Generative Federated Learning Approaches for Non-IID Data

Enhancing Federated Models with Synthetic Data

Pil Kyu Cho

Supervisor(s): David Tax, Swier Garst

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Pil Kyu Cho
Final project course: CSE3000 Research Project
Thesis committee: David Tax, Swier Garst, Alex Voulimeneas

Abstract

Federated Learning (FL) is a machine learning approach that has gained considerable interest over the years. FL allows global models to train without compromising the data privacy of the clients' training datasets by sending the global model to each client to learn the weights and propagating only the learned weights back to a central location. However, it is not without limitations as several challenges hinder the model's performance. One of those challenges is the presence of non-IID (Independent and Identically Distributed) properties in the training data. Most real-world data is non-IID, and this imbalance in data distribution has been shown to significantly affect the model's performance. To address this issue, we propose a generative federated learning by pre-training the global model on synthetic data created by a generative model that follows the collective distribution of all clients' training datasets. Our research shows that this approach bridges the performance gap between IID and non-IID in FL, except for certain extreme non-IID cases.

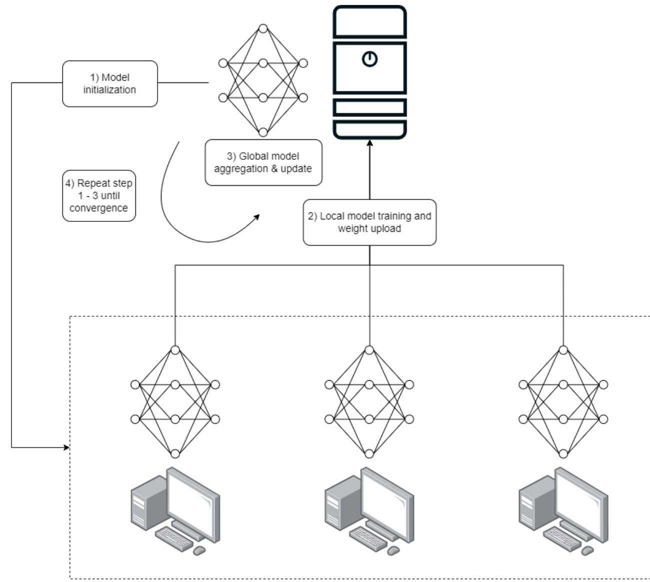


Figure 1: FL process: (1) Initialize model in central server and send to clients. (2) Clients train model on their respective local datasets and upload weights to central server. (3) Central server aggregates weights and updates global model. (4) Repeat steps 1 - 3 until global model converges.

1 Introduction to GenFL

1.1 Background to Federated Learning

Federated Learning (FL) is a distributed machine learning approach where multiple users, known as clients, train identical models locally. First introduced by Google researchers in 2016 [1], its purpose is to maintain the privacy of each client's training data. In that regard, FL has been particularly useful in domains, such as medical or financial, where data cannot be stored in a centralized location due to data privacy regulations such as GDPR.

The FL process is coordinated by a central server, where the clients send the weights of their trained models back to this central server. These weights are then combined using some aggregation method, such as averaging, to form a global model. This is an iterative method that can be repeated multiple times to improve the model's performance. Figure 1 shows these steps of a typical FL process. This demonstrates how FL allows machine learning models to train without centralizing the distributed datasets.

1.2 Background to Non-IID

A significant challenge in FL is non-IID (Independent and Identically Distributed) data. IID data indicates that the samples are independently drawn from a fixed distribution and each client dataset has the same distribution. In a non-IID setting, clients can have data distributions that differ greatly from each other, resulting in imbalanced datasets. For instance, a hospital that is close to an industrial area that releases toxic fumes may have significantly more lung cancer patients than other hospitals. Most real-life data are non-IID but models are usually trained based on the assumption that data is IID, thus leading to significant performance reduction when the data is non-IID [2]. This has led to non-IID data being one of the most significant challenges to the advancement of FL [3].

1.3 A New Approach: GenFL

In this paper, we propose a new generative FL approach called "GenFL". It introduces a generative model that has the collective distribution of each client's dataset. This is achieved by training the generative model itself using FL with the same clients that will later participate in the FL process of the downstream model. The generative model will create synthetic data, which the downstream model will pre-train on in the central server before its FL process begins. Figure 2 shows step-by-step the process of GenFL. This demonstrates how the generative model is introduced into the FL process and used with the downstream model.

1.4 Research Questions

We investigate the effectiveness of GenFL on bridging the performance degradation in non-IID settings and to what degree of data imbalance it can support. The goal is to provide more insight as to how training the downstream model with synthetic data beforehand can bridge the performance gap between IID and non-IID in FL. To achieve this, this paper aims to answer the following research question:

Would pre-training the downstream model with synthetic data help its performance during the federated training on non-IID data?

