

# Clusterização por Similaridade de Pesos como Estratégia para Mitigar o *Client Drift* no Aprendizado Federado

Enzo Rodrigues Novais Dias<sup>1</sup>, Renan R. de Oliveira<sup>1,2</sup>  
e Antonio Oliveira-Jr<sup>1,3</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal de Goiás (UFG), GO, Brasil

<sup>2</sup>Instituto Federal de Goiás (IFG), GO, Brasil

<sup>3</sup>Fraunhofer Portugal AICOS, Porto, Portugal

enzodias@discente.ufg.br, renan.rodrigues@ifg.edu.br,  
antoniojr@ufg.br

**Abstract.** *This work addresses the challenge of client drift in Federated Learning (FL) caused by Non-IID data. We propose a two-phase algorithm that clusters clients based on the similarity of their local model weights, followed by independent federated training within each cluster. The methodology was validated on the MNIST dataset under two heterogeneous scenarios. The first represents an extreme case, where each client holds data restricted to a single class. The second is based on a Dirichlet distribution ( $\alpha = 0.1$ ), producing unbalanced combinations with a predominance of few classes per client. The proposed algorithm improved the global baseline accuracy from 26.32% to 100% in the first case, and outperformed the baseline of 86.67% in the second, achieving up to 92.09%. The results show that clustering is an effective strategy to mitigate heterogeneity and enhance the stability of federated training.*

**Resumo.** *Este trabalho aborda o desafio do client drift no Aprendizado Federado (Federated Learning – FL) causado por dados Non-IID. Propomos um algoritmo em duas fases, com clusterização dos dispositivos por similaridade dos pesos de modelos locais e treinamento federado independente por cluster. A metodologia foi validada no dataset MNIST em dois cenários de heterogeneidade. O primeiro corresponde ao caso extremo, onde cada dispositivo possui dados restritos a uma única classe. O segundo é baseado na distribuição Dirichlet ( $\alpha = 0.1$ ), gerando dados desbalanceadas e com predominância de poucas classes por dispositivo. O algoritmo proposto elevou a acurácia do modelo global de 26.32% para 100% no primeiro caso e superou a baseline de 86.67% no segundo caso, atingindo até 92.09%. Os resultados mostram que a clusterização é uma estratégia eficaz para mitigar a heterogeneidade e melhorar a estabilidade do treinamento federado.*

## 1. Introdução

Com os avanços em algoritmos, poder computacional e disponibilidade de conjuntos de dados, as técnicas de Aprendizado de Máquina (*Machine Learning* – ML) têm sido exploradas em diversas aplicações, como condução autônoma, indústria inteligente, realidade aumentada/virtual e da Internet das Coisas (*Internet of Things* – IoT)

[Hellström et al. 2022]. No entanto, os algoritmos de ML tradicionais exigem grandes volumes de dados para o treinamento dos modelos, o que requer a coleta e o envio desses dados para um servidor central. Além disso, nos últimos anos, políticas e leis de privacidade de dados passaram a exigir a proteção dos dados pessoais e a regulamentar o compartilhamento de informações entre empresas, a fim de evitar abusos no uso dos dados. Dessa forma, a transferência direta dos dados brutos para um servidor centralizado pode representar um risco significativo à privacidade, uma vez que viola princípios fundamentais dessas legislações.

Neste contexto, o FL [McMahan et al. 2016] surge como uma abordagem descentralizada de treinamento de modelos de ML, na qual os dados permanecem nos dispositivos locais e apenas as atualizações dos modelos são compartilhadas com um servidor central [Li et al. 2021]. Em vez de enviar os dados brutos para um servidor centralizado, cada dispositivo participante do FL treina um modelo localmente utilizando seus próprios dados e, posteriormente, envia os parâmetros do modelo atualizado para um servidor agregador. O servidor, então, combina as atualizações recebidas de múltiplos dispositivos para gerar um modelo global atualizado [Wahab et al. 2021].

