3FL: Seleção de Clientes Mais Rápidos para Aumento de Desempenho do Aprendizado Federado *Cross-Device*

Kaylani Bochie[1], **Matteo Sammarco**[2] **e Miguel Elias M. Campista**[1] *

[1] Universidade Federal do Rio de Janeiro (UFRJ) – PEE-COPPE/DEL-Poli/GTA
[2]Pesquisador independente

{kaylani,miguel}@gta.ufrj.br, matteosam@hotmail.it

Resumo. *Este trabalho propõe uma nova técnica para melhorar o desempenho do aprendizado federado e reduzir a latência total de treinamento. A proposta, chamada Fastest-First Federated Learning (3FL), é baseada na seleção de participantes mais rápidos durante o início do treinamento para reduzir a latência de treinamento e mitigar o efeito de dispositivos retardatários. A proposta é avaliada por meio de simulações utilizando distribuições de dados e configurações de clientes realistas para o cenário de aprendizado federado cross-device horizontal. Os resultados obtidos demonstram que é possível obter reduções na latência de treinamento de até 35% em comparação ao aprendizado federado tradicional. Além disso, os experimentos confirmam que a acurácia do modelo atinge resultados similares ou mesmo superiores, chegando a valores de até 97% em problemas de classificação de imagens.*

Abstract. *This paper proposes a new technique to increase federated learning performance and decrease its overall training latency. The proposal, called Fastest-First Federated Learning (3FL), is centered on selecting faster participants at the outset of training to minimize training latency and mitigate the impact of straggler devices. The proposal's assessment involves simulations using realistic data distributions and client configurations for horizontal cross-device federated learning scenarios. The obtained results demonstrate the potential to achieve reductions in training latency of up to 35% compared with traditional federated learning. Moreover, the experiments confirm that the model accuracy reaches similar or even superior results, achieving values of up to 97% for image classification problems.*

1. Introdução

O volume de dados gerados por dispositivos inteligentes tem crescido vertiginosamente, motivando o desenvolvimento de soluções que se beneficiem de grandes massas de dados [Naeem et al., 2022b], como as aplicações baseadas em aprendizado profundo [Bochie et al., 2021a]. Paralelamente, é possível observar preocupações cada vez maiores acerca da segurança da informação e, especificamente, da privacidade dos dados [Rimol, 2022]. Essas tendências impulsionam o desenvolvimento de novas técnicas,

*O presente trabalho foi realizado com apoio do CNPq; da FAPERJ, processos E-26/211.144/2019 e E-26/202.689/2018; da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES), Código de Financiamento 001; da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo n^o 15/24494-8; e da Fundação de Desenvolvimento da Pesquisa - Fundep - Rota 2030.

nas quais os dados dos usuários podem ser usados para a construção de sistemas inteligentes sem que haja o comprometimento da privacidade. Dentre as soluções emergentes, o Aprendizado Federado (*Federated Learning* – FL) [McMahan et al., 2017] tem se apresentado como uma alternativa que possibilita o treinamento de modelos de aprendizado de maneira distribuída, sem que os usuários participantes precisem compartilhar seus dados privados em aberto [Neto et al., 2021, Naeem et al., 2022a, Qu et al., 2022].

A popularização do aprendizado federado, no entanto, é acompanhada de desafios. Um já conhecido na literatura é a presença de dados não-IID (*Independent and Identically Distributed*) para o treinamento de modelos. Um conjunto de dados é dito IID quando composto por amostras que são independentes entre si e ainda coletadas a partir da mesma distribuição probabilística, ou seja, as amostras não possuem influência ou correlação umas sobre as outras [Hsieh et al., 2020]. Os conjuntos de dados que não seguem essas características são chamados de não-IID. Sendo assim, dados coletados por dispositivos inteligentes, como *smartphones*, tendem a ser não-IID, já que cada usuário possui o seu próprio perfil de uso. A presença de dados não-IID afeta negativamente a construção de modelos de aprendizado de máquina [Kairouz et al., 2021] e seus efeitos no aprendizado federado não são diferentes. No aprendizado federado, cada cliente é responsável por treinar um modelo local cujo resultado é eventualmente agregado pelo servidor central. Visto que os conjuntos de dados locais são independentes entre si, o modelo agregado pode ter seu desempenho prejudicado, levando a propostas de técnicas que visam minimizar o impacto negativo dos dados não-IID [Souza et al., 2022].