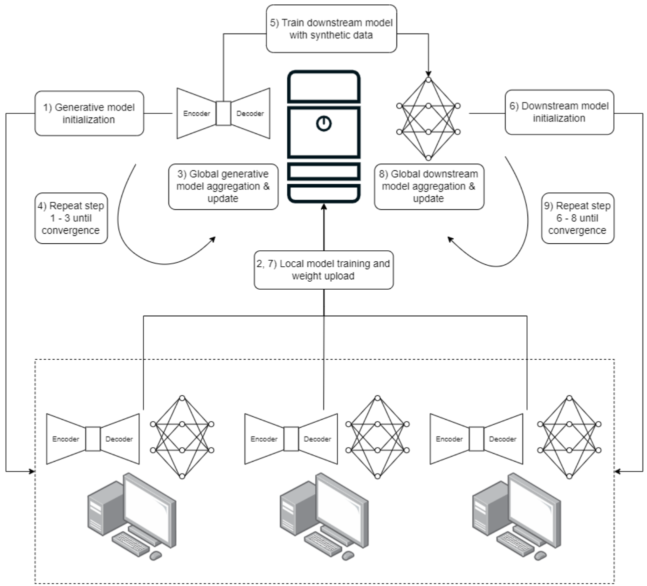


Figure 2: GenFL process: (1) Initialize generative model in central server and send to clients. (2) Clients train generative model on their respective local datasets and upload weights to central server. (3) Central server aggregates weights and updates global generative model. (4) Repeat steps 1 - 3 until global generative model converges. (5) Use generative model to create synthetic data to pre-train downstream model. (6) Send pre-trained downstream model to clients. (7) Clients train downstream model on their respective local datasets and upload weights to central server. (8) Central server aggregates weights and updates global downstream model. (9) Repeat steps 6 - 8 until global downstream model converges.

The qualitative main question can be further divided into the following quantitative sub-questions:

- What is the performance difference for downstream models that were pre-trained on synthetic data in the central server compared to those that did not?
- What is the performance impact on GenFL as data imbalance increases?

2 Research Methodology and Experiment Setup

2.1 Related Work

GenFL combines two widely researched topics in the machine learning community: 1) using synthetic data as alternatives to real data and 2) addressing the non-IID challenge prevalent in FL.

There have been extensive studies in the context of FL for generating synthetic data [4]. One possible reason for such interest is due to the advancements in AI-Generated Content and research is prevalent on how they can be alternatives to real data in scenarios where the real data is not suited for training models.

Solving the non-IID challenge is another heavily researched topic in FL [3]. This may arise from the fact that most real-world data is non-IID and the presence of non-IID in FL has been shown to considerably decrease the performance of the model [2].

[5] explores how pre-training the downstream model either with real or synthetic data can consistently improve its accuracy during the FL process. While this process very closely resembles this research, there are some key differences. When pre-training the downstream model on synthetic data, [5] experiments over only one data imbalance setting. Our research goes beyond this and tests over several data imbalance settings to observe the behaviour as the imbalance increases. Moreover, we also test over a higher data imbalance setting.

[6] also attempts to bridge the performance gap in non-IID settings of FL using synthetic data from a pre-trained generative model. The key difference is that it uses data augmentation by generating images for each client based on its unique data distribution disparities, creating an augmented dataset that becomes IID for each unique client.

Other papers attempt similar but different approaches to address the non-IID challenge in FL. This includes globally sharing a small subset of data [7] or a new benchmark called NIID-bench [8] to address the various types of non-IID settings that can be present in FL. Some papers further address the privacy issue that arises when introducing a generative model into the FL process [9].

2.2 Experiment Setup

Federated Learning Algorithm

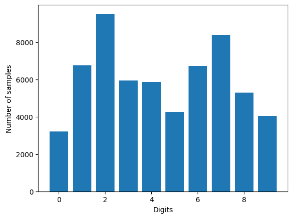
For the federated process of both the generative model and the downstream model, we use FedAVG [10]. This is a classic aggregation algorithm in FL, where the weights of optimized client models are shared and the central server averages these weights to the global model. While there are other FL algorithms, such as FedSGD or FedPA-GD, we chose FedAVG for its simplicity and popularity in FL.

Data Imbalance

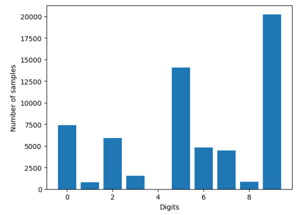
To set the degree of imbalance in the dataset, we use the Dirichlet distribution [11] to set different ratios of class labels and allocate different numbers of data samples to each client. This can be controlled by the Dirichlet distribution's $\alpha > 0$ parameter. As mentioned in 2.1, [5] only experiments with one data imbalance setting, which is $\alpha = 0.1$. For our research, we explore over difference imbalance settings, with the α value as 0.01, 0.1, 1.0, 10.0 and 100.0, where 0.01 is the most imbalanced and 100.0 is the least imbalanced (closest to IID).

