Multi-Generator MD-GAN with Reset Discriminator and Privacy Analysis

by

Bharat Jain

2020CN-02

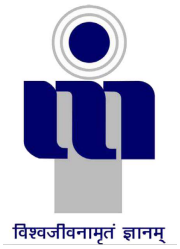
A thesis submitted in partial fulfilment of the requirements for the award of the degree of

Master of Technology

in

Computer Networks

2020-22



ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Thesis Certificate

I hereby certify that the work, which is being presented in the report/thesis, entitled Multi-Generator MD-GAN with Reset Discriminator
and Privacy Analysis, in fulfillment of the requirement for the award of the degree of **Master of Technology**
in **Computer Networks** and submitted to the institution is an authentic record of my/our own work carried out during the period *May-2015* to *June-2017* under the supervision of Prof./Dr. W. Wilfred Godfrey . I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

_____

Prof./Dr. W. Wilfred Godfrey

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the prop er format as specified in the guideline for thesis preparation.
I declare that the work containing in this rep ort is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name: Bharat Jain

Roll. No: 2020CN-02

Date:

Abstract

Federated Learning is a paradigm of machine learning which allows the training of machine learning models in a collaborative fashion by many clients or users, keeping the user or client data decentralized, thus providing data security and data privacy. It focuses on optimal utilization of resources to train a machine learning model without accessing the user's private data. There are many challenges in federated learning, including training models on non-iid data (non-independent identically distributed) and privacy leakage in federated models. In this report, we introduce a novel federated architecture to tackle these two significant challenges of federated learning, i.e., 1) To train a federated model with non-iid client image data and 2) To analyze privacy leakage of the federated model. Our approach uses a Multi-Discriminator GAN with multiple generators. We have label-specific generators at the server, and each client consists of a single discriminator with a reset property. We introduce a novel architecture named Multi-Generator GAN with Reset Discriminator (MG-GAN-RD) that learns the joint distribution of the non-iid client image dataset and trains a global classifier without accessing the client's private dataset. We also perform a membership inference attack on our model to analyze the privacy leakage in its inherent architecture.

Keywords - Federated Learning, Membership Inference Attack, Multi-Discriminator GAN, non-iid, data privacy

Dedication

Acknowledgments

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Asst. Prof. (Dr.) W. Wilfred Godfrey, Department of Computer Science, Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior, for their valuable guidance, encouragement, and help for completing this work. Their useful suggestions and co-operative behaviour are sincerely acknowledged. I also wish to express my indebtedness to my parents whose blessings and support always helped me to face the challenges ahead.

Bharat Jain

(2020CN-02)

Contents

Chapter

1 Introduction **1**

1.1 Context 1

1.2 Problem/Motivation 2

1.3 Objectives 3

1.4 Research work flow 4

2 Literature review **5**

2.1 Background 5

2.1.1 Federated Learning 5

2.1.2 Non-Identically Independently distributed Data Distribution 5

2.1.3 Generative Adversarial Networks 5

2.1.4 Membership Inference Attacks 5

2.2 Key related research 5

2.3 Analysis 7

2.4 Research gaps 7

2.5 Problem formulation 7

2.6 Conclusion 7

3 Methodology **8**

3.1 Proposed hypothesis 8

3.2 Mechanism/Algorithm . . . 9
3.2.1 Model Description . . . 9
3.2.2 Training Local Client Classifier and a Standard Classifier . . . 11
3.2.3 Training Global Generator Pool . . . 12
3.2.4 Training Local Discriminator . . . 12
3.2.5 Training the Global Classifier . . . 13
3.2.6 Performing GAN enhanced Membership Inference Attack . . . 14
3.2.7 Evaluation Metrics . . . 16
3.3 Analytical validation . . . 17
3.4 Conclusion . . . 20

4 Experiments and results **21**

4.1 Experiment design . . . 21
4.2 Training Local Classifier and Standard Classifier . . . 21
4.2.1 Parameter settings . . . 21
4.2.2 Experiment description . . . 21
4.2.3 Results . . . 22
4.3 Training MG-GAN-RD based Global Classifier . . . 23
4.3.1 Parameter settings . . . 24
4.3.2 Experiment description . . . 24
4.3.3 Results . . . 24
4.4 Training MD-GAN based Global Classifier . . . 25
4.4.1 Parameter settings . . . 25
4.4.2 Experiment description . . . 25
4.4.3 Results . . . 25
4.5 Training Global Classifier using FedAvg . . . 27
4.5.1 Parameter settings . . . 27

4.5.2 Experiment description . . . 27
4.5.3 Results . . . 27
4.6 Membership Inference attack on MD-GAN Model . . . 27
4.6.1 Parameter settings . . . 27
4.6.2 Experiment description . . . 27
4.6.3 Results . . . 27
4.7 GAN enhanced Membership Inference attack on MG-GAN-RD Model . . . 28
4.7.1 Parameter settings . . . 28
4.7.2 Experiment description . . . 28
4.7.3 Results . . . 28
4.8 Overall conclusion . . . 28

5 Discussions and conclusion **29**
5.1 Contributions . . . 29
5.2 Limitations . . . 30
5.3 Future scope . . . 30

Bibliography **31**

Appendix

A Training of MD-GAN based Classifier **33**
A.1 Training the Global Classifier . . . 33
A.2 Privacy Leakage . . . 34

B FedAvg **35**
B.1 Training a FedAvg Model . . . 35

Tables

Table

3.1 CNN Architecture 10

3.2 Generator Architecture 11

3.3 Discriminator Architecture 11

4.1 Evaluation of Standard Classifier 22

4.2 Evaluation of Local Classifier with Number of clients = 3 23

4.3 Evaluation of Local Classifier with Number of clients = 10 23

4.4 Evaluation of MG-GAN-RD Based Classifier 24

4.5 Evaluation of Quality of Images 25

4.6 Evaluation of MD-GAN Based Classifier 25

4.7 Evaluation of Quality of Images 27

4.8 Evaluation of FedAvg Classifier 27

4.9 Evaluation of Attack Classifier 27

4.10 Evaluation of Attack Classifier 28

Figures

Figure

3.1 MG-GAN-RD Architecture 9
3.2 Generator Pool at the Server. 13
3.3 Discriminator at Client. 14
3.4 Generating Synthetic Data for MIA. 16
3.5 Membership Inference Attack. 16
3.6 Heatmap of SSIM values for label-wise MNIST data. 18
3.7 Generating Synthetic Data for MIA. 19
3.8 Membership Inference Attack. 19

Chapter 1

Introduction

This chapter presents an overview about the project Multi-Generator MD-GAN with Reset Discriminator (MG-GAN-RD). The context of the project is described in section 1.1. The problems and motivations are described in section 1.2. In section 1.3 the research objectives are defined. Finally, in section 1.4, a step by step research workflow is described.

1.1 Context

McMahan et. al. [14] introduced a novel machine learning concept in 2016. The concept of federated learning was to train a global machine learning model using the information from the shared models rather than accessing the user's private data. This decentralized approach of aggregating local updates from the shared models is called Federated Learning.

Federated learning eliminates the need for data transfer to the server for training a machine learning model, which makes it a highly attractive approach in IoT, where the number of participating devices or users is huge, and a large amount of data is generated. It is not feasible to dump all the data in one place, so the federated approach solves this problem.

