Aprendizado Federado com Geração de *Embeddings* para Controle da Heterogeneidade Estatística

Gustavo S. Guaragna[1], **Joahannes B. D. da Costa**[2], **Leandro A. Villas**[1], **Allan M. de Souza**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6176 – 13083-970 – Campinas – SP – Brasil

[2]Universidade Federal de São Paulo (UNIFESP) – São José dos Campos, SP, Brasil

g198603@dac.unicamp.br, joahannes.costa@unifesp.br,
{lvillas, allanms}@unicamp.br

Abstract. *Federated Learning enables collaborative training of machine learning models without sharing local data, addressing growing concerns over data privacy. However, heterogeneous data distributions across clients remain a major challenge, often degrading model performance. In this paper, we propose FLEG, a novel approach that alternates classifier training with the training of a Conditional Generative Adversarial Network (CGAN) to augment client datasets, mitigating statistical heterogeneity and, consequently, improving classification performance. Unlike prior methods, FLEG generates synthetic embeddings instead of images, adding an extra layer of protection against data leakage. Experimental results show that FLEG outperforms the FedAvg baseline by up to 14 percentage points in validation accuracy on CIFAR-10 under the evaluated settings. The code is available at https://github.com/gustavoguaragna/FLEG.*

Resumo. *O Aprendizado Federado permite o treinamento colaborativo de modelos de aprendizado de máquina sem o compartilhamento de dados locais, sendo uma alternativa promissora diante de crescentes preocupações com a privacidade. Contudo, a heterogeneidade na distribuição dos dados entre os clientes permanece um dos principais desafios, afetando negativamente o desempenho dos modelos. Neste trabalho, propomos o FLEG, uma abordagem que alterna o treinamento de um classificador com o de uma Rede Adversária Generativa Condicional (CGAN) para aumentar os conjuntos de dados dos clientes e reduzir a heterogeneidade estatística da federação e, consequentemente, melhorar o desempenho do modelo classificador. Diferentemente de abordagens convencionais, o FLEG gera embeddings sintéticos em vez de imagens, adicionando uma camada extra de proteção a possíveis vazamentos de dados. Os resultados experimentais mostram que o FLEG supera a baseline FedAvg em até 14 pontos percentuais na acurácia de validação no conjunto CIFAR-10, nas configurações avaliadas. O código está disponível em https://github.com/gustavoguaragna/FLEG.*

1. Introdução

Modelos de Inteligência Artificial (IA) têm alcançado resultados expressivos em diversos domínios como segurança, mobilidade urbana, saúde e finanças. Entretanto, muitos

desses modelos requerem grandes volumes de dados para atingir um desempenho satisfatório, sendo particularmente evidente em tarefas complexas baseadas em técnicas de Aprendizado Profundo (*Deep Learning* - DL), como a classificação de imagens [Ahmed et al. 2023, Alzubaidi et al. 2021]. Dessa forma, uma única instituição (como uma empresa ou hospital) muitas vezes não possui dados suficientes, em quantidade e variedade, para treinar modelos de IA robustos para uma certa tarefa. Além disso, leis e normas éticas de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil e o Regulamento Geral de Proteção de Dados (GDPR) na Europa, podem impedir que diferentes instituições compartilhem informações entre si, mesmo quando isso poderia beneficiar o desenvolvimento de soluções mais eficazes.

O Aprendizado Federado (*Federated Learning* – FL) surge como uma alternativa promissora para mitigar essa limitação, pois permite o treinamento colaborativo de modelos sem a centralização dos dados brutos [McMahan et al. 2016]. No paradigma tradicional de FL, um servidor central distribui um modelo global aos clientes, que realizam o treinamento localmente e retornam os parâmetros atualizados para agregação. Entretanto, a presença de dados não Independentes e Identicamente Distribuídos (não-IID) é um dos principais desafios no FL [**?**, Zhu et al. 2021]. À medida que a heterogeneidade estatística (*i.e.*, dados não-IID) aumenta, os modelos locais tendem a divergir do ótimo global, dificultando a convergência do processo de agregação e degradando o desempenho final do modelo global [Zhao et al. 2018]. Diversas estratégias são propostas para endereçar esse problema. Uma delas é a utilização de termos regularizadores na função de perda ou nos mecanismos de agregação dos modelos, a fim de controlar as atualizações dos clientes [Acar et al. 2021, Li et al. 2020, Karimireddy et al. 2021].

Uma linha de pesquisa complementar explora soluções que atuam diretamente nos dados, buscando reduzir a heterogeneidade estatística da federação. Essa é uma abordagem promissora, uma vez que atua na raiz do problema dos dados não-IID, embora possa introduzir riscos à privacidade. Alguns trabalhos, por exemplo, sugerem permitir o compartilhamento de uma seleta quantidade de dados para ganhos significativos na acurácia do modelo [Zhao et al. 2018, Yoshida et al. 2020]. Outras soluções utilizam modelos generativos para ampliar os conjuntos de dados locais por meio da geração e compartilhamento de dados sintéticos [Pan et al. 2025, Guaragna et al. 2025, Huangsuwan et al. 2025, Maliakel et al. 2024, Jeong et al. 2018]. Embora eficazes, essas abordagens podem introduzir riscos à privacidade, sobretudo quando os dados gerados preservam características sensíveis dos dados originais.

Diante desse cenário, este trabalho propõe o FLEG (*Federated Learning with Embedding Generation*), uma estratégia de treinamento federado que organiza o processo de aprendizado em múltiplos níveis e treina modelos generativos para a geração de *embeddings* (representações vetoriais), que são incorporados ao conjunto de dados dos clientes, visando a redução da heterogeneidade estatística da federação. Diferentemente de abordagens anteriores, o FLEG alterna o treinamento de um classificador federado com o de um modelo generativo, operando no espaço de *embeddings* em vez do espaço original dos dados. Dessa forma, apenas representações latentes sintéticas são compartilhadas entre clientes e servidor, reduzindo a exposição de informações sensíveis sem abrir mão dos benefícios do balanceamento de dados.