Generative Model and Metrics

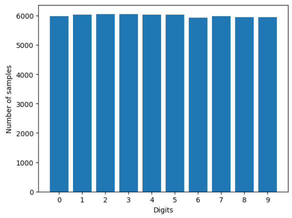
Two commonly used generative approaches in synthetic data research are Variational Autoencoders (VAE) [12] and Generative Adversarial Networks (GAN) [13]. We decided to use VAE as our generative model as they are more stable for training and better for estimating the probability distribution



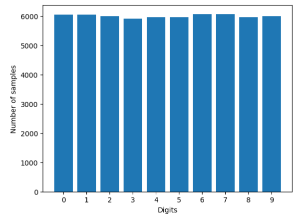
(a) VAE Dirichlet $\alpha = 1.0$



(b) VAE Dirichlet $\alpha = 0.1$



(c) CVAE Dirichlet $\alpha = 1.0$



(d) CVAE Dirichlet $\alpha = 0.1$

Figure 3: Number of generated samples for each class using MNIST, with (a) VAE Dirichlet $\alpha = 1.0$, (b) VAE Dirichlet $\alpha = 0.1$, (c) CVAE Dirichlet $\alpha = 1.0$, (d) CVAE Dirichlet $\alpha = 0.1$.

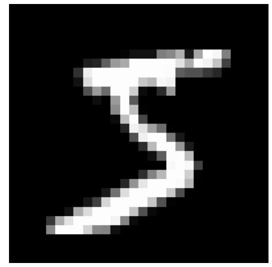
[14]. We used a VAE architecture from Google Colab [1]. After training the VAE in a federated manner, however, we realized that the VAE was not adequate for our experiment, as it gave us no control in the process of the random data generation. As a result, the VAE generated imbalanced datasets, which became worse as the Dirichlet α value decreased. Therefore, we switched to a Conditional Variational Autoencoder (CVAE) [15] instead. By conditioning on the class label, it resulted in a balanced dataset that was suited for our experiment. Figure 3 shows the imbalanced synthetic data from the VAE compared to the balanced synthetic data from the CVAE.

We use the Classification Accuracy Score (CAS) to evaluate the performance of the CVAE, which is a metric based on the accuracy of a classifier trained on synthetic images [16] created by the generative model and tested on real images. We do not use traditional image quality metrics such as Frechet Inception Distance (FID) as they are not indicative of suitability for training downstream models [16].

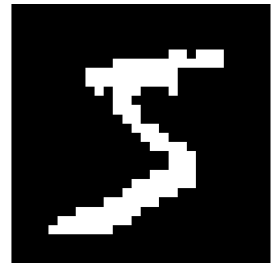
Datasets

For the datasets, we use MNIST and FashionMNIST as these are standard benchmark datasets widely used to develop and test generative models. Moreover, both datasets consist of grayscale images with 28x28 pixels and 10 classes. This simplicity makes it ideal to test on our CVAE. While we also considered CIFAR10, this proved to be a too complex dataset for our CVAE to handle, as the generated images were too poor to be used for downstream tasks.

[1] https://colab.research.google.com/drive/1wMTCIPeyim3r6DHPC2jDs2pbur7-WHOO?usp=sharing



(a) Before transformation



(b) After transformation

Figure 4: Comparison of MNIST images (a) before transformation and (b) after transformation.

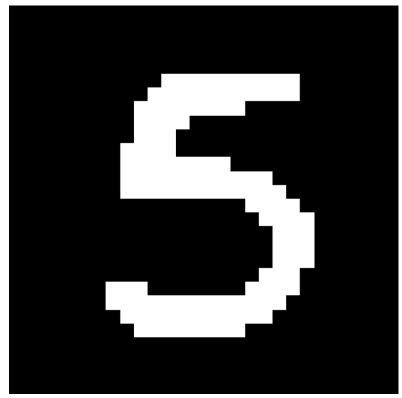


Figure 5: Synthetic image from CVAE trained on transformed MNIST images.

Both the MNIST and FashionMNIST datasets were already normalized in the range of [0, 1]. However, while the shape of the generated synthetic images was consistent with the real images, the different pixel values between the synthetic image and real image caused the CAS to drop to as low as 30%. Further attempts have been made to change the architecture of the CVAE but have shown no improvement. Therefore, additional transformations on the real dataset were required for both training and testing. We rounded the already normalized pixel values to their nearest integer, resulting in images with binary pixel values of 0 and 1. Figure 4 shows the real images before and after applying the transformation, and figure 5 shows a synthetic image from the CVAE that was trained on the transformed dataset. This resulted in an 89% CAS when we trained the classifier on the synthetic images and tested on real images, thus allowing us to proceed with the experiment.