Além do problema dos dados não-IID, um desafio que discutido recentemente é a influencia dos dispositivos retardatários, também conhecidos como *stragglers*, no aprendizado federado. Dispositivos com poder computacional reduzido podem atrasar o treinamento dos modelos, o que também exige novas técnicas de mitigação (*straggler mitigation*) [Reisizadeh et al., 2022, Kumar et al., 2022, Bochie et al., 2021b]. A heterogeneidade computacional dos dispositivos, especialmente no cenário *cross-device*, faz com que alguns clientes não sejam capazes de concluir seus treinamentos locais em tempo hábil. Esse fenômeno também afeta negativamente o desempenho do modelo global, atrasando a convergência [Bochie et al., 2021b, Asad et al., 2022]. Além da capacidade de processamento de cada cliente, outros fatores como a mobilidade dos usuários em redes móveis também pode aumentar o tempo de treinamento total, como consequência de comunicações intermitentes [Cao et al., 2021]. Logo, condições não ideais de redes, como atrasos e desconexões, impactam o tempo de convergência dos modelos no aprendizado federado, degradando o desempenho global do sistema [Bochie et al., 2021b].

Este trabalho propõe uma nova técnica para reduzir a latência do aprendizado federado em cenários *cross-device*. A proposta *Fastest-First Federated Learning* (3FL) seleciona os clientes considerando o tempo de treinamento individual. Diferentemente de abordagens encontradas na literatura, a proposta 3FL é implementada através de uma etapa de calibração prévia que substitui a tentativa de estimar o tempo esperado de treinamento de cada cliente através do uso de metadados. O desempenho da proposta 3FL é avaliado por meio de simulações com distribuições de dados realistas para o aprendizado federado. Os cenários simulados utilizam dois conjuntos de dados comumente utilizados na literatura para a avaliação de aplicações de aprendizado de máquina, mais precisamente o MNIST e o CIFAR-10. Os resultados obtidos demonstram uma redução

de latência de até 35% em relação aos métodos tradicionais. Além disso, os experimentos demonstram que a acurácia da classificação permanece similar ou até mesmo superior à observada sem o emprego da proposta, alcançando valores de até 97%. Vale notar que todas as simulações são avaliadas em distribuições de dados realistas, em que os clientes possuem conjuntos de dados amostrados de forma não-IID, reproduzindo características típicas do aprendizado federado *cross-device*.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 explica o 3FL. A Seção 4 discute a metodologia adotada e a configuração do ambiente experimental. Na Seção 5, os resultados são analisados. Por fim, a Seção 6 conclui este trabalho e apresenta possíveis direções de pesquisa.

2. Trabalhos Relacionados

Kumar et al. apresentam um esquema de codificação de dados federados visando diminuir o tempo total de treinamento dos modelos quando otimizados via regressão linear [Kumar et al., 2022]. Os autores buscam diminuir o impacto de dispositivos incapazes de enviar atualizações dos seus modelos dentro do intervalo estipulado para uma rodada de treinamento, sendo estes dispositivos conhecidos como *stragglers*, ou dispositivos retardatários. O artifício para usado para simular os dispositivos retardatários se baseia em limitar sinteticamente a capacidade de computação dos clientes simulados. Os autores concluem que o método de codificação proposto pode ser usado para acelerar a convergência do modelo, desde que o número de clientes envolvidos durante essa etapa seja apropriadamente ajustado.