Since user data is not required, the federated learning approach provides differential privacy to the user.

Despite many advantages, there are some significant challenges in federated learning like handling non-iid datasets, privacy leakage, cyber-attacks, and issues related to the robustness and effectiveness of the model. This also means that there is much scope for improvement. Many research papers have been published tackling various issues, be it external or systemic, to make the federated model more robust, safe, and accurate.

1.2 Problem/Motivation

Federated Learning : Federated learning is a machine learning paradigm in which multiple devices can participate in training a global machine learning model by sharing their local model updates to the server, which aggregates model updates based on a federated scheme without sharing their private data. Thus, federated learning not only reduces the need to transfer large amount of data to the server but also provides differential privacy to the participating devices. Despite a lot of advantages, federated learning has a lot of challenges that need to be tackled. This includes handling non-iid data, make communication more efficient and less costly, increasing the robustness and effectiveness of the federated model, prevent privacy leakage and provide robust data security, formulating more efficient federated schemes.

Non-Independent and Identically Distributed Data : Non-iid dataset refers to a dataset or part of the original dataset which is either not independent or not identical to its original dataset. The non-iid subset is not able to represent its original dataset distribution. Non-iid dataset can cause serious deterioration of the federated model. A lot of research work has been spent on coming up with techniques or new federated schemes to deal with non-iid data. Non-iid data can be of many types for eg. :

(1) Label skewed non-iid data - Each subset contains data of a subset of the labels and does

contain any data or small amount of data about the other labels.

(2) Attribute skewed non-iid data - It means the feature distribution across the attributes is different for each client. for eg. one client has an image , then the other will have the image from same distribution but rotated with some angle.

(3) Temporal skew non-iid data - temporal data or spatio-temporal data can have skewed distribution to create a non-iid dataset.

For the scope of our thesis, we deal with label skewed non-iid data.

Privacy Leakage : Federated Learning provides differential privacy to the clients. Since the data is not accessed by the server, the client's data is relatively protected. But despite that, there can some serious privacy leakage in the whole architecture. One such attack that can cause a privacy leakage in a federated model is membership inference attack. The idea of membership inference attack is to learn from the training of the federated model itself and find out some information about the training set, the model used to train. So given a data point, a membership inference attack can tell whether that data point was part of the training set of the federated model.

1.3 Objectives

With respect to the problems defined in the previous section. We define two objectives for our thesis.

Formulate a novel federated architecture to handle label skewed non-iid image data : We propose a novel architecture which is inspired from MD-GAN[16] and train a global classifier model which will be able to handle the label skewed non-iid dataset. We name this approach as Multi-Generator MD-GAN with reset discriminator (MG-GAN-RD). We will discuss about the working of the model in chapter 3.

Analyze the privacy leakage in the proposed architecture by performing a membership inference attack : We perform a membership inference attack on out proposed architecture to

analyze privacy leakage in our solution and check whether its better than the previous approach. We will discuss the attack method in detail in chapter 3.

1.4 Research work flow

According to the research objectives, the report will describe the work flow as below:

Step 1

Step 2 Using NetGen tool set [2] to generate the network topology from specification data of WSN fields and Satellite on QuickMap. The network topology can be generated into Occam structure or Compute Unified Device Architecture (CUDA) structure (chapter 2).

Step 3 Developing and analyzing the distributed protocol algorithms for the cooperation between Satellite and WSN based on generated Occam structure (chapter 3).

Step 4 Using these proposed algorithms to develop a simulation with CUDA architecture on General Purpose Graphic Processing Units (GPGPU) (chapter 4). Moreover, proposing a specific debugger interface which allows to manage the simulation execution at high level.

Chapter 2

Literature review

2.1 Background

2.1.1 Federated Learning

2.1.2 Non-Identically Independently distributed Data Distribution

2.1.3 Generative Adversarial Networks

2.1.4 Membership Inference Attacks

2.2 Key related research

Literature Review.

1. **Title** - Advances and Open Problems in Federated Learning.

   Authors - P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... and Zhao,S.

   Contribution - This paper explains the recent advances and challenges in the federated learning domain. It defines a federated learning setting, which is a machine learning paradigm in which a global model is trained without accessing the client's local data. This paper provides an overview of fully decentralized and peer-to-peer distributed learning. It also discusses the federated model life-cycle and its training process. It covers a lot of topics such as improving the efficiency and effectiveness of the model, how to handle non-iid data, optimized algorithms for model architecture with non-iid setting, reducing

communication cost and latency, providing differential privacy to clients, how privacy can be maintained, different privacy threats to federated models, prevention of privacy threats, defending against malicious attacks, prevention of any failures in the federated model, synchronous and asynchronous communication between clients and the server or among the clients in a peer to peer structure. It also lists out various challenges and opportunities in improving training methods, the robustness of the model, and other system challenges.

2. **Title** - Federated Learning Challenges, methods, and future directions.

   Authors - Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith.

   Contribution - This paper defines and discusses the federated learning paradigm and provides a perspective from the internet of things point of view. It talks about the use of federated learning in smartphones and organizations where a global model is trained using the hyper-parameters of the local client models without accessing the private data of the clients, thus providing privacy to the clients. It defines some core challenges in the federated learning paradigm, which include Communication cost and latency, i.e., what is hardware and software cost for the communication to take place between the client and server, how many communication rounds are needed and the latency, System Heterogeneity, which covers systems with different hardware or software capabilities interacting with each other to train a global federated model, Statistical Heterogeneity, what kind of data the clients have? Data distribution affects the training process a lot and can cause high accuracy loss to the global model if not taken into consideration. It also discusses the privacy aspect of the federated architecture may cause. Finally, it discusses various schemes and techniques to train a federated model and its future direction.

2.3 Analysis

2.4 Research gaps

2.5 Problem formulation

2.6 Conclusion

Chapter 3

Methodology

This section introduces the hypothesis and the analytical validation of the proposed solution.

3.1 Proposed hypothesis

Our model architecture MG-GAN-RD (Multi-Generator MD-GAN with Reset Discriminator), as shown in fig. 3.1 is based on a client-server setup. We make use of GANs to learn the joint distribution of the non-iid dataset of each of the clients without accessing them, thus, providing data privacy. At the server, we have a generator pool and a global classifier. Each label has it's a respective generator at the server, which means the number of generators in the generator pool is equal to the number of labels in the dataset. Each client contains a discriminator, a storage unit for storing discriminator weights, a local classifier, and the non-iid dataset. The global classifier at the server will be trained after the tranining the GAN models has taken place.

Each client will separate its local dataset based on the labels. Training of the GAN model will happen in a label-wise iterative fashion.

Reset Property: During training with a specific label, the discriminator at each client can set its weights for training with that specific label with the help of the storage unit.

With a label-specific generator pool and client-specific discriminator with reset property, our model will be able to train a GAN model for each label without any redundant generator and discriminator models. This will increase the generated image quality, which will improve the training of global

classifier and also reduces the model and time complexity. Since for each label, we are training a GAN; we see better convergence rate and better image quality.