Essa abordagem reduz os riscos de vazamento de privacidade, ao mesmo tempo em que minimiza o consumo de largura de banda para transmissão dos modelos locais, pois apenas os parâmetros dos modelos são transmitidos para o servidor agregador [Yang et al. 2022]. Além disso, a colaboração entre os dispositivos de borda permite obter um modelo global mais robusto e preciso do que aquele treinado em um único dispositivo, tornando o FL uma solução promissora para aplicações que exigem segurança, eficiência e escalabilidade. Além do mais, o paralelismo no FL pode acelerar o processo de treinamento [Brecko et al. 2022].

Além do mais, as abordagens de FL devem considerar a heterogeneidade estatística dos dados que compromete a generalização do modelo global. Por exemplo, dispositivos móveis e IoT geram e coletam dados que, de forma intrínseca, apresentam distribuições de forma Non-IID (*Non-Independent and Identically Distributed*) [Brecko et al. 2022], ou seja, os dados apresentam variações significativas entre os dispositivos, tanto na distribuição desigual das classes quanto na variação da quantidade de amostras. Isto reflete o uso particular e o contexto de cada dispositivo, como padrões de tráfego, localização geográfica, preferências do usuário, bem como, o tipo de sensor IoT empregado. Essa heterogeneidade gera o fenômeno conhecido como *client drift* [Ma et al. 2022], onde cada modelo local tende a se especializar em seus próprios dados enviesados, desviando-se da direção ótima global. Na etapa de agregação, esses vieses podem colidir, desviando o modelo global em direções conflitantes [Zhao et al. 2022].

Diante desse desafio, este artigo propõe uma estratégia de “dividir para conquistar” como forma de gerenciar a heterogeneidade dos dados e mitigar seus efeitos adversos na agregação dos modelos locais enviados por dispositivos participante de tarefas de FL. Neste caso, em vez de treinar um único modelo global para todos os dispositivos, propõe-se agrupar os dispositivos em *clusters* com maior similaridade de dados e, em seguida, treinar um modelo de FL especializado para cada grupo. A principal contribuição metodológica deste estudo consiste em na capacidade do servidor agregador em utilizar os pesos iniciais dos modelos locais (enviados na etapa de inicialização do algoritmo) como uma “impressão digital”, o que possibilita mensurar a similaridade entre os dispositivos e

realizar a clusterização de maneira eficiente, sem a necessidade de acessar ou inspecionar diretamente os dados privados.

Os resultados obtidos mostram que o baseline global, correspondente ao treinamento federado via FedAvg (*Federated Averaging*) [McMahan et al. 2016] sem clusterização, apresentou forte limitação frente à heterogeneidade dos dados. No cenário em que cada dispositivo possui apenas um único rótulo, o modelo global alcançou apenas 26.32% de acurácia, resultado de gradientes altamente conflitantes. Nesse mesmo cenário, a clusterização utilizando o algoritmo K-Means foi bem-sucedida, formando grupos homogêneos de dispositivos e permitindo que cada modelo por *cluster* atingisse 100% de acurácia, superando o problema de *client drift*. Já no cenário mais realista, baseado na distribuição Dirichlet com  $\alpha = 0.1$ , o baseline global obteve acurácia de 86.67%, enquanto os modelos treinados por cluster superaram esse valor em todos os casos, alcançando valores de acurácia entre 89.14% e 92.09%.

As contribuições centrais deste artigo são elencadas a seguir: (i) propõe-se um algoritmo de duas fases para o FL, baseada na clusterização de dispositivos por similaridade dos pesos iniciais dos modelos locais e posterior treinamento federado por *cluster*; (ii) são apresentadas validações empíricas da proposta em cenários com dados de dispositivos heterogêneos, demonstrando a mitigação do *client drift* e ganhos de desempenho em relação ao baseline global; (iii) o estudo evidencia que a clusterização é uma estratégia prática e eficaz para melhorar a estabilidade da agregação de modelos por *cluster*; (iv) disponibiliza-se o código-fonte<sup>1</sup> da simulação para permitir a reprodução e validação dos resultados.