Downstream Model and Metrics

The downstream model we use is ExquisiteNetV1 from [17], which is a lightweight CNN that can be used for image classification. We chose ExquisitenetV1 due to its small size and fast computation speed, while still delivering adequate performance. This makes ExquisitenetV1 very suited for downstream tasks in FL, particularly when the participating clients may be relatively low-computational devices such as mobile

Dataset distribution	MNIST	FMNIST
non-federated IID (baseline)	89.03 $\pm$0.33	65.27 $\pm$0.52
federated IID (baseline)	87.09 $\pm$0.17	62.11 $\pm$0.27
federated non-IID ($\alpha = 100$)	87.48 $\pm$0.52	63.41 $\pm$0.13
federated non-IID ($\alpha = 10$)	86.77 $\pm$0.32	63.83 $\pm$0.4
federated non-IID ($\alpha = 1$)	86.87 $\pm$0.56	63.9 $\pm$0.28
federated non-IID ($\alpha = 0.1$)	84.38 $\pm$0.3	62.85 $\pm$0.09
federated non-IID ($\alpha = 0.01$)	60.65 $\pm$1.16	48.71 $\pm$0.47

Table 1: CAS of CVAE after 20 communication rounds for different dataset distributions.

devices or IoT. For determining the performance of the downstream model, we use image classification accuracy. The accuracy is depicted as the weighted average of each client in the FL, where the weights are determined by the ratio of the data quantity distributed to each client.

3 Experiment Results

3.1 Performance of Generative Models

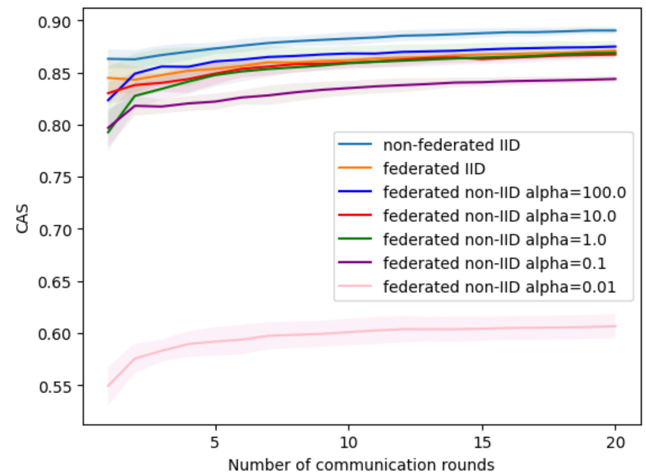
Training the CVAE federatively was done over 20 communication rounds, 10 clients, and two local epochs per round. The hyperparameters of the CVAE include a batch size of 32 and a learning rate of 0.001.

For each data distribution setting, the experiment was carried out over five trials. The mean and standard deviation of the CAS results can be seen in table 1. In addition, figure 6 displays the CAS per communication round. It shows that it doesn't take many communication rounds for the CVAE to converge to its performance upper bound.

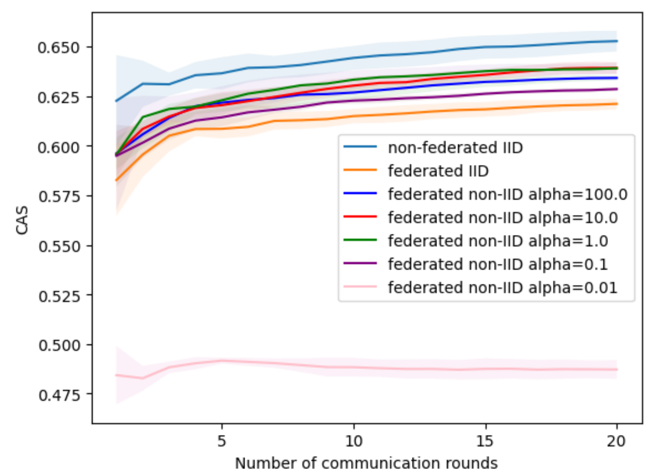
3.2 Performance of Downstream Models

Training the ExquisiteNetV1 federatively was done over 10 communication rounds, 10 clients, and two local epochs per round. The hyperparameters of the ExquisiteNetV1 include a batch size of 32 and a learning rate of 0.001. For baseline comparison, each data distribution setting involved experiments comparing the ExquisiteNetV1 without pre-training on synthetic data against ExquisiteNetV1 with pre-training on synthetic data. Pre-training the ExquisiteNetV1 with synthetic data was done over five epochs before the federated learning process began. It should be noted that for baseline comparison, we used CVAEs with corresponding Dirichlet α values to generate the synthetic data. This is to simulate the real-life federated environment where the data distribution of each client remains the same when training the generative model and training the downstream model.