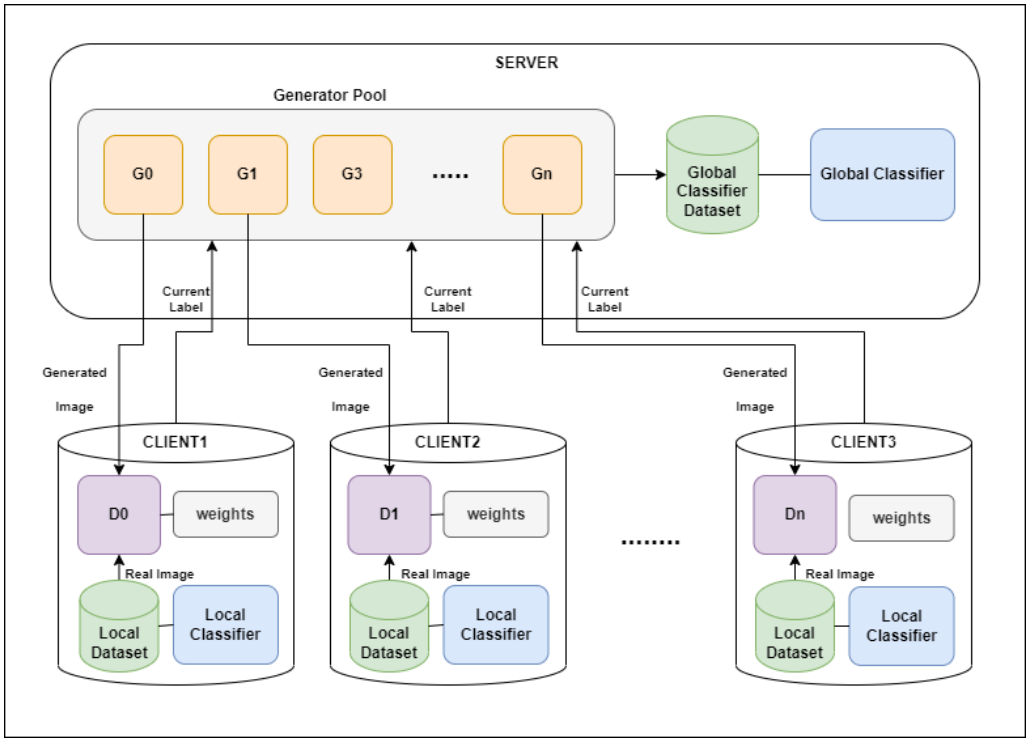


Figure 3.1: MG-GAN-RD Architecture

3.2 Mechanism/Algorithm

3.2.1 Model Description

3.2.1.1 CNN Architecture

We use a Convolutional Neural Network model to classify images. It consists of 3 convolutional blocks. Each Convolutional Block consist of the following:

- Convolutional Layer with 64 filters, kernel of size (5,5), with stride length of 1.
- It consists of ReLu activation.
- MaxPooling2D layer with pool size (2,2) and stride length of 1

Next it consists of a Flatten Layer, 2 Dense Layers with 64 hidden units and ReLu activation and lastly the output layer with 10 hidden units corresponding to the number of labels.We compile the model using Adam optimizer and Sparse Categorical Cross Entropy as the loss function.

Table 3.1: CNN Architecture

Layer	**Description**	**Hyper-Parameters**
1	Conv2D	(64, (5,5))
2	MaxPooling2D	((2,2))
3	Conv2D	(64, (5,5))
4	MaxPooling2D	((2,2))
5	Conv2D	(64, (5,5))
6	MaxPooling2D	((2,2))
7	Dense	(64)
8	Dense	(64)
9	Dense	(10)

3.2.1.2 GAN Architecture

GAN consists of two components : 1. Generator and 2. Discriminator.

1. Generator : The generator contains a one Densse block, Reshape layer, 2 Conv2DTranspose Block, and one Conv2DTranspose layer (see Table 3.2).

- The Dense block contains a Dense layer followed by Batch Normalization Layer and Leaky ReLu Activation Layer.

- The Conv2DTranspose Block contains one Conv2DTranspose Layer followed by a Batch Normalization Layer and a Leaky ReLu Activation Layer.

2. Discriminator : The Discriminator contains 2 Conv2D Blocks, a Flatten Layer and a single Dense Layer (see Table 3.3).

Table 3.2: Generator Architecture

Layer	Description	Hyper-Parameters
1	Dense Block	((7*7*256))
2	Reshape Layer	((7*7*256))
3	Conv2DTranspose Block	(128, (5,5), (1,1))
4	Conv2DTranspose Block	(64, (5,5), (2,2))
5	Conv2DTranspose Layer	(1, (5,5), (2,2))

- The Conv2D block contains a Conv2D layer followed by Leaky ReLu Activation Layer and a Dropout Layer with the dropout rate of 0.3.

We compile the discriminator using Adam optimizer with learning rate of value 0.0002 and beta_1 of value 0.5. We use Binary Cross entropy as loss function.

Table 3.3: Discriminator Architecture

Layer	Description	Hyper-Parameters
1	Conv2D Block	(64, (5,5), (2,2)
2	Conv2D Block	(128, (5,5), (2,2)
3	Dense Layer	(1)

3.2.1.3 Attack Classifier Architecture

The Attack Classifier is a binary classifier. We use Support Vector Classifier as our Attack Classifier with regularization parameter as 1 and Radial Basis Function as the kernel.

3.2.2 Training Local Client Classifier and a Standard Classifier

1. Each client has a local classifier which is the Local Client Classifier. We use CNN Architecture defined in section 3.2.1.1 as the model for out local client classifier. Each client has it's own dataset on which the local classifier trains upon. Since the data is non-iid, the accuracy of the local client classifier is relatively low.

2. We also train a Standard Classifier on the complete dataset. We use the CNN architecture defined in section 3.2.1.1 as the model for our Standard Classifier. The Standard Classifier represents the ideal condition, i.e., the limit of accuracy the defined model can reach it trained on a complete dataset. This will be useful for comparison purposes.

3.2.3 Training Global Generator Pool

1. A generator pool is present at the server. The generator pool consists of a number of generator instances. We use the generator architecture defined in section 3.2.1.2 for defining our generator instances. The number of generator instances equals the total number of labels the complete dataset has. Thus each generator instance participates in training each label.

2. During the training phase, Each client informs the server of the current label it is training on, i.e., the client will train its discriminator on a specific label batch data. Thus, that client will get generated images from the generator instance of that specific label.

3. After the server gets the client's output, the generator loss of that generator instance is calculated and used to update the weights of that generator instance.

4. Since each generator instance trains on a specific label. The generator instance can communicate with one or many clients, depending upon the label the client is training it's own discriminator on. Thus, a generator pool with number of generator instances equal to the number of labels can manage a large amount of clients.

3.2.4 Training Local Discriminator

1. The client creates label specific batches of the local data. It then selects which label it wants to train with the discriminator. The client tells the server the label it is currently training and then it gets coupled the generator instance in the generator pool which trains on that specific label.