Essa estratégia promove uma adaptação progressiva do modelo global, permi-

tindo que o classificador incorpore informações complementares ao longo das rodadas de comunicação, enquanto controla a divergência causada por distribuições de dados desbalanceadas. O FLEG foi avaliado em cenários de classificação de imagens amplamente utilizados na literatura, demonstrando ganhos consistentes de desempenho em ambientes federados heterogêneos. Os resultados mostram que FLEG amplia a capacidade dos modelos treinados com as *baselines* utilizados na maioria dos cenários avaliados, aumentando a acurácia em até 14 pontos percentuais.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados. A Seção 3 descreve em detalhes o método proposto. A Seção 4 apresenta a avaliação experimental. A Seção 5 discute limitações e trabalhos futuros. Por fim, a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

Esta seção apresenta soluções baseadas em dados que buscam diminuir a divergência dos parâmetros dos modelos locais em FL sob heterogeneidade estatística.

Uma linha de pesquisa busca balancear os dados dos clientes por meio de técnicas de aumento de dados (*data augmentation*). Em Astrea [Duan et al. 2020], os clientes compartilham suas distribuições de dados, para então, balancear seus dados com *under-sampling* em classes majoritárias e técnicas clássicas de *data augmentation* em visão computacional como cortes, *zoom* e rotações em imagens para classes minoritárias. Já, FedHome [Wu et al. 2020] utiliza SMOTE [Chawla et al. 2002] para gerar amostras sintéticas a partir de vizinhos próximos. Essas soluções, embora úteis, tornam-se limitadas quando classes inteiras estão ausentes em determinados clientes.

Outros trabalhos utilizam modelos generativos para produzir dados sintéticos realistas e reduzir a heterogeneidade estatística da federação. Por exemplo, FIMI [Li et al. 2024a] pré-treina um modelo de difusão em dados públicos para aumentar os conjuntos locais. Já, argumentando que modelos de difusão podem ser computacionalmente caros para serem treinados e terem seu processo de inferência lento, o trabalho em [Pan et al. 2025] treina uma GAN por classe de forma centralizada e distribui as imagens sintéticas de maneira uniforme entre os clientes, que possuem imagens reais de câncer de mama. Porém, um possível limitante dessas abordagens é a dificuldade de encontrar dados públicos que possam ser efetivos para melhorar tarefas em dados privados.

Há também abordagens que treinam modelos generativos diretamente no ambiente federado. FedDrip [Huangsuwan et al. 2025] treina um modelo de difusão via FedAvg [McMahan et al. 2016] (algoritmo tradicional de FL, no qual os modelos locais são agregados por média ponderada) e o utiliza para gerar imagens que permanecem no servidor para compor um pseudo-cliente que também participa da federação. Já, o FLIGAN [Maliakel et al. 2024] agrupa clientes de acordo com suas distribuições e treina GANs federadas específicas por classe, priorizando clientes com maior representatividade. Apesar dos ganhos, treinar uma GAN por classe pode ser computacionalmente custoso.

Outra solução relevante é FedER [Pennisi et al. 2023], que adota uma arquitetura *peer-to-peer*, ou seja, não há um servidor central na federação. Em FedER, cada nó treina localmente uma CGAN, utilizando um termo penalizador na função de perda que reduz riscos de vazamento de dados privados. Em seguida, compartilha seu modelo

classificador com outro nó aleatório, juntamente com um *buffer* de dados sintéticos gerados pela CGAN, promovendo aprendizado colaborativo. Trabalhos adicionais também pré-treinam modelos generativos para então usá-los no aumento de dados locais em FL [Li et al. 2022, Zhao et al. 2023]. Já, FedGenIA [Guaragna et al. 2025] treina uma CGAN nas mesmas rodadas em que também treina um modelo classificador, utilizando uma estratégia incremental de aumento de dados ao longo das rodadas. Contudo, a imputação dos dados não direciona as classes específicas que cada cliente necessita, reduzindo a velocidade de homogeneização.

Embora esses trabalhos reduzam a necessidade de centralização dos dados, muitos ainda dependem do compartilhamento de dados sintéticos ou de modelos generativos completos, o que pode expor informações sensíveis. Para mitigar esse risco, HFMDS [Li et al. 2024b] propõe a geração de dados sintéticos no espaço latente. De forma semelhante, outros trabalhos aproveitam o conhecimento de modelos fundacionais para treinar CVAEs (*Variational Auto-Encoders*) para gerar *embeddings*, que possam ser utilizados em tarefas posteriores. [Salvo et al. 2024, Salvo et al. 2025].

Já, a solução proposta neste artigo alterna o treinamento do classificador com o de uma CGAN para gerar *embeddings* representativos de imagens reais. Diferentemente de abordagens baseadas em modelos fundacionais, o FLEG organiza o treinamento em níveis, permitindo que os *embeddings* incorporem informações aprendidas pelo classificador no respectivo nível. Assim, o FLEG não depende de modelos externos, adquirindo conhecimento específico da tarefa de forma progressiva. Além disso, evita o compartilhamento de dados sintéticos em sua forma original, adicionando uma camada extra de proteção a vazamentos de dados. Os detalhes são apresentados na Seção 3.

3. FLEG

Esta seção descreve o funcionamento do FLEG. Inicialmente, na Subseção 3.1 é introduzido o processo de treinamento das Redes Adversárias Generativas Condicionais (CGANs) e apresentado a definição geral do FLEG com as variáveis utilizadas. Em seguida, o fluxo específico completo do método é detalhado na Subseção 3.2.

3.1. Definição Geral

O FLEG busca reduzir a heterogeneidade estatística em aprendizado federado alternando o treinamento do modelo classificador com o de um modelo generativo, especificamente uma CGAN [Mirza and Osindero 2014], conhecida por ter capacidade de geração de dados sintéticos mais fidedignos aos reais do que autoencoders variacionais (*Variational Autoencoders* - VAE) e terem um processo de treinamento e inferência menos custoso do que de um modelo de difusão. Em uma GAN [Goodfellow et al. 2014], uma rede geradora e uma discriminadora são treinadas de forma adversária: a discriminadora tenta distinguir dados reais de sintéticos, enquanto a geradora aprende a produzir amostras realistas o suficiente para enganar a discriminadora. Na CGAN, esse processo é condicionado a rótulos de classe, concatenados às entradas de ambas as redes.