Para além desta seção introdutória, o restante deste artigo está organizado em seções, conforme descrito a seguir. A Seção 2 apresenta a fundamentação teórica e discute os trabalhos relacionados. A Seção 3 apresenta a proposta do algoritmo de FL baseado em clusterização. A Seção 4 apresenta o cenário de simulação. A Seção 5 discute os resultados e a análise da simulação. Por fim, a Seção 6 apresenta as considerações finais.

## 2. Fundamentação Teórica

### 2.1. Aprendizado Federado

O primeiro algoritmo de FL, denominado FedAvg [McMahan et al. 2016], foi introduzido em 2016 como uma solução eficiente para treinamento colaborativo de modelos sem a necessidade de compartilhamento dos dados dos dispositivos. O processo treinamento do FedAvg envolve a troca periódica de parâmetros entre os dispositivos e o servidor agregador, da seguinte forma:

1. **Seleção de dispositivos e distribuição do modelo global:** O servidor identifica e seleciona dispositivos que atendem aos critérios necessários para participar da rodada de treinamento. Após a seleção, o modelo global mais recente é transmitido aos dispositivos escolhidos, garantindo que cada um receba a versão atualizada como base para o treinamento local.
2. **Treinamento local:** Após receber o modelo global, cada dispositivo atualiza o modelo local com base no seu conjunto de dados.

---

<sup>1</sup>Disponível em: <https://github.com/enzodias1/ic-federated-learning>

3. **Upload da atualização do modelo local:** Cada dispositivo envia sua atualização para o servidor por meio de um enlace de comunicação.
4. **Agregação do modelo global:** O servidor atualiza o modelo global com base na agregação dos modelos locais recebidos, calculando a média ponderada com base na quantidade de dados dos dispositivos.

De acordo com [Li et al. 2021], o processo de treinamento distribuído pode ser repetido por várias rodadas de comunicação até que a convergência seja alcançada ou interrompido conforme a definição de um critério de parada.

## 2.2. Desafios de Dados Non-IID

O problema da distribuição de dados Non-IID está relacionado ao problema de heterogeneidade de dados. De acordo com [Zhao et al. 2022], os algoritmos de ML tradicionais são geralmente treinados com base em dados balanceados, na suposição de que os dados são IID (*Independent and Identically Distributed*). No entanto, os conjuntos de dados gerados por dispositivos móveis e IoT baseiam-se no padrão de uso um determinado usuário e apresentam uma distribuição bastante diferente entre si [Brecko et al. 2022]. Portanto, o conjunto de dados de cada usuário não será uma representação da distribuição da população [McMahan et al. 2016].

A heterogeneidade da distribuição de dados no FL pode ser expressa pela caracterização da assimetria da distribuição dos diferentes conjuntos. De acordo com [Ma et al. 2022], essa assimetria pode ser ocasionada pela distribuição enviesada dos dados locais, que varia entre os diferentes clientes. Além do mais, a distribuição dos rótulos dos dados locais dos dispositivos podem ser distintos, ainda que a distribuição da probabilidade do rótulo considerando os demais dispositivos seja a mesma.

A assimetria também pode ser causada pelas diferentes características dos dados locais de cada dispositivo que podem indicar o mesmo rótulo, bem como, diferentes rótulos podem indicar as mesmas características dos dados locais de cada dispositivo. Por fim, a quantidade de dados locais de cada dispositivo pode ser significativamente diferente, provocando a heterogeneidade da distribuição de dados [Ma et al. 2022].

## 2.3. Trabalhos Relacionados

Na formulação inicial do FL proposta por [McMahan et al. 2016], os autores introduziram conceitos fundamentais e apresentaram o algoritmo FedAvg. Em seguida, foram desenvolvidas variantes deste algoritmo, visando mitigar os efeitos da heterogeneidade estatística dos dados dos dispositivos. As abordagens como FedProx [Li et al. 2020] e Scaffold [Karimireddy et al. 2020] incorporam técnicas que ajustam a agregação e a atualização do modelo local, restringindo desvios excessivos, promovendo o alinhamento das representações aprendidas pelos modelos locais com o modelo global e estabilizando a convergência.