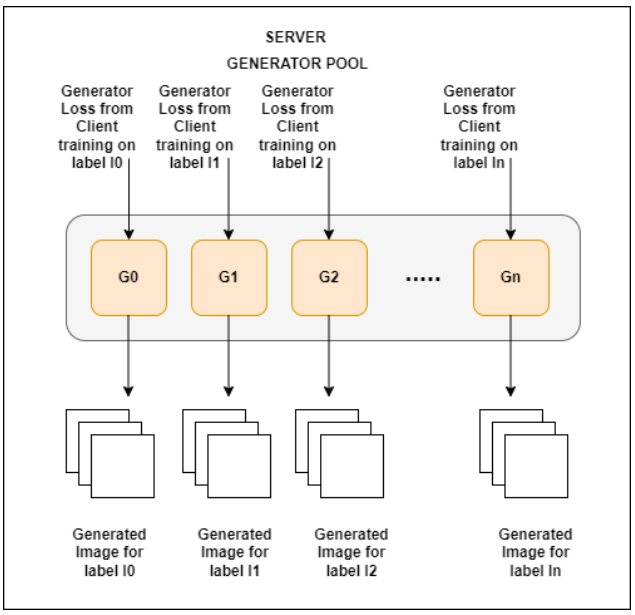


Figure 3.2: Generator Pool at the Server.

2. The generator instance sends some fake samples of the specific label to the client which is given as input to the discriminator along with the real images of that specific label.

3. After the discriminator finishes training on the data, it then stores the computed model weights into the storage unit. And initializes as a new untrained discriminator model.

4. The client repeats the process for the next label. The discriminator queries the storage unit for that particular label model weights. If present , then discriminator set it weights to the label specific, previously calculated model weights. If the label specific model weights are not present then the discriminator initializes as a new discriminator model.

3.2.5 Training the Global Classifier

1. Once all the generator instances in the generator pool are trained. They are used to generate a significant amount of dataset which will be stored in the Global Classifier Dataset. Since each generator instance is coupled with a single label. We need not query client's local classifiers for the predicting the label.

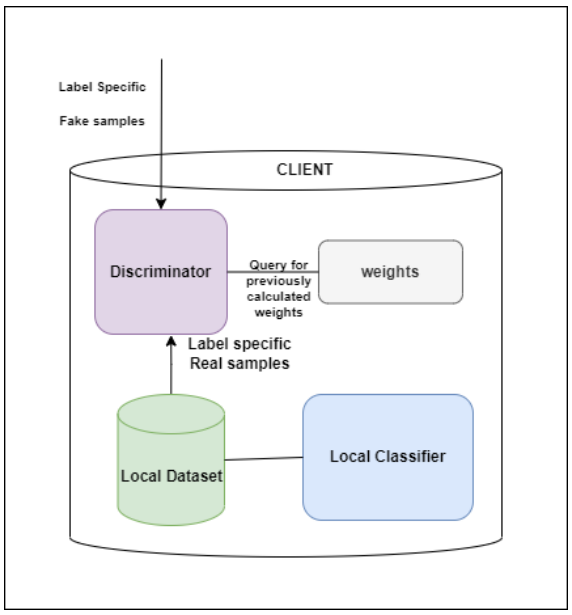


Figure 3.3: Discriminator at Client.

2. Using the synthetic dataset, we train our global classifier model. If the generators have learned image distribution properly, then it should create a dataset statistically closer the original dataset which means higher accuracy of our global classifier.

3. Once training of global classifier is completed, it will be available to all the clients.

3.2.6 Performing GAN enhanced Membership Inference Attack

1. Now we perform the membership inference attack on our federated architecture. Since the the server need not query the clients local classifier for the prediction of synthetic images, there is no synthetic data to use for the membership inference attack.

2. We assume one of the participants in the federated learning is a malicious user which will perform the membership inference attack.

3. Shadow models are defined. The number of shadow models equals the number of labels the attacker had during the training. Shadow model CNN model only whose architecture is defined in section 3.2.1.1.

4. Since the attacker is also a legitimate client in the opinion of the server, the attacker can access the global classifier model. We use GANs to generate synthetic data for the attack.

5. The generator of the GAN is based on the architecture defined in the section 3.2.1.2 and the discriminator architecture is based on the global classifier. The discriminator is initialized with the global classifier. So the GAN will generate images similar to the distribution of the original dataset.

6. Once we have the synthetic images, we query the images to the global classifier to get the labels for the generated images. We then prepare the dataset for out shadow model (see fig. 3.4).

7. A part of the dataset including the synthetic images in prepared as training dataset for the shadow model and the rest is testing data. The shadow models are trained.

8. After training the shadow models, we get the prediction output for all the training and testing data. We then label prediction output for training data as 1 and prediction output of the testing data is labelled as 0.

9. label 1 means the data point was part of the training process and 0 means the data point was not part of the training process. Thus, if we have a data point we can find out if it was a part of the training of the machine learning model.

10. We then prepare the dataset for the attacker. The attacker is a binary classifier, it's architecture is defined in the section 3.2.1.3. The for all the images in the training set, their prediction output and the label is the input values for the attack classifier and the label indicating whether they were part of training or not is the output (see fig. 3.5).

11. The MIA attack classifier is trained and its attack success rate is measured. The attack success rate will indicate the level of privacy leakage in the federated architecture.

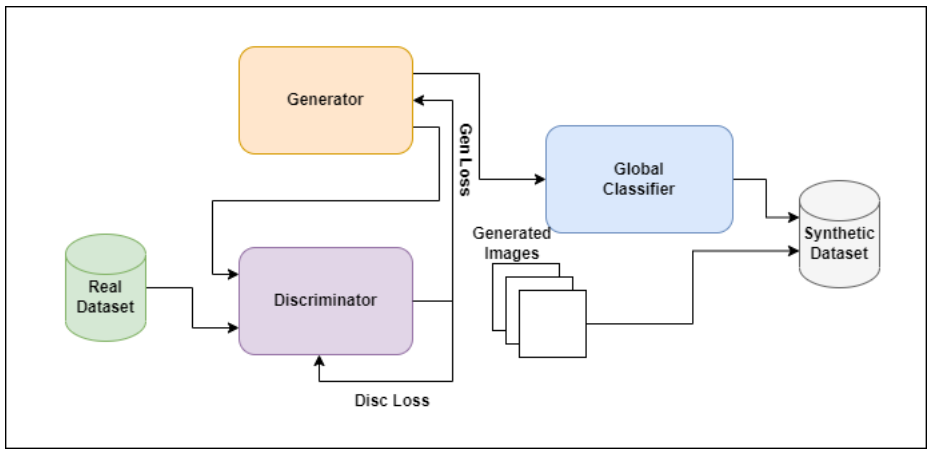


Figure 3.4: Generating Synthetic Data for MIA.

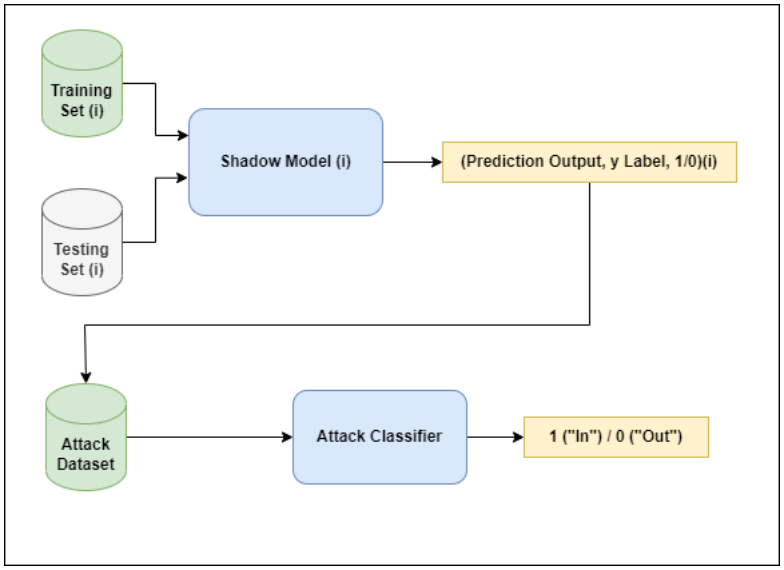


Figure 3.5: Membership Inference Attack.