No FLEG, o treinamento ocorre em níveis sucessivos (iniciando no nível 0). Em cada nível, treina-se primeiramente o classificador e, em seguida, a CGAN, ambos de forma federada. Considerando um classificador com L camadas, no nível i treinam-se as últimas $L - i$ camadas, denominadas *classificadora*, enquanto as i primeiras camadas

atuam como *extratora* de *embeddings*. O treinamento da classificadora é interrompido com base em um parâmetro de paciência p, definido como o número de épocas sem melhoria na acurácia de validação. (É importante ressaltar que utilizar a acurácia de validação como critério de parada para o treinamento federado não é algo prático, uma vez que não há um banco de dados de validação no servidor. Porém o objetivo aqui é avaliar a capacidade de treinamento da classificadora em cada nível, a fim de verificar se, realmente, a adição de embeddings sintéticos promove melhorias na acurácia e não por simplesmente treinar por mais rodadas. Em uma solução prática, poderia ser utilizado um número fixo de rodadas ou até mesmo a acurácia média de validação dos clientes).

Após o término do treinamento da classificadora em um nível, inicia-se o treinamento da CGAN utilizando a estratégia F2U [Yonetani et al. 2019], na qual cada cliente treina uma discriminadora local, enquanto uma geradora global é treinada no servidor, utilizando a discriminadora que retorna a maior probabilidade de um respectivo dado sintético ser real. Para melhorar a estabilidade do treinamento adversário da CGAN com F2U, o conjunto de dados é dividido em C *chunks*, conforme observado em FedGenIA [Guaragna et al. 2025]. Para reduzir custos de comunicação, apenas os *embeddings* sintéticos gerados são compartilhados com os clientes. O treinamento da CGAN é limitado a E épocas, ao final das quais a geradora produz S *embeddings* sintéticos no formato da camada $i + 1$ do classificador, que serão incorporados pelos clientes no próximo nível.

3.2. Fluxo Específico Completo de FLEG

O processo inicia no nível 0, no qual o classificador completo (com todas as $L = 5$ camadas) é treinado de forma federada, com agregação via FedAvg [McMahan et al. 2016] e paciência $p = 10$, definida heuristicamente. Em seguida, a CGAN é treinada para gerar *embeddings* correspondentes à primeira camada do classificador. Para isso, a camada extratora é compartilhada com os clientes, permitindo a transformação das imagens locais em *embeddings* compatíveis com o treinamento da CGAN. Após receber *embeddings* gerados pela geradora, cada cliente divide seus dados em C *chunks*, treina sua discriminadora local com os *embeddings* reais de um dos *chunks* e os sintéticos recebidos, e envia o modelo ao servidor, que atualiza a geradora global com 20 passos de otimização, utilizando uma única amostra sintética por passo, configuração que proporcionou melhor equilíbrio entre o treinamento da discriminadora e da geradora da CGAN.

O uso de diferentes valores para o número de *chunks* C é apresentado na Seção 4. O número de épocas da CGAN E foi definido heuristicamente como 25, mas outros valores também foram avaliados conforme C, visando equilibrar os custos de comunicação e computação. Após as E épocas, os *embeddings* sintéticos gerados são utilizados, junto aos dados reais, apenas no treinamento da classificadora do próximo nível. O número S de amostras geradas por nível pode ser definido de duas formas: "*Fixed*", no qual S é constante e suficientemente grande para garantir probabilisticamente a homogeneização das classes entre os clientes da federação; ou "*Dynamic*", no qual S cresce ao longo dos níveis, atingindo no último nível um valor grande o suficiente para a homogeneização. O efeito desses modos é analisado na Seção 4.

A partir do nível 1, os clientes passam a incorporar os *embeddings* sintéticos gerados no nível anterior. Para acelerar a homogeneização dos dados e a convergência do classificador, esses *embeddings* são filtrados: cada cliente contabiliza suas amostras por classe

e define quantas amostras sintéticas necessita, com base no esperado em uma distribuição uniforme. Assim, para cada classe, o número de *embeddings* solicitados por um cliente corresponde à diferença entre a quantidade esperada e a disponível localmente. Para compatibilidade, os dados reais também são transformados em *embeddings* pela extratora. O classificador, reduzido às 4 camadas mais profundas, é então treinado novamente, e o processo se repete com as devidas adaptações das redes até o nível 5, no qual a classificadora se resume à camada de saída e não há treinamento da CGAN.

Assim, em FLEG, o treinamento é alternado entre modelo classificador e generativo e dividido em níveis. A cada nível que se avança, a classificadora é reduzida para conter somente as camadas mais profundas (mais próximas da camada de saída), enquanto a GAN gera *embeddings* com dimensões iguais ao vetor de entrada da classificadora do nível subsequente. Dessa maneira, a classificadora pode melhorar seu poder discriminativo a cada nível ao incorporar dados sintéticos em representação de *embeddings* que reduzam a divergência nos pesos das classificadoras de diferentes clientes. O processo completo é apresentado na Figura 1.