Asad et al. propõem um protocolo baseado em recursos para melhorar o desempenho do aprendizado federado chamado *Clients' Eligibility Protocol* (CEP), no qual uma entidade de confiança é responsável por eleger quais clientes devem participar do treinamento em uma dada rodada [Asad et al., 2022]. O protocolo atribui pontuações positivas ou negativas conforme a contribuição de cada cliente durante as rodadas de treinamento. Ações como se manter disponível para o treinamento ou completar o treinamento em tempo hábil são recompensadas, enquanto ações como falhar em rodadas consecutivas ou apresentar modelos com grande desvio são punidas pelo protocolo. Os autores simulam o protocolo inicializando todos os clientes como possíveis participantes e concluem que o CEP atinge melhor desempenho médio ao longo das rodadas de treinamento devido à sua capacidade de eliminar os clientes retardatários durante as rodadas iniciais. No entanto, se faz necessário uma nova análise acerca do desempenho nos clientes que são eliminados do treinamento, visto que seus modelos podem ficar desatualizados indefinidamente. Uma possível melhoria para o protocolo CEP seria a reinclusão de clientes retardatários em rodadas posteriores.

Reisizadeh et al. criam o meta-algoritmo FLANP (*Federated Learning method with Adaptive Node Participation*) baseado na seleção de clientes com maior poder computacional para realizar o início do treinamento [Reisizadeh et al., 2022]. A proposta se destaca pela utilização de porções cada vez maiores de clientes no aprendizado e por usar um modelo treinado pelos clientes mais rápidos como *warm start* para o próximo grupo de clientes. Os autores comparam o novo algoritmo a três técnicas de referência: FedAvg, FedGATE e FedNova. A eficácia da proposta é comprovada para diferentes cenários, desde que o número de clientes iniciais seja apropriadamente selecionado. No entanto,

os autores avaliam seus resultados em um cenário onde os clientes coletam suas amostras a partir de uma única distribuição de dados, condição que pode não ser satisfeita em cenários *cross-device*.

Diferentemente dos trabalhos apresentados, este trabalho propõe uma nova forma de diminuir o impacto de dispositivos retardatários através da seleção de clientes no aprendizado federado *cross-device*, onde as distribuições de dados se apresentam de forma não-IID.

3. Fastest-First Federated Learning (3FL)

Como explicado na Seção 1, a heterogeneidade de dispositivos em cenários *cross-device* é um obstáculo para o treinamento de modelos de aprendizado federado de forma rápida e efetiva. Nesse sentido, técnicas como seleção de clientes e redução do impacto de dispositivos retardatários se apresentam como possíveis soluções para melhorar o desempenho dos modelos e reduzir a latência de treinamento. Este trabalho, portanto, propõe o *Fastest-First Federated Learning* (3FL) com o objetivo de reduzir a latência do treinamento no aprendizado federado e ainda melhorar o desempenho dos modelos finais. O 3FL se baseia na seleção de clientes usando como critério a velocidade de treinamento individual. Tal informação é obtida através de uma etapa de calibração de *timeout*, onde janelas de seleção de clientes progressivamente maiores são usadas para identificar os clientes mais rápidos.

No 3FL, apenas os clientes com maior velocidade de treinamento participam das primeiras rodadas do aprendizado federado, com o intuito de reduzir drasticamente a latência nas rodadas iniciais. Após o treinamento inicial, todos os clientes disponíveis são incluídos no treinamento, recebendo um modelo treinado pelos clientes mais rápidos. Dessa forma, os clientes com limitação computacional utilizam um modelo pré-treinado como um *warm start*, similarmente à técnica de transferência de aprendizado. A Figura 1 ilustra a proposta 3FL. Nela, as setas 1 representam a transmissão do modelo para os clientes, as setas 2 representam o treinamento local de cada cliente e as setas 3 representam as transmissões dos modelos de volta para o servidor. Note que na Etapa 1 apenas o subconjunto de clientes mais rápidos é selecionado, enquanto na Etapa 2 todos podem participar.