3.2.7 Evaluation Metrics

The following are the evaluation metrics which will be used to measure the performance of the models, quality of the images generated and the success rate of an attacker model.

3.2.7.1 Accuracy

Accuracy is the total number of correct prediction out of total number of samples. It is one of the common metric used to evaluate classification models. It provides a metric of how correct the model is.

3.2.7.2 Structural Similarity Index

Structural Similarity Index [13] is used to evaluate GANs for the quality of image they produce. It used to calculate the similarity between two images. Comparison of two images is based on the factors like luminance, contrast and structure. It is bounded between -1 and 1.

3.2.7.3 Attack Success Rate

Attack success rate indicates the amount of times the attack was successful out of all the initiated attacks. It is same as calculating the accuracy of the attack model. It indicates how good the attack model is.

3.3 Analytical validation

1. Our solution to solving the learning of non-iid dataset is based on the fact that GAN models trained for specific labels produce much better result. Since the GAN has to only learn one label, it is able to learn the underlying distribution of the images of the same label. Our model removes the non-iid effect on the training process by training individual GAN for each label. Whether the client data is iid or non-iid, our solution will be consistent.

2. Training of GAN per label is better because images of same label are more similar to each other than the images from the other label (see fig. 3.6). Instead of learning the joint distribution, our model learns individual distributions of the labels and it seems that the

whole architecture has learnt the joint distribution.

3. This is also supported by the fact that GANs learning individual labels have faster convergence rate (see fig. 3.7 3.8). In our federated architecture we were able to take benefit of this fact and were able to learn the joint distribution of the dataset by learning the individual distribution of the labels without any redundant generators or discriminators. In fact, with limited number of generators , our model can accommodate large number of clients.

4. Our experiments in chapter 4 with further demonstrate that the images generated by our model are of better quality and the global classifier trained on the synthetic dataset using our model approach gives good results.

5. NOTE : In fig. 3.8 the discriminator loss is increasing because the discriminator loss here is the sum of real loss and fake loss. Thus, increase in discriminator loss is due to increase in fake loss. Increase in fake loss indicates that the fake image is much closer to the real image.

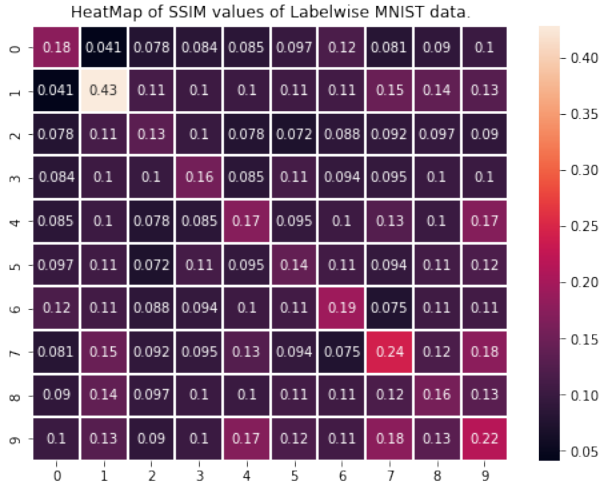


Figure 3.6: Heatmap of SSIM values for label-wise MNIST data.

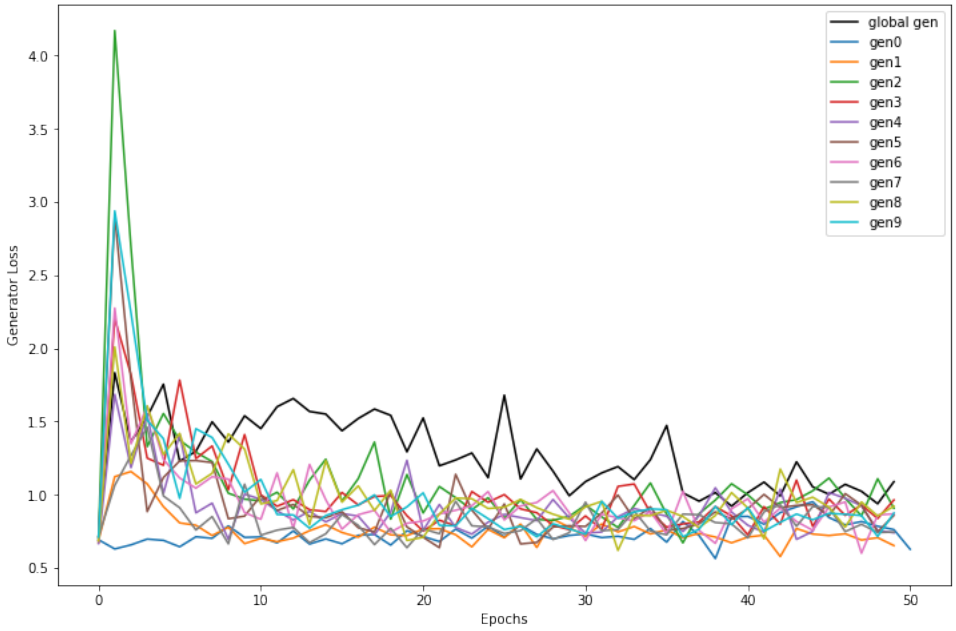


Figure 3.7: Generating Synthetic Data for MIA.

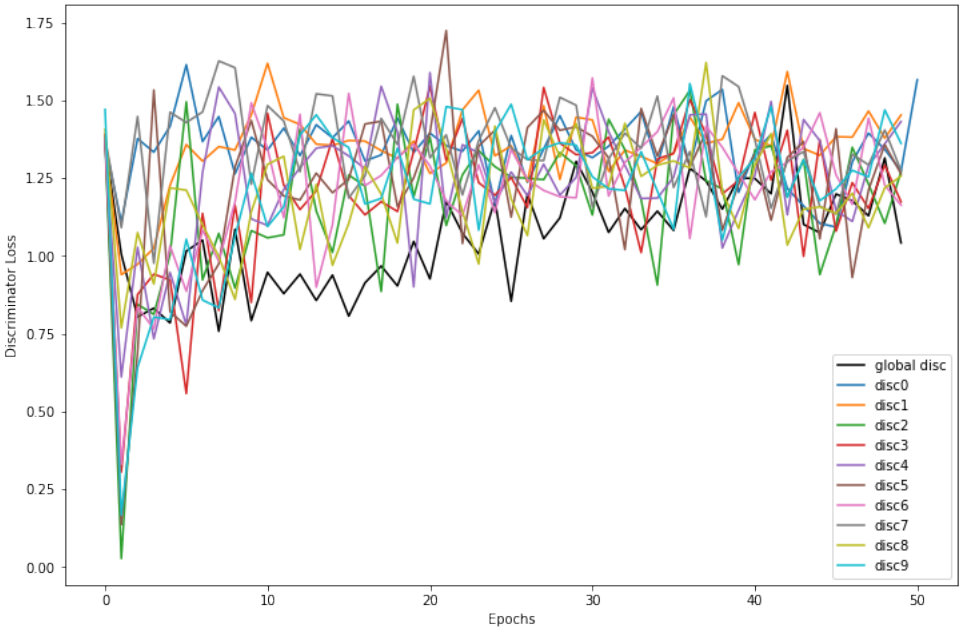


Figure 3.8: Membership Inference Attack.