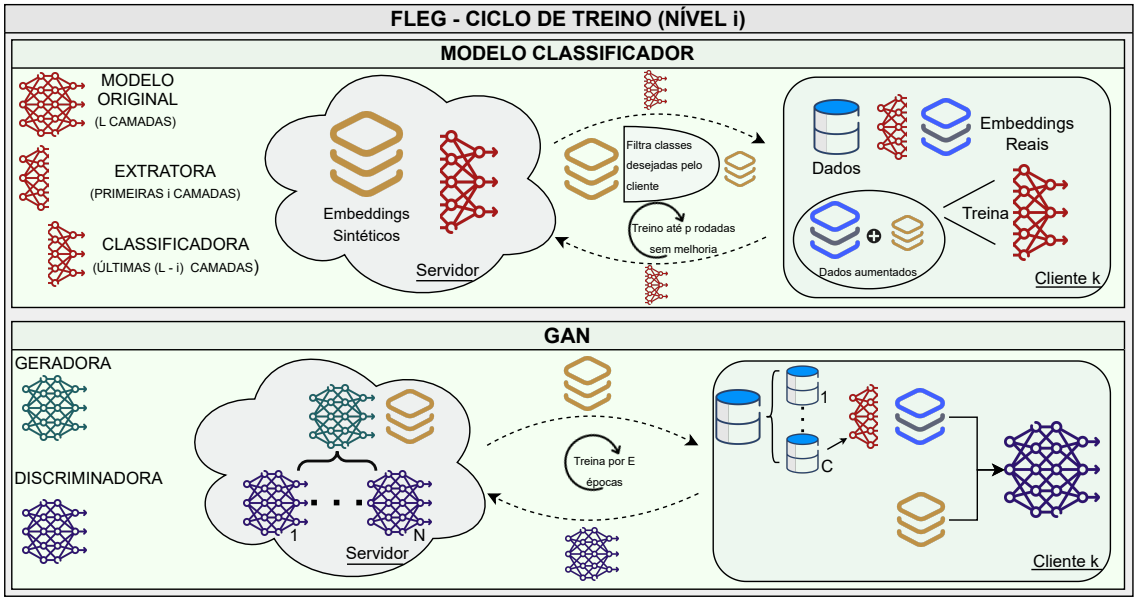

Figura 1. Fluxo de Treinamento em FLEG

4. Avaliação de Desempenho

Nesta seção, o desempenho do FLEG é analisado por meio de figuras e tabelas. Inicialmente, descrevemos os experimentos e apresentamos um panorama dos resultados na Subseção 4.1. Em seguida, avaliamos o FLEG em termos da acurácia de validação ao longo das épocas de treinamento na Subseção 4.2, bem como sua relação com o custo de comunicação na Subseção 4.3 e com o custo computacional na Subseção 4.4.

4.1. Experimentos

Para avaliar o desempenho de FLEG, utilizamos os conjuntos de dados MNIST [LeCun et al. 2002] e CIFAR10 [Krizhevsky et al. 2009], amplamente utilizados em tarefas de visão computacional. Visando um cenário de aprendizado federado *cross-silo*, no

qual os clientes são grandes instituições, os experimentos foram conduzidos com quatro clientes. A fim de analisar a robustez do método proposto em federações com diferentes níveis de heterogeneidade estatística nos rótulos das classes, avaliamos três diferentes cenários, seguindo a mesma metodologia adotada em FedGenIA [Guaragna et al. 2025]:

- **ClassPartition**: corresponde ao cenário de maior heterogeneidade avaliado. Nesse caso, a divisão dos dados é realizada de forma que cada cliente receba amostras de classes específicas e exclusivas, não havendo sobreposição de classes entre os clientes.
- **Dir01**: representa um nível elevado de heterogeneidade estatística, embora menos extremo que o ClassPartition. A partição dos dados é realizada por meio de uma distribuição de Dirichlet, controlada pelo parâmetro α, sendo que valores menores resultam em distribuições mais desbalanceadas. No cenário Dir01, α é definido como 0.1.
- **Dir05**: caracteriza um nível moderado de heterogeneidade estatística, sendo o cenário mais suave avaliado. Assim como em Dir01, a partição segue uma distribuição de Dirichlet, porém com o parâmetro α definido como 0.5.

Em todos os experimentos do FLEG, fixamos o número de clientes em $K = 4$ e o número de níveis — e, portanto, de camadas do classificador — em $L = 5$ (0 a 4). Para o classificador, adotamos paciência $p = 10$ e uma época local por rodada. Para a CGAN, o número de *chunks* C variou entre 1, 10, 50, 100 e 200. Também avaliamos diferentes números de épocas E, conforme C, visando equilibrar o custo computacional. Para cada C, foi avaliado $E = 25$ e, para valores menores de C, avaliamos também maiores número de épocas: para C definido como 1, 10, 50 e 200, avaliamos também E como 40, 35, 30 e 20, respectivamente, sendo que para $C = 100$, foram testados também E como 20 e 30. Por fim, o número de dados sintéticos por nível foi definido pelos modos *Fixed* ou *Dynamic*. Para *Fixed*, utilizamos $S = D$, sendo D o total de amostras de todos os clientes juntos. Já no modo *Dynamic*, definimos $S_i = \left\lfloor \left\lceil \frac{D}{K} \right\rceil \frac{(i+1)}{L} \right\rfloor$ para cada nível i.

Para cada combinação de parâmetros — número de *chunks* C, épocas da GAN E e modo de geração dos dados sintéticos S — realizamos três execuções independentes (*trials*). A Tabela 1 apresenta algumas dessas combinações ('C', 'E' e 'Modo S'), juntamente com a acurácia máxima no conjunto de validação ('Acurácia Máx.'), o tempo total de execução em minutos ('Tempo (min)') e a quantidade de informação a ser transmitida na rede ('Tráf. (GB)') para cada conjunto de dados ('Dataset') e cenário de heterogeneidade ('Partição'). Os experimentos foram realizados em uma única GPU *NVIDIA Quadro RTX 6000*, sem simulação federada; assim, os clientes são serializados, inflando o tempo computacional, mas desconsiderando o tempo de comunicação, já que clientes e servidor compartilham a mesma máquina. Todos os resultados reportados correspondem ao *trial* que obteve a mediana entre as acurácias máximas de cada experimento.

As combinações específicas selecionadas e apresentadas na Tabela 1 foram feitas de acordo com os resultados obtidos, de forma a considerar a vantagem e a desvantagem de FLEG, ou seja, a melhora da acurácia, mas com o aumento dos custos computacional e de comunicação. Rotulamos essas combinações da seguinte forma:

- **FLEG Full**: retrata a combinação que resultou na maior acurácia em todo o experimento, revelando a capacidade máxima do FLEG. É interessante notar que para

a maioria dos casos (conjunto de dados e estratégia de partição), o FLEG Full foi realizado com 100 *chunks* no treinamento da GAN. Somente com o CIFAR-10, usando a partição Dir05, essa configuração usou 50 *chunks*. Curiosamente, esse também é o único experimento selecionado a usar *Fixed* no parâmetro para o número de dados sintéticos. Na Tabela 1, seus parâmetros estão em negrito.