For each data distribution setting, the experiment was carried out over five trials. As mentioned in 2.2, the accuracy is depicted as the weighted average of the clients. The results of the ExquisiteNetV1 can be seen in table 2. Figure 7 shows the accuracy average and its standard deviation over time throughout the communication rounds for ExquisiteNetV1 pre-trained with synthetic data.



(a) MNIST

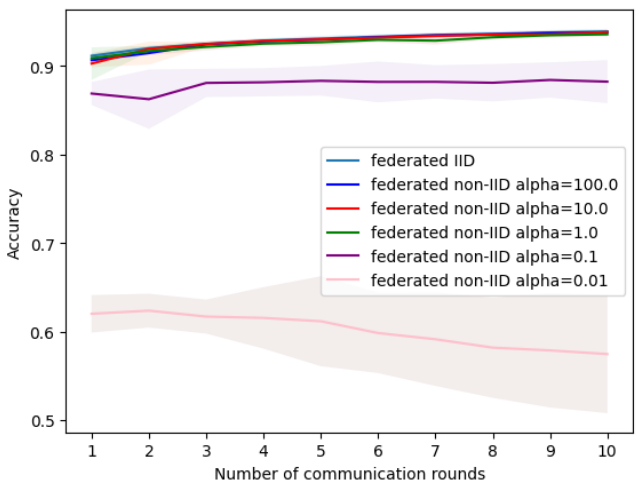


(b) FashionMNIST

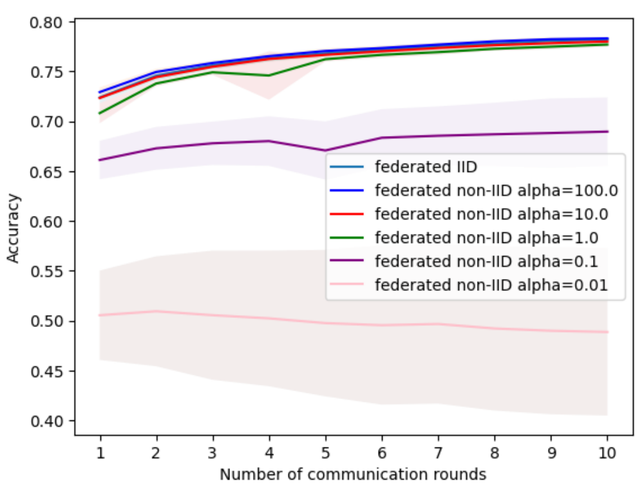
Figure 6: Mean and standard deviation of CAS over 20 communication rounds of various distribution settings for (a) MNIST and (b) FashionMNIST

Dataset distribution	MNIST		FashionMNIST	
	Without pre-training	With pre-training	Without pre-training	With pre-training
federated IID	87.08 $\pm$1.03	93.92 $\pm$0.28	71.35 $\pm$1.02	78.21 $\pm$0.24
federated non-IID ($\alpha = 100$)	83.78 $\pm$5.32	93.80 $\pm$0.33	70.26 $\pm$1.93	78.32 $\pm$0.13
federated non-IID ($\alpha = 10$)	81.70 $\pm$8.91	93.85 $\pm$0.16	70.23 $\pm$2.55	78.0 $\pm$0.21
federated non-IID ($\alpha = 1$)	85.87 $\pm$2.87	93.59 $\pm$0.26	69.48 $\pm$2.64	77.7 $\pm$0.21
federated non-IID ($\alpha = 0.1$)	41.35 $\pm$15.28	88.24 $\pm$2.43	56.94 $\pm$4.32	68.96 $\pm$3.42
federated non-IID ($\alpha = 0.01$)	22.52 $\pm$5.24	57.45 $\pm$6.67	30.9 $\pm$4.9	48.85 $\pm$8.39

Table 2: Mean and standard deviation of ExquisiteNetV1 accuracy after 10 communication rounds for different dataset distributions.