3.4 Conclusion

In this chapter, we have proposed our hypothesis with the analytical validation and explained the methodology for our Multi-Generator MD-GAN with Reset Discriminator architecture. In the next chapter, i.e., chapter 4, we will be performing experiments which will demonstrate the better performance of our model.

Chapter 4

Experiments and results

This section discusses the various experiments pertaining to the proposed hypothesis and their findings.

4.1 Experiment design

We have a client-server architecture setup. Based on this setup we perform experiments of training a MD-GAN based global classifier (see Appendix A), training of our proposed model, and training of a state-of-the-art federate learning scheme (see Appendix B). We use two data sets - MNIST digit dataset and Fashion MNIST dataset for out experiments.

4.2 Training Local Classifier and Standard Classifier

4.2.1 Parameter settings

We train a local classifier for each each dataset and for each dataset we have different number of clients. We take 2 cases for each dataset with number of clients as 3 and 10.

4.2.2 Experiment description

We define a standard classifier model: standard_model

For number of clients = 3, we define 3 local classifiers:

- client1_local_model

- client2_local_model
- client3_local_model

For number of clients = 10, we define 10 local classifiers:

- client0_local_model
- client1_local_model
- client2_local_model
- client3_local_model
- client4_local_model
- client5_local_model
- client6_local_model
- client7_local_model
- client8_local_model
- client9_local_model

4.2.3 Results

The following tables (Table 4.1, 4.2 4.3) show the test loss and test accuracy of standard and local classifier

Table 4.1: Evaluation of Standard Classifier

Model	**Dataset**	**Test Loss**	**Test Accuracy**
standard_model	MNIST	0.0383	0.9909
	FMNIST	0.4031	0.8952

Table 4.2: Evaluation of Local Classifier with Number of clients = 3

Model	Dataset	Test Loss	Test Accuracy
Client1_local_model	MNIST	9.2616	0.3877
	FMNIST	18.7311	0.3905
Client2_local_model	MNIST	16.4997	0.3058
	FMNIST	44.1642	0.2949
Client3_local_model	MNIST	12.5581	0.3025
	FMNIST	23.3917	0.2856

Table 4.3: Evaluation of Local Classifier with Number of clients = 10

Model	Dataset	Test Loss	Test Accuracy
Client0_local_model	MNIST	68.3671	0.0980
	FMNIST	88.6014	0.1000
Client1_local_model	MNIST	88.2528	0.1135
	FMNIST	69.4871	0.1000
Client2_local_model	MNIST	68.7307	0.1032
	FMNIST	89.9049	0.1000
Client3_local_model	MNIST	77.5681	0.1010
	FMNIST	78.7531	0.1000
Client4_local_model	MNIST	62.7917	0.0982
	FMNIST	63.5702	0.1000
Client5_local_model	MNIST	79.7707	0.0892
	FMNIST	125.3997	0.1000
Client6_local_model	MNIST	79.6432	0.0958
	FMNIST	88.2672	0.1000
Client7_local_model	MNIST	87.0097	0.1028
	FMNIST	83.4782	0.1000
Client8_local_model	MNIST	64.8761	0.0974
	FMNIST	70.8772	0.1000
Client9_local_model	MNIST	2.3035	0.1071
	FMNIST	2.3044	0.1000

4.3 Training MG-GAN-RD based Global Classifier

As mentioned earlier we perform the experiment on two data sets - MNIST and Fashion MNIST. With different number of participating clients.

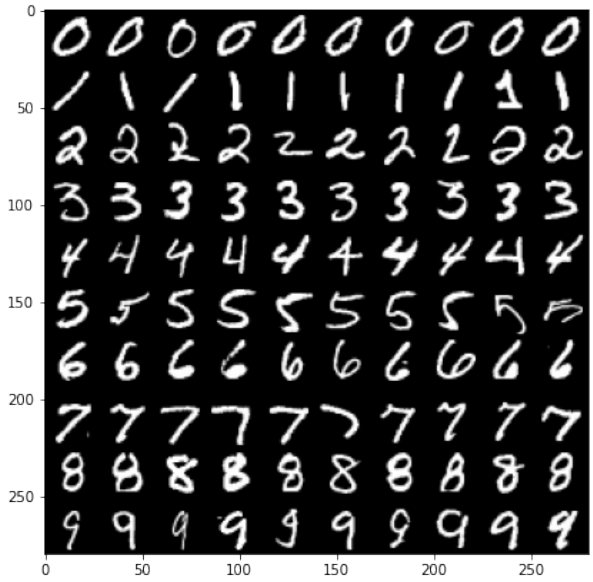
4.3.1 Parameter settings

For each Dataset, i.e., MNIST and Fashion MNIST, we perform the experiment for number of clients equal to 3 and 10.

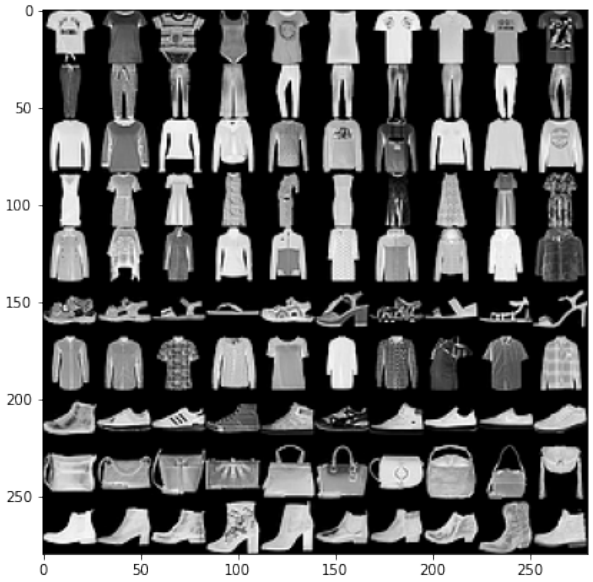
4.3.2 Experiment description

We train a global classifier based on MG-GAN-RD. After training the MG-GAN-RD, we create a synthetic dataset to train our classifier on.