Em [Zhao et al. 2022], os autores demonstram que, em cenários Non-IID altamente heterogêneos, a acurácia do FedAvg pode sofrer reduções de até 55%, associadas à divergência dos pesos dos modelos treinados localmente pelos dispositivos. De forma complementar, o trabalho de [Ma et al. 2022] apresenta um levantamento abrangente sobre o problema dos dados Non-IID em FL, destacando que a heterogeneidade estatística é um dos maiores desafios para a estabilidade e a acurácia do modelo global.

Em [Zhao et al. 2024], os autores questionam a viabilidade do treinamento de um único modelo global. Dessa forma, os autores apresentam um algoritmo de FL que utiliza a similaridade de gradientes para formar *clusters* dinâmicos e combina os modelos resultantes em um *ensemble*, alcançando maior estabilidade e melhor desempenho em cenários de dados Non-IID.

Inspirado pela estratégia de “dividir para conquistar”, este trabalho se diferencia na metodologia de clusterização ao adotar uma abordagem distinta. Em vez de utilizar gradientes, que se mostraram numericamente instáveis em nossos experimentos preliminares, propomos e validamos o uso dos pesos completos dos modelos como uma “impressão digital” capaz de medir a similaridade entre dispositivos e formar *clusters*, sem comprometer a privacidade dos dados brutos dos dispositivos.

### 3. Proposta de Algoritmo de FL baseado em Clusterização

A heterogeneidade estatística dos dados no FL gera o *client drift*, dificultando a convergência do modelo global. Para mitigar esse efeito, propõe-se o Algoritmo 1 em duas fases: inicialmente, os dispositivos são agrupados pela similaridade de seus modelos locais e, em seguida, os modelos de FL são treinados e avaliados por *cluster*.

#### 3.1. Fase 1: Clusterização via Similaridade de Modelos

Na fase 1, o servidor inicia o processo enviando  $w_0$  para todos os  $\mathcal{K}$  dispositivos. Cada dispositivo treina esse modelo por  $E_{\text{init}}$  épocas usando os seus dados locais. Em seguida, o servidor recebe  $w_{\text{init}}^k$  de cada dispositivo, como sendo uma versão levemente ajustada do modelo inicial, e trata os pesos ajustados como uma “impressão digital” dos dados de cada dispositivo.

A premissa é que dispositivo que possuam dados semelhantes acabarão produzindo modelos também semelhantes. Por fim, o servidor aplica um algoritmo de agrupamento (neste caso, o K-Means) para separar os dispositivos em  $N_c$  grupos com maior similaridade. Assim, espera-se que dispositivos que têm distribuições de dados parecidas são colocados no mesmo grupo.

#### 3.2. Fase 2: Treinamento e Avaliação Distribuída por *Cluster*

Na Fase 2, o treinamento ocorre separadamente dentro de cada *cluster* de dispositivos previamente definido. Para cada *cluster*, o servidor inicializa um modelo próprio, e em seguida, inicia o ciclo padrão do FL, conforme apresentado na Seção 2.1. Em cada rodada, alguns dispositivos do *cluster* são selecionados, recebem a versão atual  $w_t^c$  do modelo do grupo, realizam o treinamento local com seus dados e depois retornam  $w_{t+1}^{c,k}$  ao servidor. O servidor agrupa essas atualizações e gera a nova versão  $w_{t+1}^c$  do modelo global do *cluster*.

Após a atualização do modelo  $w_{t+1}^c$ , é realizada a avaliação distribuída por *cluster*. Nessa etapa, o modelo resultante é enviado aos dispositivos para ser testado nos dados locais dos dispositivos pertencentes ao próprio grupo. Essa avaliação mostra o quanto o modelo especialista consegue capturar e atender às características específicas daquele *cluster*, sem considerar dados externos ou de outros grupos.