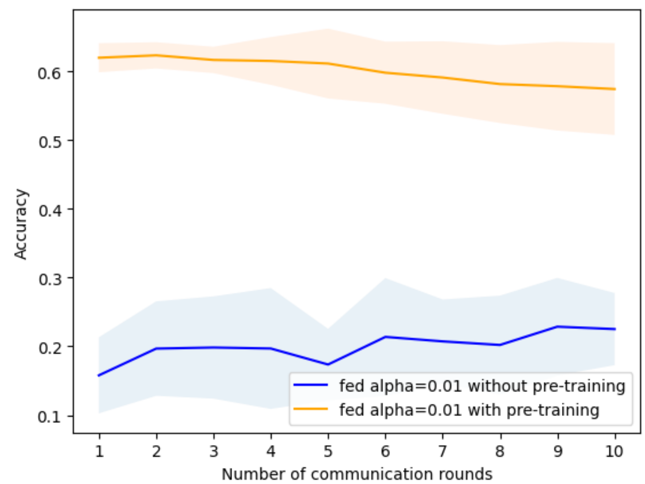


(a) MNIST

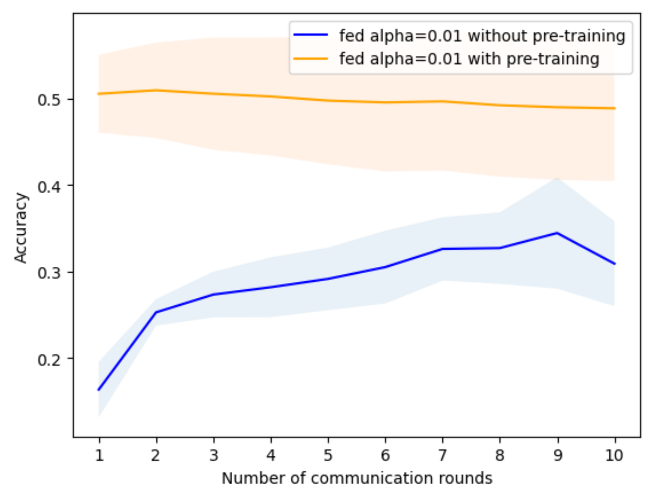


(b) FashionMNIST

Figure 7: Mean and standard deviation of pre-trained ExquisiteNetV1 accuracy over 10 communication rounds of various distribution settings on (a) MNIST and (b) FashionMNIST

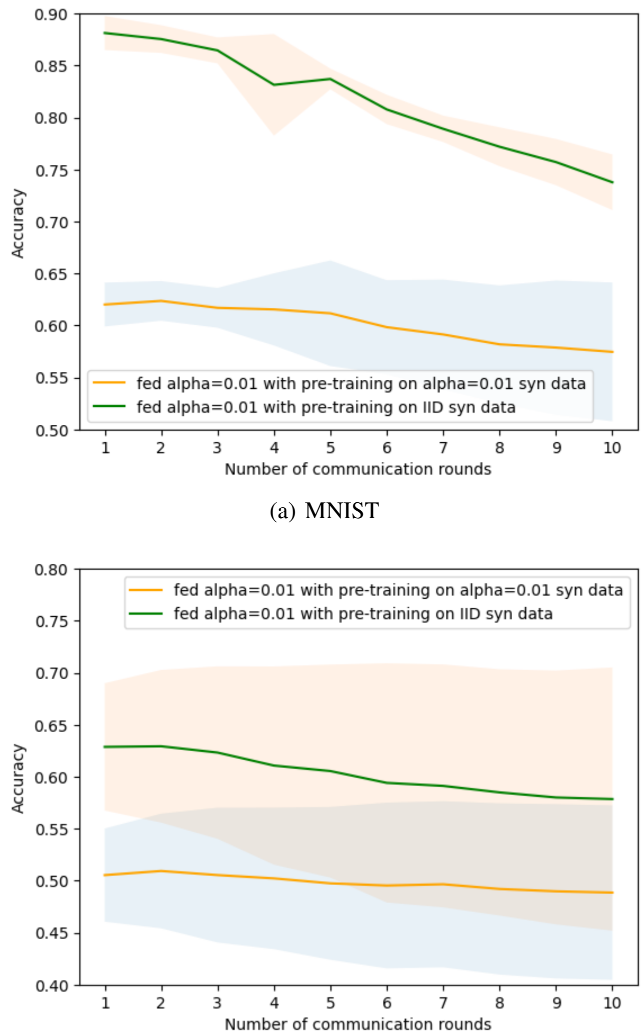


(a) MNIST



(b) FashionMNIST

Figure 8: Mean and standard deviation of ExquisiteNetV1 accuracy over 10 communication rounds of non-IID Dirichlet $\alpha = 0.01$ with and without pre-training



(a) MNIST

(b) FashionMNIST

Figure 9: Mean and standard deviation of ExquisiteNetV1 accuracy over 10 communication rounds of non-IID Dirichlet $\alpha = 0.01$ with pre-training on same non-IID synthetic data against IID synthetic data.

One notable behaviour from figure 7 is when the Dirichlet $\alpha = 0.01$, the accuracy actually decreases over time. This is not the case when ExquisiteNetV1 goes through the FL process with the same degree of imbalance but without pre-training on synthetic data, as depicted in 8. To determine whether pre-training the ExquisiteNetV1 with better synthetic data can mitigate this behaviour, an additional experiment was carried out where Dirichlet $\alpha = 0.01$ but the ExquisiteNetV1 was pre-trained with synthetic data from a CVAE trained on IID data. Figure 9 compares the performance between pre-training on $\alpha = 0.01$ synthetic data against IID synthetic data for the $\alpha = 0.01$ ExquisiteNetV1. The observed behaviour is the same, where the accuracy of the ExquisiteNetV1 pre-trained with IID synthetic data also decreases over time.