- **FLEG Eco**: busca menores custos de computação e principalmente de comunicação. Em todos os casos, utiliza 1 *chunk* e 25 épocas no treinamento da GAN. Na Tabela 1, seus parâmetros estão em itálico.
- **FLEG Smart**: representa um *trade-off* entre FLEG Full e FLEG Eco. Aceita uma pequena perda na acurácia, por custos significativamente menores. Em todos os casos, utiliza 10 *chunks*, variando o número de épocas entre 25 e 35 para o treinamento da GAN. Na Tabela 1, seus parâmetros estão sublinhados.

Tabela 1. Resumo dos resultados de diferentes combinações de parâmetros em FLEG

Dataset	Partição	C	E	Modo S	Acurácia Máx.	Tempo (min)	Tráf. (GB)
CIFAR10	Class.	**100**	**30**	**Dyn.**	0.3095	45.3191	68.9167
CIFAR10	Class.	10	35	Dyn.	0.2774	42.1456	8.1349
CIFAR10	Class.	*1*	*25*	*Dyn.*	0.2614	33.2384	0.6952
CIFAR10	Dir01	**100**	**30**	**Dyn.**	0.3907	55.4752	68.9153
CIFAR10	Dir01	10	35	Dyn.	0.3471	39.9115	8.1381
CIFAR10	Dir01	*1*	*25*	*Dyn.*	0.3175	32.6008	0.6846
CIFAR10	Dir05	**50**	**25**	**Fix.**	0.4567	55.8751	29.0478
CIFAR10	Dir05	10	25	Dyn.	0.4553	41.8758	5.8725
CIFAR10	Dir05	*1*	*25*	*Dyn.*	0.4360	38.3409	0.7062
MNIST	Class.	**100**	**25**	**Dyn.**	0.9418	46.4474	19.6244
MNIST	Class.	10	25	Dyn.	0.9179	33.2042	2.0502
MNIST	Class.	*1*	*25*	*Dyn.*	0.8601	32.5163	0.3031
MNIST	Dir01	**100**	**25**	**Dyn.**	0.9790	41.3099	19.6142
MNIST	Dir01	10	25	Dyn.	0.9742	33.9948	2.0512
MNIST	Dir01	*1*	*25*	*Dyn.*	0.9710	34.6386	0.3043
MNIST	Dir05	**100**	**20**	**Dyn.**	0.9877	53.7496	15.7144
MNIST	Dir05	*1*	*25*	*Dyn.*	0.9863	37.2471	0.2999

Na Tabela 1, os experimentos estão ordenados de forma decrescente pela acurácia. Não por acaso, o tráfego de rede também segue essa ordenação, evidenciando o *trade-off* entre acurácia e custo de comunicação. O tempo total do experimento apresenta comportamento semelhante, com exceção do FLEG Smart e FLEG Eco para o MNIST sob Dir01. A escolha das combinações de parâmetros apresentadas nessa tabela foi feita a partir de uma tabela completa com todas as combinações experimentadas, priorizando a acurácia para o FLEG Full (parâmetros em negrito), os custos de comunicação e computação para o FLEG Eco (parâmetros em itálico) e um equilíbrio entre ambos para o FLEG Smart (parâmetros sublinhados). Nota-se que, para o *dataset* MNIST sob a partição Dir05, não há a presença de FLEG Smart, uma vez que o Eco já atinge níveis altos e próximos ao FLEG Full, dispensando a necessidade de um meio-termo.

De forma geral, observa-se que o uso da estratégia *Dynamic* para o número de dados sintéticos resulta em acurácias ligeiramente superiores. Isso pode ser explicado pela relação entre a qualidade da geração e a capacidade do classificador em distinguir as classes. Assim, ao empregar uma estratégia de geração progressiva, menos dados são gerados nos níveis iniciais, quando a rede ainda está menos desenvolvida. Além disso, valores mais altos de *chunks* tendem a aumentar a acurácia, sendo $C = 100$ um bom equilíbrio entre o treino da discriminadora e da geradora, como observado em FedGenIA [Guaragna et al. 2025]. O número de épocas da GAN também apresentou correlação positiva com a acurácia, mas negativa com o tempo e o tráfego, como esperado. Nota-se que o número de *chunks* é o fator que mais influencia o aumento no custo de comunicação, já que há transmissão a cada *chunk*. Ressalta-se que o tempo reportado reflete majoritariamente o custo computacional, pois não há simulação de comunicação. Assim, o tempo de treino não cresce proporcionalmente com C, pois *chunks* menores reduzem o custo de cada treinamento, compensando o maior número de execuções.

4.2. Acurácia

Nesta seção, avaliamos com maior granularidade a acurácia de validação do FLEG. Além da acurácia máxima, analisamos seu comportamento ao longo das épocas. As variações do FLEG (Full, Smart e Eco) são comparadas entre si e com as *baselines* FedAvg [McMahan et al. 2016] e FedProx [Li et al. 2020], esta última uma solução simples que melhora o desempenho do FedAvg em cenários com dados não-IID ao adicionar um termo proximal à função de perda. Como o FLEG atua apenas nos dados e não nos algoritmos de agregação, também avaliamos sua combinação com o FedProx, utilizando as mesmas configurações da Tabela 1. Para as *baselines*, adotamos $p = 100$, uma escolha conservadora que fornece ampla margem para a obtenção de melhores acurácias ao permitir longos treinamentos.

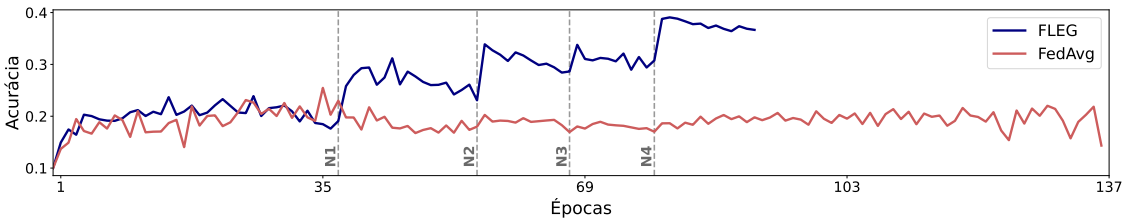

Figura 2. Acurácia ao longo das épocas de FLEG com marcas transitórias de níveis comparado com FedAvg usando CIFAR-10 e particionamento Dir01.