---

**Algoritmo 1:** FL baseado em clusterização de modelos.

---

```
1  $\triangleright$  Fase 1: Clusterização de Dispositivos
2   Servidor:
3     inicialização de  $w_0$ 
4     para cada dispositivo  $k$  em paralelo faça  $\triangleright \forall k \in \mathcal{K}$ 
5        $[w_{\text{init}}^k] \leftarrow \text{InicializaçãoNoDispositivo}(k, w_0)$ 
6       Clusters  $\leftarrow \text{KMeans}([w_{\text{init}}^k], N_c)$ 
7     InicializaçãoNoDispositivo( $k, w_0$ ):
8        $w_{\text{init}}^k \leftarrow \text{TreinamentoLocal}(k, w_0, E_{\text{init}})$ 
9       EnviaModeloAoServidor( $w_{\text{init}}^k$ )
10   $\triangleright$  Fase 2: Treinamento de Modelos por Cluster
11  para cada  $c \in \text{Clusters}$  em paralelo faça
12    inicialização de  $w_1^c$ 
13     $\triangleright$  Execução do Loop de FL por cluster
14    para cada rodada  $t = 1, 2, \dots$  em paralelo faça
15       $\triangleright$  Seleção de dispositivos e distribuição de  $w_t^c$ 
16       $\triangleright$  Treinamento local de  $w_t^{c,k}$  no dispositivo
17       $\triangleright$  Upload de  $w_{t+1}^{c,k}$  com a atualização do modelo local
18       $\triangleright$  Agregação do modelo  $w_{t+1}^c \leftarrow \text{Agregação}(\forall w_{t+1}^{c,k} \in c)$ 
19       $\triangleright$  Avaliação distribuída de  $w_{t+1}^c$  por cluster
20  Saída: Modelos  $w_{\text{global}}^c$  por cluster
```

---

## 4. Configuração dos Experimentos

### 4.1. Cenário da Simulação

Considere um ambiente com dados heterogêneos e um conjunto de 100 dispositivos participantes de uma tarefa de FL. As tarefas envolvem problemas de classificação de imagens utilizando o conjunto de dados MNIST, amplamente adotado como *benchmark* em pesquisas de FL, conforme os trabalhos de [McMahan et al. 2016], [Zhu et al. 2020], [Chen et al. 2021], [Zhao et al. 2022] e [Chen et al. 2022].

### 4.2. Cenários de Heterogeneidade dos Dados

As simulações consideram dois cenários de heterogeneidade dos dados. No primeiro, é definido um caso extremo de heterogeneidade estatística, onde cada dispositivo possui apenas uma única classe do MNIST. No segundo, adota-se uma distribuição Dirichlet com parâmetro  $\alpha = 0.1$ , assegurando uma partição realista, com combinações desbalanceadas e com a predominância de poucas classes para cada dispositivo.

### 4.3. Arquitetura de ML

Para a tarefa de classificação do MNIST, considera-se uma arquitetura de rede neural do tipo *Multi-Layer Perceptron* (MLP) com 101.770 parâmetros treináveis, composta por uma camada de entrada, seguida de uma única camada oculta com 128 neurônios, uma função de ativação ReLU e uma camada de saída *softmax*. Além do mais, é utilizado o

ADAM (*Adaptive Moment Estimation*) como método de otimização do modelo, *Sparse Categorical Crossentropy* como função de perda, com um *batch size* de tamanho 128.