4.3.3 Results



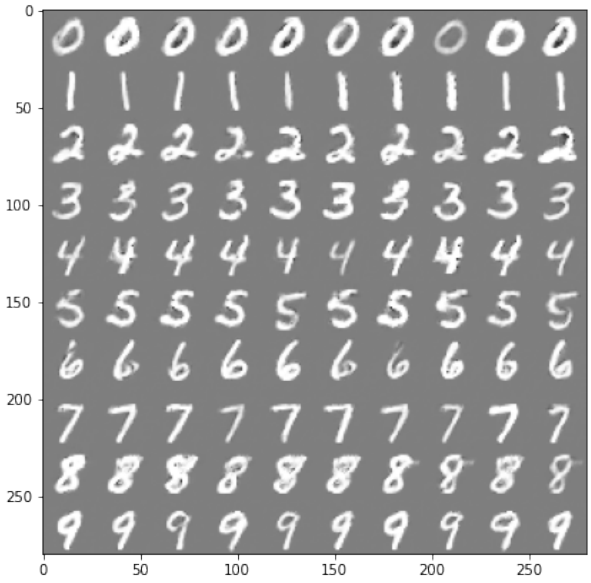
(a) MNIST Dataset



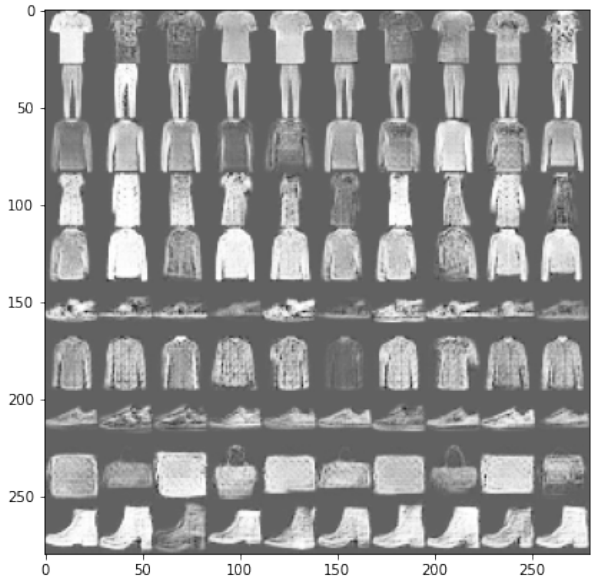
(b) Fashion MNIST Dataset

Table 4.4: Evaluation of MG-GAN-RD Based Classifier

Model	Dataset	Num_Clients	Test Loss	Test Accuracy
global_classifier	MNIST	3	0.0383	0.9909
		10	0.0	0.0
	FMNIST	3	0.0383	0.9909
		10	0.0	0.0



(a) MNIST Synthetic Dataset



(b) Fashion MNIST Synthetic Dataset

Table 4.5: Evaluation of Quality of Images

Quality Metric	**Dataset**	**Metric Value**
SSIM	MNIST	3
	FMNIST	0.0

4.4 Training MD-GAN based Global Classifier

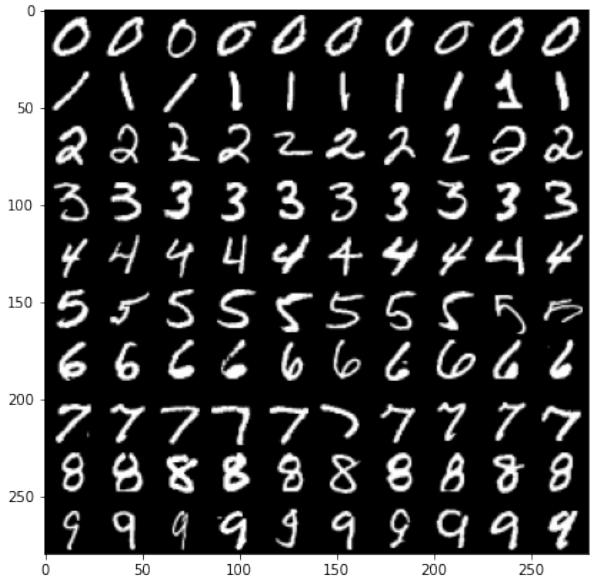
4.4.1 Parameter settings

4.4.2 Experiment description

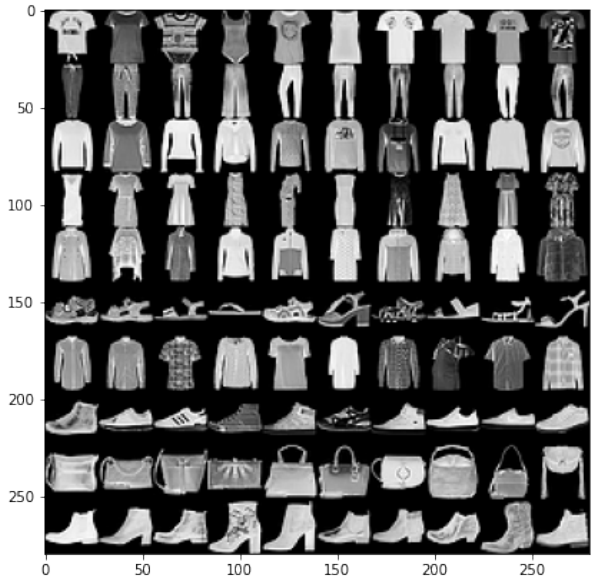
4.4.3 Results

Table 4.6: Evaluation of MD-GAN Based Classifier

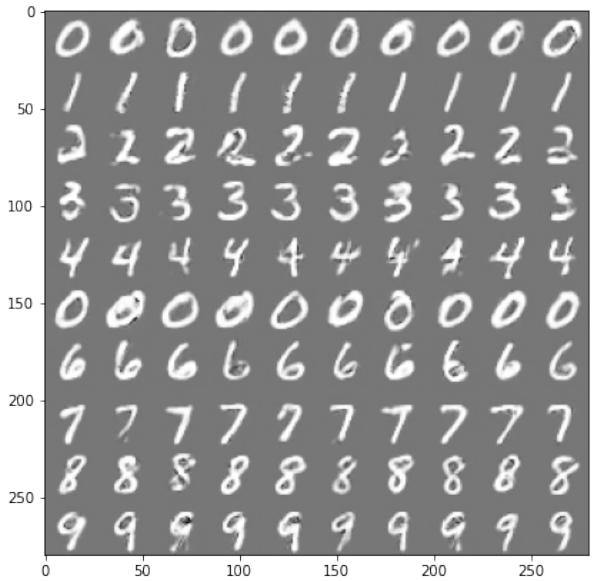
Model	**Dataset**	**Num_Clients**	**Test Loss**	**Test Accuracy**
global_classifier	MNIST	3	0.0383	0.9909
		10	0.0	0.0
	FMNIST	3	0.0383	0.9909
		10	0.0	0.0



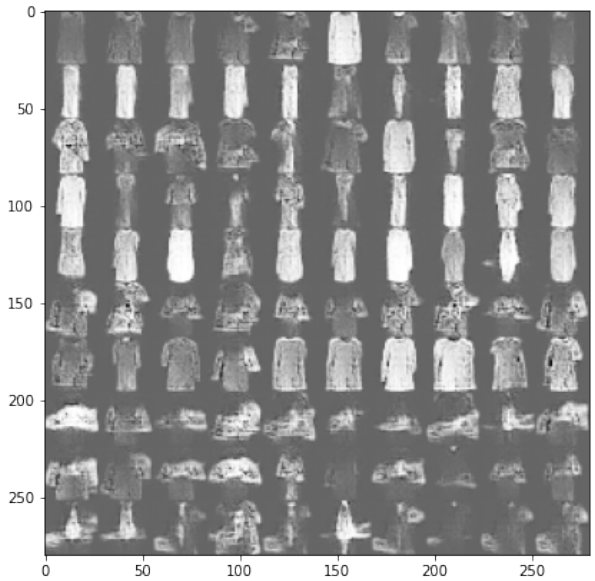
(a) MNIST Dataset



(b) Fashion MNIST Dataset



(a) MNIST Dataset



(b) Fashion MNIST Dataset

Table 4.7: Evaluation of Quality of Images

Quality Metric	Dataset	Metric Value
SSIM	MNIST	3
	FMNIST	0.0

4.5 Training Global Classifier using FedAvg

4.5.1 Parameter settings

4.5.2 Experiment description

4.5.3 Results

Table 4.8: Evaluation of FedAvg Classifier

Model	Dataset	Num_Clients	Test Loss	Test Accuracy
global_classifier	MNIST	3	0.0383	0.9909
		10	0.0	0.0
	FMNIST	3	0.0383	0.9909
		10	0.0	0.0