Este trabalho considera duas alternativas para definir quais clientes podem ser considerados rápidos ou lentos: (i) coletar metadados de cada usuário, como modelo do processador e tipo do enlace utilizado; ou (ii) realizar uma etapa inicial de calibração para obter heurísticas de tempo. Diferentemente de trabalhos anteriores [Reisizadeh et al., 2022], este trabalho implementa uma etapa de calibração antes do início do treinamento para a seleção de clientes, devido ao desafio adicional de estimar a velocidade do treinamento de acordo com informações de *hardware* dos clientes [Chen et al., 2020]. Mesmo que o uso de metadados seja utilizado em algumas configurações do aprendizado federado, essa técnica não faz parte do método FedAVG tradicional [Zhang et al., 2020]. Consequentemente, este trabalho não faz uso de metadados, a fim de buscar uma opção que não necessite de alterações nas implementações de estratégia diretamente no *framework* de aprendizado federado.

A etapa de calibração consiste em difundir inicialmente o modelo de treinamento para todos os clientes disponíveis e definir um tempo limite para o treinamento de uma

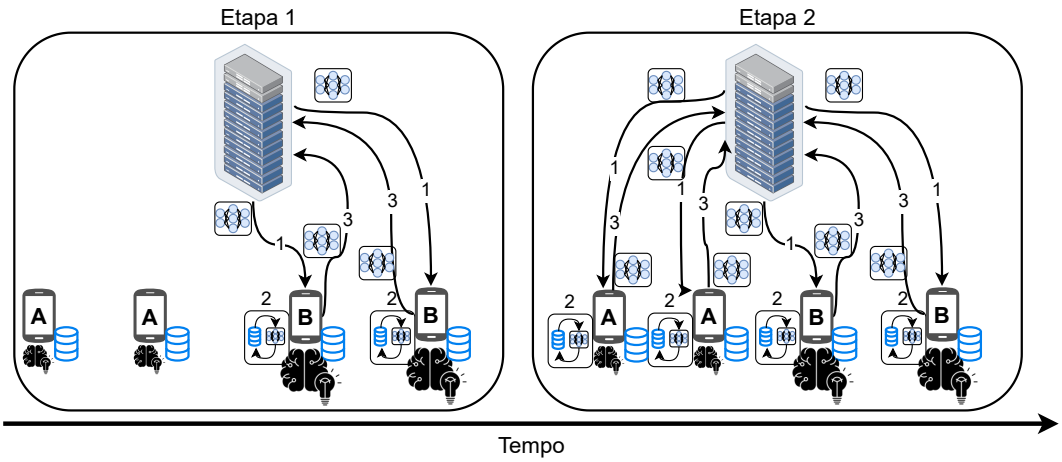


Figura 1. Proposta 3FL onde os dois clientes "B" possuem maior poder computacional que os dois clientes "A".

rodada. Caso um número insuficiente de usuários consiga realizar o treinamento dentro do tempo limite da rodada, este é duplicado e a etapa é repetida. Esse procedimento é executado até que o número pré-definido de usuários responda ao treinamento, sendo esses considerados os usuários mais rápidos para prosseguir ao início do treinamento. O tempo de resposta não depende exclusivamente da capacidade de treinamento do usuário, porém, para efeitos práticos, a latência de comunicação é considerada uma parcela menor do tempo total de treinamento [Kairouz et al., 2021]. A latência de transmissão foi estimada utilizando a ferramenta `iPerf3`[1] assim como o tamanho em MB dos modelos de aprendizado. Finalmente, um diagrama de execução incluindo a etapa de calibração de *timeout*, destacada em verde, pode ser visto na Figura 2.