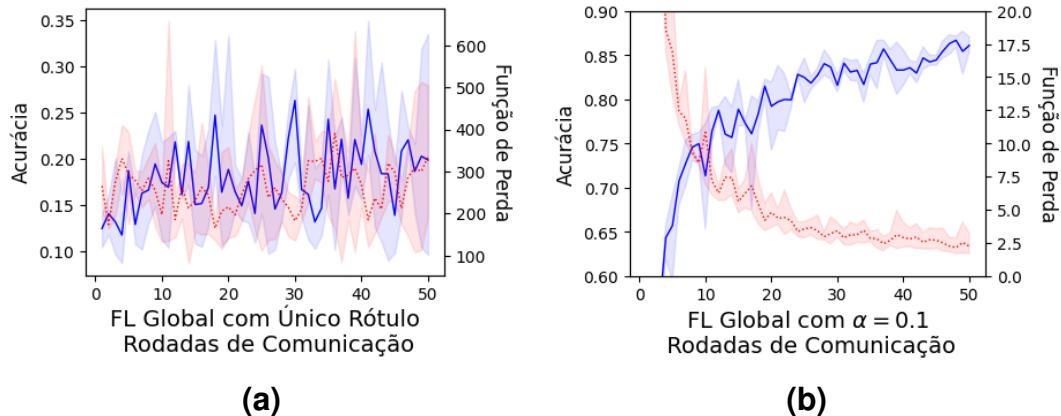
## 5. Análise dos Resultados

As simulações de FL foram realizadas por 50 rodadas de comunicação, onde os resultados representam a média obtida em três execuções. Nos experimentos com clusterização, foram selecionados  $\sim \frac{1}{3}$  dos membros do *cluster* (máx. 10) como os dispositivos participantes em cada rodada de comunicação. Na fase de inicialização, cada modelo local foi treinado por 10 épocas locais antes de ser enviado ao servidor, para posteriormente, ser utilizado no processo de clusterização. Além do mais, o algoritmo K-Means foi configurado para formar 10 grupos, correspondentes aos subconjuntos de dispositivos utilizados no treinamento federado por *cluster*.

### 5.1. Baseline de Referência

Para efeito de comparação, foi definida uma baseline FL Global, onde os 100 dispositivos foram considerados em conjunto, sem aplicação da clusterização. Neste caso, em cada rodada de comunicação, 10 dispositivos foram selecionados aleatoriamente para receber o modelo global, realizar treinamento local em seus dados por 1 época local, e posteriormente, enviar as atualizações ao servidor. A agregação foi definida com base no FedAvg, resultando nos valores da acurácia do  $w_{global}$  e da função de perda  $f(w_{global})$  que serviram como referência para avaliar os ganhos obtidos com a clusterização.

A Figura 1a apresenta a evolução do treinamento federado da baseline de referência no cenário de dados de único rótulo. Nesse caso, cada dispositivo dispõe exclusivamente de exemplos de uma única classe, o que levou a baseline a alcançar um valor médio de apenas 26.32% de acurácia. Esse resultado evidencia a acentuada degradação do modelo global causada pelo *client drift*, pois os dispositivos treinam em conjuntos de dados altamente enviesados e as atualizações enviadas ao servidor tornam-se conflitantes, comprometendo a convergência para um modelo capaz de representar todas as classes de forma equilibrada.



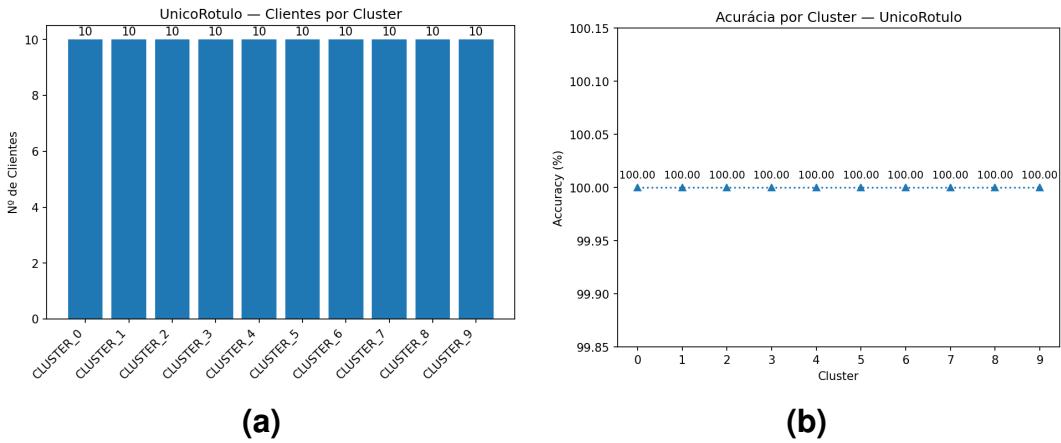
**Figura 1. Baselines de referência para o FL Global.**

Por outro lado, a Figura 1b apresenta a evolução do treinamento federado da baseline de referência no cenário de dados com uma distribuição baseada em Dirichlet com

$\alpha = 0.1$ . Neste caso, os dados representam um ambiente de heterogeneidade realista, assegurando uma partição não uniforme dos dados, com combinações desbalanceadas e predominância de poucas classes em cada dispositivo. Nesse cenário, o modelo global alcançou uma acurácia média de 86,67%, demonstrando que, embora a heterogeneidade imponha desafios, ainda é possível atingir boa convergência.