4.6 Membership Inference attack on MD-GAN Model

4.6.1 Parameter settings

4.6.2 Experiment description

4.6.3 Results

Table 4.9: Evaluation of Attack Classifier

Model	Dataset	Num_Clients	Attack Success Rate
attack_classifier	MNIST	3	0.0383
		10	0.0
	FMNIST	3	0.0383
		10	0.0

4.7 GAN enhanced Membership Inference attack on MG-GAN-RD Model

4.7.1 Parameter settings

4.7.2 Experiment description

4.7.3 Results

Table 4.10: Evaluation of Attack Classifier

Model	Dataset	Num_Clients	Attack Success Rate
attack_classifier	MNIST	3	0.0383
		10	0.0
	FMNIST	3	0.0383
		10	0.0

4.8 Overall conclusion

In this section, relate the conclusions obtained in the above experiments with the gaps identified in Chapter 2. Derive conclusion about how far the set gaps were met and if not, the reason for the deviation.

Chapter 5

Discussions and conclusion

In this chapter, the work is concluded and future plan is presented. Next, the research contribution are presented. Finally, limitation of the work and possible future extensions are described respectively.

5.1 Contributions

(1) The proposed federated architecture performs consistently better than its counterparts. Our model bridges the gap between the iid and non-iid dataset. Whether the client has iid or non-iid dataset. Since our model learns label specific distribution, it will perform consistently well and its performance will not deteriorate.

(2) The proposed federated architecture can handle large amount of clients without the increasing model complexity at the server. The number of generator instances in the generator pool depends upon the number of classification in the dataset. So with finite amount of generator instance, the proposed architecture can train with a large amount of clients.

(3) **Label Privacy** : Since before training , each client gets coupled with a generator instance by providing the label to the server , the client is currently training on, it can lead to leakage of label data. However, we can replace the labels with an arbitrarily generated random id and use that for coupling the client with the generator instance, because label is not important , but the order in which the client is training the data.

5.2 Limitations

(1) The proposed model is less susceptible to membership inference attack due to its inherent architecture but still the privacy leakage issue persists.

(2) It can happen that the server decides to make the generator pool accessible to clients, then also it can be a serious threat to data privacy. However, doing behaviour analysis of the clients might be able to detect a malicious client/User and prevent further privacy loss.

5.3 Future scope

(1) We can use more complex GAN models like WGAN, CGAN etc. to train the generator pool for better accuracy and performance of the model, and also improve the quality of image generated at the server.

(2) There is a scope of reducing the privacy leakage in the proposed architecture to make it more safe and robust.

(3) The proposed model is highly scalable but it will also increase the model complexity. In the future, we might be able create highly scalable model inspired from our proposed architecture without increasing the model complexity.

Bibliography

[1] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2):1–210, 2021.

[2] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3):50–60, 2020.

[3] Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. IEEE Communications Surveys & Tutorials, 2021.

[4] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133, 2020.

[5] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. Future Generation Computer Systems, 115:619–640, 2021.

[6] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. Neurocomputing, 465:371–390, 2021.

[7] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. arXiv preprint arXiv:1806.00582, 2018.

[8] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. arXiv preprint arXiv:2102.02079, 2021.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[10] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. Advances in neural information processing systems, 29, 2016.

[11] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In International Conference on Machine Learning, pages 634–643. PMLR, 2019.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. PMLR, 2017.

[15] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems, 2:429–450, 2020.

[16] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In 2019 IEEE international parallel and distributed processing symposium (IPDPS), pages 866–877. IEEE, 2019.

[17] Yuezhou Wu, Yan Kang, Jiahuan Luo, Yuanqin He, and Qiang Yang. Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning. arXiv preprint arXiv:2111.08211, 2021.

[18] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. ACM Computing Surveys (CSUR), 2021.

[19] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP), pages 3–18. IEEE, 2017.

[20] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. Gan enhanced membership inference: A passive local attack in federated learning. In ICC 2020-2020 IEEE International Conference on Communications (ICC), pages 1–6. IEEE, 2020.

Appendix A

Training of MD-GAN based Classifier

A.1 Training the Global Classifier

(1) MG-GAN architecture is a client-server architecture in which a global generator is present at the server and each client has a local discriminator model. The global generator creates fake samples which are distributed to all the clients and then after passing the fake samples and real samples from the local dataset to the local discriminator , we can get the generator loss value. This generator loss in sent back to the server, where the global generator calculates the gradients and update it's weights.

(2) An MD-GAN based classifier makes use of the MD-GAN architecture to train a global classifier. In addition to the discriminator, each client also contains a local classifier. After training the global generator, the server produces lots of synthetic data which are then distributed to all the clients.

(3) After getting the synthetic data, each client passes the data to the local classifier where it predicts the label of the synthetic data. The label and the prediction output for that label, for all the synthetic images is sent back to the server by all clients.

(4) The server, after receiving the output from all the clients, then use this output values to

create a global dataset to train the global classifier. For each synthetic images it finds the maximum prediction output from the prediction output from all clients. The label corresponding to the maximum prediction value becomes the label for the synthetic image.

(5) Once the dataset is created, the global classifier is trained on the dataset and the trained global classifier is then made available to all the participating clients.

A.2 Privacy Leakage

(1) Since training of the global classifier requires the server distributing synthetic images to all the clients and getting the output. It opens the door to serious privacy leakage in the model.

(2) Suppose one of the participating clients is a malicious attacker. After training the global classifier, the attacker will also get a large amount of synthetic data. Combined with the original dataset, the attacker can now perform a membership inference attack on the model.

(3) It can use the synthetic data combined with its local data, now has a dataset which is statistically similar to the original dataset. The attacker can now create shadow models which are based on the global classifier architecture and train the shadow models to prepare the attack dataset. The attacker then can use the attack dataset to train a binary classifier which will be able to tell if a particular data point was part of the training/original dataset or not

Appendix B

FedAvg

B.1 Training a FedAvg Model

(1) McMahan et. al. [14] introduced the concept of federated learning and provided the fedavg algorithm to train a federated model. This algorithm provided a way to train the local machine learning model without sharing the private dataset of the client. This removed the communication heavy data transfer to the server and also provided data privacy and security to the clients.

(2) The architecture is assumed to be a client-server architecture. Each client contains a local dataset and a local machine learning model. At the server, aggregation of the model parameters takes place , which once calculated are provided to all the participating clients.

(3) The sever provides the initial weights to all the clients, thus all local machine learning models are initialized using same weights. Each client trains the local machine learning model on the local dataset. After the training, the client sends the model weights to the server.

(4) After collecting the model weights of all client's machine learning model. It averages the weights and sends it to all the clients. The machine learning model at the client then updates its weights. This constitutes one communication round. After several such rounds, the training process is over.