O impacto do FLEG sobre a acurácia da classificadora em cada nível é destacado na Figura 2. Nessa figura, observa-se que, no conjunto CIFAR-10 sob a partição Dir01, o modelo classificador apresenta ganhos significativos de acurácia a cada novo nível. Em contraste, o FedAvg mantém desempenho semelhante ao do nível inicial do FLEG, mesmo após rodadas adicionais de treinamento.

Na Figura 3, é apresentado o comportamento da acurácia no conjunto de validação ao longo das épocas de treinamento. O eixo horizontal foi limitado para facilitar a comparação direta com o FLEG, uma vez que as *baselines* são treinadas por mais épocas. No entanto, essas mantêm comportamento semelhante em rodadas subsequentes, e o treinamento completo pode ser observado na Figura 5, na qual a acurácia é analisada em

função do tempo total de treinamento. Observa-se que a principal distinção entre as variações do FLEG e suas respectivas *baselines* ocorre nas épocas mais avançadas do treinamento, indicando que níveis mais elevados do FLEG continuam impulsionando o aprendizado, enquanto o FedAvg tende a estagnar precocemente.

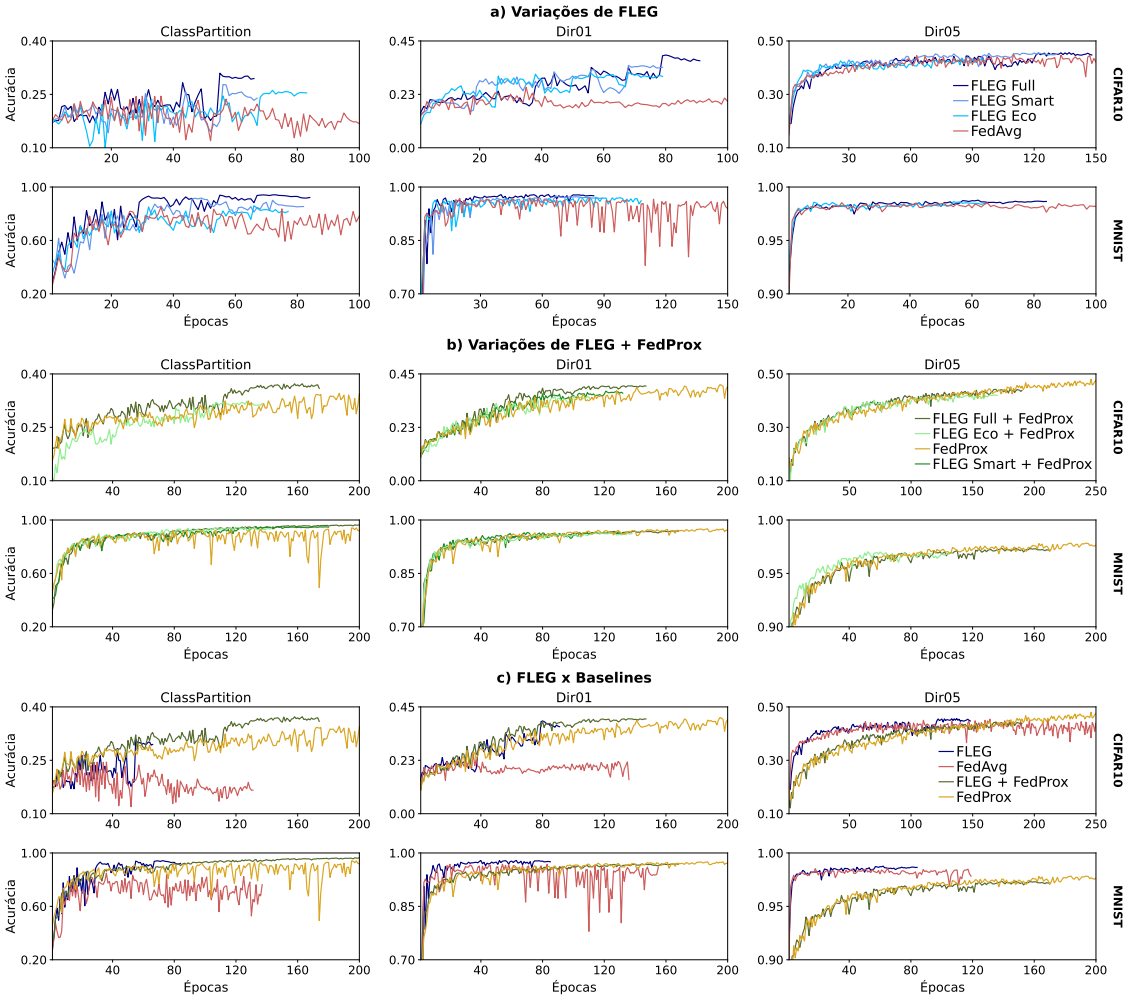

Figura 3. Acurácia ao longo das épocas para as três partições e para ambos conjuntos de dados avaliados. a) Variações de FLEG. b) Variações de FLEG + FedProx. c) FLEG x Baselines.

As maiores diferenças de desempenho ocorrem no CIFAR-10, nos cenários ClassPartition e Dir01, onde a heterogeneidade é mais elevada e o FedAvg tem maior dificuldade de convergência. O maior destaque é com o CIFAR-10 sob Dir01, com cerca de 14 pontos percentuais de diferença entre FLEG Full (0.39) e FedAvg (0.25). Em cenários mais simples, como MNIST sob Dir01 e Dir05, os ganhos são mais modestos, especialmente entre as variações do FLEG. Em três dos quatro experimentos com Dir05, não há espaço para a configuração Smart, indicando que estratégias de menor custo podem ser suficientes nesses casos. Nesse cenário mais suave, o FedProx supera FLEG + FedProx em épocas mais avançadas, enquanto em cenários mais complexos, como ClassPartition com CIFAR-10, FLEG + FedProx alcança maiores acurácias.