## 5.2. Clusterização com Cenário de Único Rótulo

Conforme a Figura 2a, o cenário de clusterização com único rótulo foi capaz de recuperar perfeitamente a estrutura subjacente dos dados, uma vez que o algoritmo K-Means ( $k = 10$ ) identificou corretamente os 10 grupos homogêneos, cada um composto por 10 dispositivos. Neste contexto, a Figura 2b mostra que o treinamento federado conduzido por *cluster* atingiu 100% de acurácia distribuída em todos os grupos, confirmando que os modelos especialistas convergiram sem dificuldades.



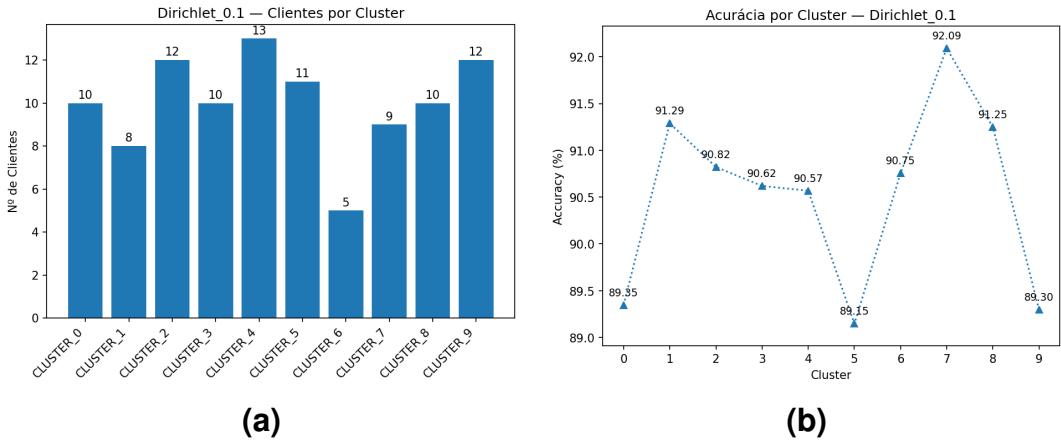
**Figura 2. Resultados do cenário de Único Rótulo.**

Considerando que o modelo global da baseline global de referência no cenário de único rótulo convergiu para apenas 26.32% de acurácia, a abordagem de clusterização resultou em um ganho de +73.68% para todos os grupos. A separação em grupos homogêneos elimina conflitos entre atualizações locais, garantindo a convergência. No cenário de único rótulo, a clusterização torna-se trivial, já que cada dispositivo possui apenas uma classe, e o treinamento federado por *cluster* também se simplifica, pois todos os dispositivos colaboram sobre dados idênticos.

## 5.3. Clusterização com Cenário baseado em Dirichlet com $\alpha = 0.1$

Conforme a Figura 3a, o cenário de clusterização baseado em Dirichlet com  $\alpha = 0.1$  apresentou variação no número de dispositivos por *cluster*, refletindo o desbalanceamento introduzido pela distribuição. Essa irregularidade na composição dos grupos impacta diretamente a qualidade dos modelos treinados, mas não de forma linear em relação ao número de dispositivos. A Figura 3b mostra que *clusters* maiores não apresentaram necessariamente maior acurácia, evidenciando que a homogeneidade interna dos dados exerce papel mais relevante do que a quantidade de dispositivos.