4 Discussion of Experiment

4.1 Analysis of Experiment Results

As the data imbalance increases, the overall CAS of the CVAE decreases as depicted in table 1. This drop in performance was not noticeable, however, except for the most extreme data imbalance when the Dirichlet $\alpha = 0.01$. This could be attributed to the fact that when the data imbalance is high to such a degree, each client in the FL process is getting datasets with at most two classes, possibly even one class, which leads to extreme class imbalance. In addition, the imbalance in the quantity of the data per client can be extreme as well. Given our experiment setup where there are 10 clients in the FL process, one client could be assigned as much as 20% of the entire dataset while another could be assigned as less as 0.1% when the Dirichlet $\alpha = 0.01$. This demonstrates that high degrees of data imbalance can have a noticeable impact on the performance of the CVAE.

Table 2 show that across all data distribution settings, pre-training ExquisiteNetV1 with synthetic data before the FL process has a noticeable impact compared to results that didn't incorporate pre-training. This observation is consistent with the conclusions from [5]. One observation to note is the difference in accuracy with pre-trained ExquisiteNetV1 right from the start (communication round #1) of the FL process. This is particularly apparent between $\alpha = 0.1$ and $\alpha = 0.01$ as shown in figure 7. This can be explained by the difference in the synthetic data quality each ExquisiteNetV1 was pre-trained on. As shown in table 1, there is a noticeable drop in synthetic data quality when $\alpha = 0.01$. Due to pre-training on lower-quality synthetic data, the ExquisiteNetV1 under $\alpha = 0.01$ setting would have a lower accuracy compared to the ExquisiteNetV1 under the $\alpha = 0.1$ setting. This is further supported by figure 9 where there is a noticeable starting accuracy difference in the ExquisiteNetV1 under $\alpha = 0.01$ setting when it is pre-trained on $\alpha = 0.01$ synthetic data compared to IID synthetic data.

There are, however, apparent limitations of GenFL as the degree of data imbalance increases. As shown in figures 8 and 9, there is a noticeable decrease in accuracy over time when data imbalance is extreme ($\alpha = 0.01$), regardless of the quality of the synthetic data the ExquisiteNetV1 was pre-trained on. One possible explanation for this is due to the extreme class imbalance and quantity imbalance in the client dataset as mentioned previously. As a result, local training would result in extremely biased local weights towards its respective local majority class(es). Since our GenFL process doesn't incorporate any weight adjustments to mitigate class imbalances and simply averages all the clients' weights (i.e. FedAVG), it suggests that the initialized weights of the pre-trained ExquisiteNetV1 will degrade over time with such extreme data imbalance. This degradation would accumulate for each communication round, thus leading to a performance decrease.

An interesting behaviour to note when ExquisiteNetV1 is pre-trained with synthetic data is the increase in its accuracy standard deviation in two cases: (1) as α decreases as depicted in table 2 and (2) as the FL communication rounds progress for the ExquisiteNetV1 under $\alpha = 0.01$ setting as depicted in figures 7, 8, and 9. These cases can be attributed to a similar explanation as mentioned previously: as class imbalance and quantity imbalance increase, the bias in local weights towards its respective local majority class(es) also increases. Without any mitigation process for such bias during the aggregation step, it accumulates over each communication round and results in a more unstable ExquisiteNetV1.

4.2 Experiment Limitations and Future Work

Generative Model and Datasets

For this research, we only used a VAE model, specifically a CVAE. This was sufficient for this research as it used MNIST and FashionMNIST, which are considered relatively simple image datasets. Therefore, we can not make any definitive relations from the conclusion of this research to other research settings that used more advanced generative models, such as GANs or Diffusion models, and complex datasets. Moreover, we only considered image data but tabular data is another common data type used in FL research [4]. Despite the limited experiment settings, one could expect similar results when using GenFL under different settings. [5] has different experiment settings but methods similar to GenFL to increase the performance of its downstream model in FL.

Data Imbalance

While there are several types of non-IID [8], this paper only considered label distribution imbalance and quantity imbalance in the data. Other types of non-IID, including feature imbalance, are prevalent in the real world, thus future research should look into the effect GenFL can have on other types of data imbalances. Specifically, future research can explore non-IID challenges using the NIID benchmark [8].

Differential Privacy

Introducing generative models raises privacy issues. There are known methods that uses generated synthetic data to extract the real data the generative models were trained on [18]. This is a crucial issue as ignoring data privacy defeats one of the main purposes of FL. There are ways to introduce differential privacy, such as adding noise during the generating process [9], which has been shown to increase privacy but at the cost of some performance. Due to limited resources, we do not consider differential privacy in our experiment but it is something extensions of this research should take into account.