4.3. Custo de Comunicação

Apesar de atingir o objetivo de melhorar o desempenho do modelo classificador em cenários com alta heterogeneidade estatística, o FLEG implica um aumento significativo nos custos de comunicação. Na Figura 4, apresentamos o volume total de dados transmitidos pela rede ao longo do treinamento, medido em gigabytes, em comparação com a acurácia máxima atingida. Nessa análise, comparamos as três configurações do FLEG com a *baseline* FedAvg (FedProx não altera os custos de comunicação), limitada a 150 épocas de treinamento, evidenciando o aumento do custo de comunicação associado à solução proposta. Observa-se também uma diferença clara entre as variações do FLEG, indicando que é possível reduzir significativamente os custos de comunicação ao se permitir uma pequena degradação no desempenho do modelo. Em cenários menos desafiadores, como MNIST sob a partição Dir05, pode ser mais vantajoso empregar FedAvg ou FedProx. Por outro lado, em cenários mais complexos, como CIFAR-10 com partição Dir01 ou CIFAR, o uso do FLEG torna-se mais atrativo, demonstrando que a abordagem proposta é particularmente adequada para tarefas em níveis de maior heterogeneidade.

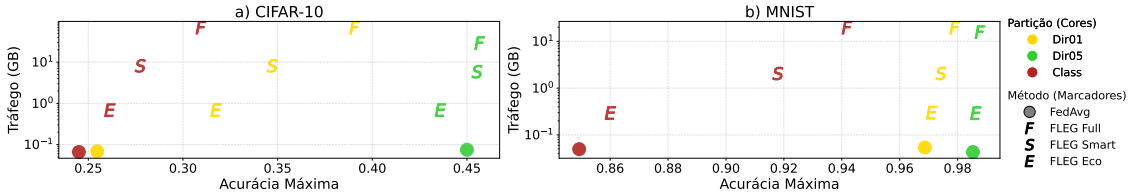

Figura 4. Custo de comunicação pela acurácia máxima. a) Experimentos realizados com o CIFAR-10. b) Experimentos realizados com o MNIST.

4.4. Custo de Computação

Também avaliamos o custo computacional dos métodos por meio do tempo total de execução. O *trade-off* entre tempo e acurácia é apresentado na Figura 5, em que o eixo horizontal indica o tempo (minutos) necessário para atingir cada valor de acurácia no eixo vertical. Observa-se a evolução gradual da acurácia ao longo do treinamento, evidenciando que, em pouco tempo, o FLEG já supera o FedAvg. No cenário com CIFAR-10 sob Dir01, por exemplo, em pouco mais de 10 minutos o FLEG Eco atinge acurácia de 0.3, enquanto o FedAvg alcança no máximo 0.25.

Nota-se que a diferença de usar FLEG em FedProx é menos acentuada, mostrando uma pequena superioridade na acurácia com mesmos tempos de treinamento, nos cenários com maior heterogeneidade estatística. Além disso, nota-se que o FedProx tem um tempo de treinamento maior do que FedAvg, mas isso se da por conta do número de épocas que cada solução treinou, como pode-se ver na Figura 3. FedProx acaba treinando por mais épocas, uma vez que o critério de parada utilizado foi a melhoria na acurácia e FedProx tem a capacidade de continuar melhorando por mais rodadas, enquanto FedAvg logo estagna.

Outro ponto interessante perceptível pela Figura 5 é o tempo de transição entre dois níveis de treinamento da classificadora. Essas transições são caracterizadas pelas longas linhas contínuas entre dois pontos, que corresponde ao tempo de treinamento da CGAN, enquanto as maiores variações em curto espaço de tempo evidenciam o treinamento da rede classificadora.

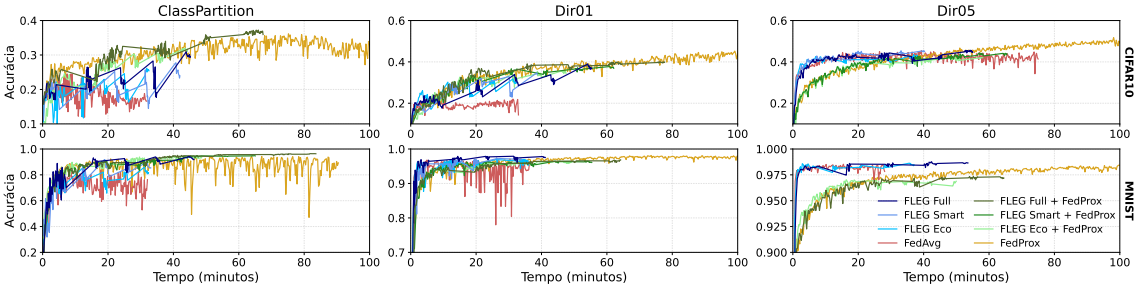

Figura 5. Acurácia por tempo.

5. Limitações e Trabalhos Futuros

O FLEG demonstrou capacidade relevante de melhorar o desempenho de modelos de classificação de imagens mesmo quando treinados com conjuntos combinados de *embeddings* reais e sintéticos, reduzindo riscos de vazamento de dados quando comparado ao uso de imagens sintéticas. Porém, a análise dos resultados também revelou limitações que apontam direções para trabalhos futuros. Em todos os experimentos, utilizamos um modelo classificador com cinco camadas, sendo a última responsável pela geração dos *logitos*, o que resultou em quatro níveis possíveis para a geração de *embeddings* sintéticos. A ausência de estagnação do desempenho no avanço dos últimos níveis, observada nas Figuras 2, 3 e 5, sugere que arquiteturas mais profundas, com maior número de níveis, podem proporcionar ganhos adicionais de desempenho.

Além disso, constatamos que o custo de comunicação associado ao treinamento da GAN no nível 0 é significativamente elevado. Uma possível alternativa para mitigar esse custo seria iniciar o treinamento da GAN com um modelo de menor dimensionalidade, utilizando como molde uma camada mais profunda do classificador. Em outras palavras, seria possível "pular" o nível 0 da GAN (e, consequentemente, o nível 1 da classificadora) e iniciar o processo diretamente a partir do nível 1 da GAN, após o treinamento inicial da classificadora. Outras possíveis formas de mitigar o custo de comunicação são explorar outros tipos de modelos, como autoencoders variacionais (VAEs), ou utilizar técnicas direcionadas a reduzir esse custo em aprendizado federado, como a fatoração de matrizes [Yu et al. 2026].