A Tabela 1 apresenta o ganho por *clusters*, onde os grupos superaram a baseline global de referência com valor médio 86,67% de acurácia, resultando em ganhos entre



**Figura 3. Resultados do cenário baseado em Dirichlet com  $\alpha = 0.1$ .**

$+2.47\%$  e  $+5.42\%$ . Embora os incrementos sejam menos expressivos que no cenário de único rótulo, eles evidenciam que a clusterização reduz o *client drift* mesmo em contextos realistas, contribuindo para maior estabilidade e desempenho do treinamento federado.

| Dados          | Cluster | Acurácia (%) | $\Delta\text{ACC}$ |
|----------------|---------|--------------|--------------------|
| $\alpha = 0.1$ | 0       | 89.34        | +2.67              |
|                | 1       | 91.28        | +4.61              |
|                | 2       | 90.82        | +4.15              |
|                | 3       | 90.61        | +3.94              |
|                | 4       | 90.56        | +3.89              |
|                | 5       | 89.14        | +2.47              |
|                | 6       | 90.75        | +4.08              |
|                | 7       | 92.09        | +5.42              |
|                | 8       | 91.24        | +4.57              |
|                | 9       | 89.29        | +2.62              |

**Tabela 1. Acurácia distribuída por cluster no cenário Dirichlet com  $\alpha = 0.1$ , onde  $\Delta\text{ACC}$  é a diferença com relação a baseline de referência.**

## 6. Considerações Finais

Este trabalho investigou a heterogeneidade de dados no FL e validou uma metodologia na clusterização de dispositivos por “impressão digital” de pesos e o treinamento federado por *cluster*. Os experimentos com o MNIST mostraram que a abordagem supera o FL global em cenários realistas (baseado em Dirichlet com  $\alpha = 0.1$ ) e atinge 100% de acurácia em cenários extremos (único rótulo). Assim, confirmou-se que a clusterização mitiga o *client drift* e melhora o desempenho do modelo global. Como trabalhos futuros, pretende-se automatizar a escolha da quantidade de grupos, o uso de informações e métodos alternativos para computar a similaridade dos modelos e a avaliação em *datasets* mais complexos. Além disso, pretende-se avaliar o algoritmo considerando métricas de custo de comunicação e de consumo energético, tanto durante o treinamento quanto na transmissão dos modelos.

## Referências

- Brecko, A., Kajati, E., Koziorek, J., and Zolotova, I. (2022). Federated Learning for Edge Computing: A Survey. *Applied Sciences*, 12(18).
- Chen, H., Huang, S., Zhang, D., Xiao, M., Skoglund, M., and Poor, H. V. (2022). Federated Learning Over Wireless IoT Networks With Optimized Communication and Resources. *IEEE Internet of Things Journal*, 9(17):16592–16605.
- Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., and Cui, S. (2021). A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. *IEEE Transactions on Wireless Communications*, 20(1):269–283.
- Hellström, H., da Silva Jr, J. M. B., Amiri, M. M., Chen, M., Fodor, V., Poor, H. V., and Fischione, C. (2022). Wireless for Machine Learning: A survey. *Foundations and Trends in Signal Processing*, 15(4):290–399.
- Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2020). SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning. In *International Conference on Machine Learning (ICML)*.
- Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.
- Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated Optimization in Heterogeneous Networks. In *Conference on Machine Learning and Systems (MLSys)*.
- Ma, X., Zhu, J., Lin, Z., Chen, S., and Qin, Y. (2022). A state-of-the-art Survey on Solving non-IID Data in Federated Learning. *Future Generation Computer Systems*, 135:244–258.
- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv*.
- Wahab, O. A., Mourad, A., Otrok, H., and Taleb, T. (2021). Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future Directions in Communication and Networking Systems. *IEEE Communications Surveys & Tutorials*, 23(2):1342–1397.
- Yang, Z., Chen, M., Wong, K.-K., Poor, H. V., and Cui, S. (2022). Federated Learning for 6G: Applications, Challenges, and Opportunities. *Engineering*, 8:33–41.
- Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2022). Federated Learning with Non-IID Data. *arXiv*.
- Zhao, Z., Wang, J., Hong, W., Quek, T. Q. S., Ding, Z., and Peng, M. (2024). Ensemble Federated Learning With Non-IID Data in Wireless Networks. *IEEE Transactions on Wireless Communications*, 23(4):3557–3571.
- Zhu, G., Wang, Y., and Huang, K. (2020). Broadband Analog Aggregation for Low-Latency Federated Edge Learning. *IEEE Transactions on Wireless Communications*, 19(1):491–506.