the target model and MIA attack recall. The higher value of this trade-off conveys the message that the mitigation technique keeps the accuracy of the target model high and reduces the attack recall as much as possible.

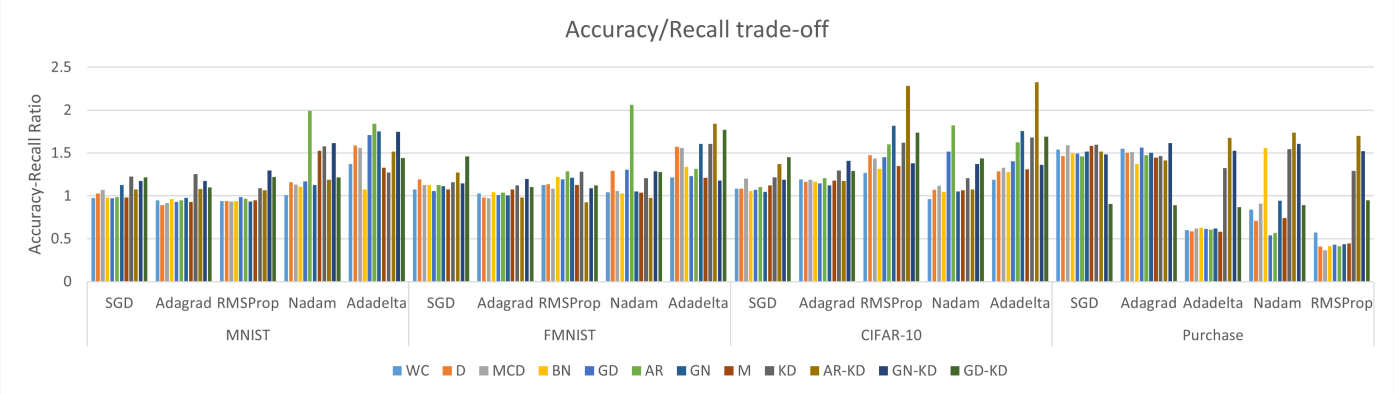

Figure 14. The ratio of the accuracy of the model over recall of the attack model in FL environment.

5.2.4. Privacy and Utility

Concluding from Tables 3–6, it is noted that combination of KD with either AR, GN, or GD has significant advantages over using each one of them separately as well as over other conventional countermeasures. Experiments are showing that the new combinations of countermeasures successfully handle the trade-off between privacy and utility. Generally speaking, in all datasets and almost all optimizers (AR, GD, and GN), KD is capable of reducing the attack recall while preserving the accuracy of the model at a high level. Not only do they preserve the utility of the model at a high level but also, due to the nature of KD, in some cases, they increase model accuracy as well.

6. Conclusions

This research paper presents a thorough examination of the accuracy of centralized and federated learning models, as well as the recall rates associated with different membership inference attacks. Additionally, it evaluates the effectiveness of various defense mechanisms within both centralized and federated learning environments. Our experimental findings reveal that Attack 1 [9] yields the highest advantage for potential attackers,

while Attack 4 [11] is the least favorable for malicious actors. Among the defense strategies examined, the combination of knowledge distillation (KD) with activity regularization (AR), Gaussian dropout (GD), or Gaussian noise (GN) emerges as the most effective in the context of centralized and federated learning. Notably, these three combinations stand out for their ability to effectively balance the trade-off between preserving privacy and maintaining utility. This comparative analysis holds significant importance for guiding future advancements in model development.

Author Contributions: Conceptualization, A.A.T. and D.A.; methodology, A.A.T., S.D. and D.A.; software, A.A.T., S.D. and N.M.; validation, A.A.T., D.A. and N.M.; formal analysis, A.A.T.; investigation, A.A.T. and N.M.; resources, D.A.; data curation, A.A.T.; writing—original draft preparation, A.A.T., S.D. and D.A.; writing—review and editing, A.A.T. and D.A.; visualization, A.A.T.; supervision, D.A.; project administration, D.A.; funding acquisition, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (RGPIN-2019-05689).

Data Availability Statement: The codes and data are available at https://github.com/University-of-Windsor/ComparitiveAnalysis.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Commun. Mag.* **2020**, *58*, 46–51. [CrossRef]
2. Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; Tramer, F. Membership inference attacks from first principles. In Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022; pp. 1897–1914.
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
4. Regulation, P. General data protection regulation. *Intouch* **2018**, *25*, 1–5.
5. Act, A. Health insurance portability and accountability act of 1996. *Public Law* **1996**, *104*, 191.
6. Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; Song, D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In Proceedings of the USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; Volume 267.
7. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706.
8. Backes, M.; Berrang, P.; Humbert, M.; Manoharan, P. Membership privacy in MicroRNA-based studies. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 319–330.
9. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
10. Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; Backes, M. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv* **2018**, arXiv:1806.01246.
11. Liu, L.; Wang, Y.; Liu, G.; Peng, K.; Wang, C. Membership Inference Attacks Against Machine Learning Models via Prediction Sensitivity. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 2341–2347. [CrossRef]
12. Dayal, S.; Alhadidi, D.; Abbasi Tadi, A.; Mohammed, N. Comparative Analysis of Membership Inference Attacks in Federated Learning. In Proceedings of the 27th International Database Engineered Applications Symposium, Heraklion, Greece, 5–7 May 2023; pp. 185–192.
13. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
14. Bjorck, N.; Gomes, C.P.; Selman, B.; Weinberger, K.Q. Understanding batch normalization. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018), Montreal, QC, Canada, 8 December 2018; pp. 31–40.
15. Xiao, Y.; Yan, C.; Lyu, S.; Pei, Q.; Liu, X.; Zhang, N.; Dong, M. Defed: An Edge Feature Enhanced Image Denoised Networks Against Adversarial Attacks for Secure Internet-of-Things. *IEEE Internet Things J.* **2022**, *10*, 6836–6848. [CrossRef]
16. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
17. Keras Documentation: Masking Layer. Available online: https://keras.io/api/layers/core_layers/masking/ (accessed on 29 September 2023).

18. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
19. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4133–4141.
20. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010: 19th International Conference on Computational Statistics, Paris, France, 22–27 August 2010; pp. 177–186.
21. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
22. McMahan, H.B.; Streeter, M. Adaptive bound optimization for online convex optimization. *arXiv* **2010**, arXiv:1002.4908.
23. Poggiolini, P. The GN model of non-linear propagation in uncompensated coherent optical systems. *J. Light. Technol.* **2012**, *30*, 3857–3879. [CrossRef]
24. Keras Documentation: Activityregularization Layer. Available online: https://keras.io/api/layers/regularization_layers/activity_regularization/ (accessed on 29 September 2023).
25. Dozat, T. Incorporating Nesterov Momentum into Adam. Available online: https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ (accessed on 29 September 2023).
26. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
27. Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
28. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
29. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
30. Datasets. Available online: https://www.comp.nus.edu.sg/~reza/files/datasets.html (accessed on 29 September 2023).
31. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 739–753.
32. Conti, M.; Li, J.; Picek, S.; Xu, J. Label-Only Membership Inference Attack against Node-Level Graph Neural Networks. In Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, Los Angeles, CA, USA, 11 November 2022; pp. 1–12.
33. Zheng, J.; Cao, Y.; Wang, H. Resisting membership inference attacks through knowledge distillation. *Neurocomputing* **2021**, *452*, 114–126. [CrossRef]
34. Shejwalkar, V.; Houmansadr, A. Membership privacy for machine learning models through knowledge transfer. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 9549–9557.
35. Lee, H.; Kim, J.; Ahn, S.; Hussain, R.; Cho, S.; Son, J. Digestive neural networks: A novel defense strategy against inference attacks in federated learning. *Comput. Secur.* **2021**, *109*, 102378. [CrossRef]
36. Su, T.; Wang, M.; Wang, Z. Federated Regularization Learning: An Accurate and Safe Method for Federated Learning. In Proceedings of the 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), Washington, DC, USA, 6–9 June 2021; pp. 1–4.
37. Xie, Y.; Chen, B.; Zhang, J.; Wu, D. Defending against Membership Inference Attacks in Federated learning via Adversarial Example. In Proceedings of the 2021 17th International Conference on Mobility, Sensing and Networking (MSN), Exeter, UK, 13–15 December 2021; pp. 153–160.
38. Firdaus, M.; Larasati, H.T.; Rhee, K.H. A Secure Federated Learning Framework using Blockchain and Differential Privacy. In Proceedings of the 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom), Xi'an, China, 25–27 June 2022; pp. 18–23.
39. Bai, Y.; Fan, M. A method to improve the privacy and security for federated learning. In Proceedings of the 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Las Vegas, CA, USA, 4–6 October 2021; pp. 704–708.
40. Chen, H.; Li, H.; Dong, G.; Hao, M.; Xu, G.; Huang, X.; Liu, Z. Practical membership inference attack against collaborative inference in industrial IoT. *IEEE Trans. Ind. Infor.* **2020**, *18*, 477–487. [CrossRef]
41. Novak, R.; Bahri, Y.; Abolafia, D.A.; Pennington, J.; Sohl-Dickstein, J. Sensitivity and generalization in neural networks: An empirical study. *arXiv* **2018**, arXiv:1802.08760.
42. Milanés-Hermosilla, D.; Trujillo Codorniú, R.; López-Baracaldo, R.; Sagaró-Zamora, R.; Delisle-Rodriguez, D.; Villarejo-Mayor, J.J.; Núñez-Álvarez, J.R. Monte Carlo Dropout for Uncertainty Estimation and Motor Imagery Classification. *Sensors* **2021**, *21*, 7241. [CrossRef]
43. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Theory of Cryptography Conference, New York, NY, USA, 4–7 March 2006; pp. 265–284.
44. Dwork, C. A firm foundation for private data analysis. *Commun. ACM* **2011**, *54*, 86–95. [CrossRef]
45. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]

46. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
47. Wu, C.; Wu, F.; Lyu, L.; Huang, Y.; Xie, X. Communication-efficient federated learning via knowledge distillation. *Nat. Commun.* **2022**, *13*, 2032. [CrossRef]
48. Jiang, D.; Shan, C.; Zhang, Z. Federated learning algorithm based on knowledge distillation. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 23–25 October 2020; pp. 163–167.
49. Li, X.; Chen, B.; Lu, W. FedDKD: Federated learning with decentralized knowledge distillation. *Appl. Intell.* **2023**, *53*, 18547–18563. [CrossRef]
50. Available online: https://github.com/University-of-Windsor/ComparitiveAnalysis (accessed on 29 September 2023).
51. Yuan, X.; Zhang, L. Membership Inference Attacks and Defenses in Neural Network Pruning. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022.
52. Asad, M.; Moustafa, A.; Ito, T. Federated learning versus classical machine learning: A convergence comparison. *arXiv* **2021**, arXiv:2107.10976.
53. Peng, S.; Yang, Y.; Mao, M.; Park, D.S. Centralized Machine Learning Versus Federated Averaging: A Comparison using MNIST Dataset. *KSII Trans. Internet Inf. Syst. (TIIS)* **2022**, *16*, 742–756.
54. Drainakis, G.; Katsaros, K.V.; Pantazopoulos, P.; Sourlas, V.; Amditis, A. Federated vs. centralized machine learning under privacy-elastic users: A comparative analysis. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–8.