O FLEG foi avaliado em condições experimentais iniciais, de modo que seria interessante a realização de mais experimentos que incluam também a variação de outros hiperparâmetros como número de clientes e épocas locais. Contudo, para escalar o número de clientes, pode ser necessário utilizar técnicas de seleção que evitem um aumento significativo no custo de comunicação [Capanema et al. 2025, Maciel et al. 2024]. Ainda, seria relevante a implementação de outras soluções também baseadas em dados como as apresentadas na Seção 2 para a comparação com a estratégia proposta neste trabalho.

Por fim, a diferença de desempenho entre o FLEG e as *baselines* mostrou-se mais pronunciada no CIFAR-10 do que no MNIST, e em cenários mais heterogêneos como o ClassPartition e o Dir01, sugerindo que o FLEG tende a ser mais vantajoso em tarefas mais complexas. Assim, uma direção promissora para trabalhos futuros é a avaliação do método em cenários mais desafiadores e práticos, como em conjuntos de dados de imagens médicas para a classificação de enfermidades. Em FLEG, ambos os modelos classificador e generativo serem treinados do zero, porém para conjuntos de dados com

domínios mais específicos, pode-se utilizar modelos pré-treinados e realizar ajuste fino deles.

6. Conclusão

Neste artigo, apresentamos FLEG, uma nova solução para melhorar o desempenho de modelos de classificação de imagens em cenários de aprendizado federado com heterogeneidade estatística. Em FLEG, o treinamento do modelo classificador é organizado em níveis e alternado com o treinamento de uma Rede Adversária Generativa Condicional (CGAN), responsável pela geração de *embeddings* sintéticos que são compartilhados com os clientes, que os incorpora de forma a homogeneizar suas distribuições de classes. Os resultados demonstram que o FLEG é capaz de melhorar significativamente a acurácia de modelos classificadores ao empregar *embeddings* sintéticos para reduzir a heterogeneidade estatística da federação. Além disso, ao evitar o compartilhamento de imagens sintéticas, a abordagem contribui para a mitigação de potenciais riscos de vazamento de dados sensíveis.

7. Agradecimentos

Gostaríamos de agradecer ao Banco BTG Pactual pelo financiamento dessa pesquisa. Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTEX, coordenado pela Softex e publicado Arquitetura Cognitiva (Fase 3), DOU 01245.003479/2024-10. Este trabalho foi parcialmente patrocinado pelo projeto CNPq 407192/2025-5.

Referências

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. (2021). Federated learning based on dynamic regularization.

Ahmed, N., Wahed, M., and Thompson, N. C. (2023). The growing influence of industry in ai research. *Science*, 379(6635):884–886.

Alzubaidi, L., Zhang, J., Humaidi, A. J., and et al. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8:53.

Capanema, C. G. S., de Souza, A. M., da Costa, J. B. D., Silva, F. A., Villas, L. A., and Loureiro, A. A. F. (2025). A novel prediction technique for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 13(1):5–21.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Duan, M., Liu, D., Chen, X., Liu, R., Tan, Y., and Liang, L. (2020). Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):59–71.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Guaragna, G. S., Da Costa, J. B. D., and De Souza, A. M. (2025). Federated learning with iterative synthetic data augmentation.

Huangsuwan, K., Liu, T., See, S., Beng Ng, A., and Vateekul, P. (2025). Feddrip: Federated learning with diffusion-generated synthetic image. *IEEE Access*, 13:10111–10125.

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., and Kim, S. (2018). Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2021). Scaffold: Stochastic controlled averaging for federated learning.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, P., Zhang, H., Wu, Y., Qian, L., Yu, R., Niyato, D., and Shen, X. (2024a). Filling the missing: Exploring generative ai for enhanced federated learning over heterogeneous mobile edge devices. *IEEE Transactions on Mobile Computing*, 23(10):10001–10015.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks.

Li, Z., Shao, J., Mao, Y., Wang, J. H., and Zhang, J. (2022). Federated learning with gan-based data synthesis for non-iid clients.

Li, Z., Sun, Y., Shao, J., Mao, Y., Wang, J. H., and Zhang, J. (2024b). Feature matching data synthesis for non-iid federated learning. *IEEE Transactions on Mobile Computing*, 23(10):9352–9367.

Maciel, F., da Costa, J. B. D., Gonzalez, L. F. G., de Souza, A. M., Villas, L. A., and Bittencourt, L. F. (2024). Adaptive fit fraction based on model performance evolution in federated learning. In *2024 11th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 77–84.

Maliakel, P. J., Ilager, S., and Brandic, I. (2024). Fligan: Enhancing federated learning with incomplete data using gan.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2016). Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Pan, H., Hong, Z., Durak, G., Xu, Z., and Bagci, U. (2025). Federated breast cancer detection enhanced by synthetic ultrasound image augmentation.

Pennisi, M., Salanitri, F. P., Bellitto, G., Casella, B., Aldinucci, M., Palazzo, S., and Spampinato, C. (2023). Feder: Federated learning through experience replay and privacy-preserving data synthesis.

Salvo, F. D., Nguyen, H. H. M., and Ledig, C. (2025). Embedding-based federated data sharing via differentially private conditional vaes.

Salvo, F. D., Tafler, D., Doerrich, S., and Ledig, C. (2024). Privacy-preserving datasets by capturing feature distributions with conditional vaes. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA.

Wu, Q., Chen, X., Zhou, Z., and Zhang, J. (2020). Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Transactions on Mobile Computing*, 21(8):2818–2832.

Yonetani, R., Takahashi, T., Hashimoto, A., and Ushiku, Y. (2019). Decentralized learning of generative adversarial networks from non-iid data.

Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., and Yonetani, R. (2020). Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data.

Yu, S., Zhu, K., Liang, F., Wang, J., Kant, K., and Yin, L. (2026). Robust multimodal federated learning for non-iid multimodal data with incompleteness. *Future Generation Computer Systems*, 174:107948.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhao, Z., Yang, F., and Liang, G. (2023). Federated learning based on diffusion model to cope with non-iid data. In *Pattern Recognition and Computer Vision: 6th Chinese Conference, PRCV 2023, Xiamen, China, October 13–15, 2023, Proceedings, Part IX*, page 220–231, Berlin, Heidelberg. Springer-Verlag.

Zhu, H., Xu, J., Liu, S., and Jin, Y. (2021). Federated learning on non-iid data: A survey.