Specific Domain

This research setup is focused mainly on the medical domain but FL also includes the IoT and the cloud domain. These domains have thousands, potentially millions of clients participating in the learning process with additional complexities, such as communication latency and the number of participants in the aggregation step. The idea of pre-training the downstream model with synthetic images before starting the federated process in such a setting can also be explored in future research.

5 Conclusion to GenFL

FL is a keen interest in the machine learning community due to its ability to enable distributed machine learning where centralizing the training data is not an option. However, non-IID data can severely affect the performance of the model. In this research, we proposed a new FL approach called "GenFL", which pre-trains the downstream model with synthetic data from a generative model that follows the collective distribution of all the clients' datasets. Our results show that GenFL can be an effective approach to bridging the performance gap that occurs when data is imbalanced amongst the clients. However, this performance increase is affected by the degree of data imbalance. Our research showed that as data imbalance increased, it resulted in less performance gains and higher instability in the downstream model. In addition, GenFL was not able to bridge the performance gap for extreme data imbalance settings, even showing decreasing performance as the training progressed. Despite the limited scope of our experiment, our results and related works mentioned in this paper show that there is potential in GenFL and can be a guide to more effective approaches to solving non-IID challenges in FL.

6 Responsible Research

We declare that there are no competing financial interests or personal relationships that could have influenced the work in this report. The code used in the experiments and resulting experiment data are publicly available on GitHub. Moreover, there were no human subjects involved in the experiment, thus the research did not require any ethical approvals. As mentioned in 4.2, introducing generative models raises privacy issues as the generated synthetic data can be used to reconstruct the real training data. However, all data used in this research are publicly available datasets specifically for machine learning research purposes. Therefore, there are no data privacy issues that arise from the context of the experiments carried out in this research.

References

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y. Arcas, "Federated learning of deep networks using model averaging," 2016. DOI: 10.48550/ARXIV.1602.05629.

[2] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. DOI: 10.48550/ARXIV.1806.00582.

[3] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Generation Computer Systems*, vol. 135, pp. 244–258, Oct. 2022, ISSN: 0167-739X. DOI: 10.1016/j.future.2022.05.003.

[4] C. Little, M. Elliot, and R. Allmendinger, "Federated learning for generating synthetic data: A scoping review," *International Journal of Population Data Science*, vol. 8, no. 1, Oct. 2023, ISSN: 2399-4908. DOI: 10.23889/ijpds.v8i1.2158.

[5] H.-Y. Chen, C.-H. Tu, Z. Li, H.-W. Shen, and W.-L. Chao, *On the importance and applicability of pre-training for federated learning*, 2022. DOI: 10.48550/ARXIV.2206.11488.

[6] M. Morafah, M. Reisser, B. Lin, and C. Louizos, *Stable diffusion-based data augmentation for federated learning with non-iid data*, 2024. DOI: 10.48550/ARXIV.2405.07925.

[7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. DOI: 10.48550/ARXIV.1806.00582.

[8] Q. Li, Y. Diao, Q. Chen, and B. He, *Federated learning on non-iid data silos: An experimental study*, 2021. DOI: 10.48550/ARXIV.2102.02079.

[9] B. Xin, Y. Geng, T. Hu, *et al.*, "Federated synthetic data generation with differential privacy," *Neurocomputing*, vol. 468, pp. 1–10, Jan. 2022, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.10.027.

[10] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016. DOI: 10.48550/ARXIV.1602.05629.

[11] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, T. N. Hoang, and Y. Khazaeni, *Bayesian nonparametric federated learning of neural networks*, 2019. DOI: 10.48550/ARXIV.1905.12022.

[12] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2013. DOI: 10.48550/ARXIV.1312.6114.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. DOI: 10.48550/ARXIV.1406.2661.

[14] A. Singh and T. Ogunfunmi, "An overview of variational autoencoders for source separation, finance, and bio-signal applications," *Entropy*, vol. 24, no. 1, p. 55, Dec. 2021, ISSN: 1099-4300. DOI: 10.3390/e24010055.

[15] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[16] S. Ravuri and O. Vinyals, *Classification accuracy score for conditional generative models*, 2019. DOI: 10.48550/ARXIV.1905.10887.

[17] S.-Y. Zhou and C.-Y. Su, "An efficient and small convolutional neural network for pest recognition – exquisitenet," 2021. DOI: 10.48550/ARXIV.2107.07167.

[18] B. Hitaj, G. Ateniese, and F. Perez-Cruz, *Deep models under the gan: Information leakage from collaborative deep learning*, 2017. DOI: 10.48550/ARXIV.1702.07464.