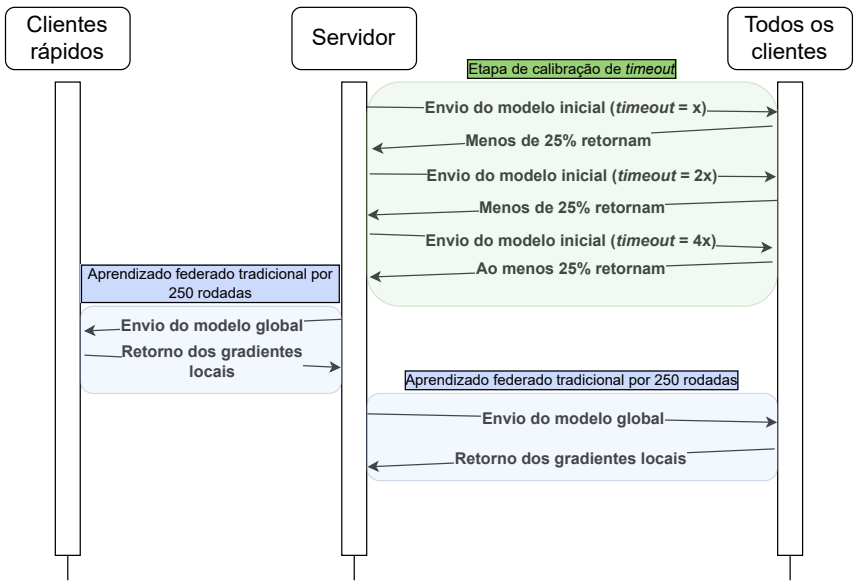


Figura 2. Diagrama de execução da proposta 3FL. Cenário com *timeout* inicial t selecionando ao menos 25% dos clientes totais para 250 rodadas de treinamento iniciais.

[1] Acessado em `https://iperf.fr/`.

Nota-se que a etapa de calibração depende de um limiar que define o percentual dos clientes selecionados para o treinamento, no qual o servidor conhece o número total de clientes disponíveis para o treinamento. A porção de clientes incluídos no início do treinamento impacta diretamente a latência. Logo, há um compromisso entre a redução de latência pretendida e o número de clientes participantes nas rodadas inicias de treinamento. Como o número de rodadas de treinamento é normalmente da ordem de centenas, o tempo necessário para realizar a etapa de calibração é negligenciável diante do tempo total de treinamento. Este trabalho demonstra que a proposta 3FL é capaz de reduzir o tempo total de treinamento, mesmo incluindo a etapa de calibração. Ressalta-se que após o treinamento inicial com os clientes mais rápidos, novas rodadas de treinamento são executadas com todos os clientes disponíveis.

4. Metodologia e Configuração dos Experimentos

Esta seção apresenta as ferramentas e as configurações usadas nos experimentos.

4.1. *Software* e *hardware* utilizados

Este trabalho utiliza o conjunto de ferramentas tipicamente empregado na exploração de dados em Python, como Numpy e scikit-learn. Ademais, os *frameworks* TensorFlow, Keras e Flower [Beutel et al., 2020], este último específico para implementação e gerenciamento de treinamento federado, foram usados. Note que todos os resultados foram obtidos utilizando ferramentas e *frameworks* de código aberto. A ideia é permitir a reprodutibilidade dos experimentos, inclusive a partir da disponibilização dos códigos usados em um repositório GitHub[2].

Em termos de *hardware*, diversos servidores foram utilizados durante as simulações, a fim de obter resultados mais rapidamente. Além disso, os tempos de execução médios em cada máquina foram usados para definir os valores de latência usados durante as simulações. A simulação dos clientes mais lentos e mais rápidos na análise de latência considerou o tempo de execução médio de cada cliente, como medido em máquinas com menor ou maior poder computacional, respectivamente. Em especial, a máquina de menor poder computacional foi um Intel Core i5-10400 2,90GHz com 32GB de RAM, enquanto a de maior poder computacional foi um AMD Epyc 7452 2.35GHz com 378GB de RAM.

4.2. Conjuntos de dados e modelos de aprendizado profundo

Este trabalho utiliza os conjuntos de dados MNIST [Lecun et al., 1998] e CIFAR-10 [Krizhevsky, 2012] nas simulações. O conjunto de dados MNIST é composto de 60.000 amostras de treinamento e 10.000 amostras de teste. Cada amostra consiste de uma imagem de 28 *pixels* de largura e 28 *pixels* de altura em escala de cinza, representando um dígito escrito à mão. Já o conjunto de dados CIFAR-10 é composto de 50.000 amostras de treinamento e 10.000 amostras de teste. Cada amostra consiste de uma imagem colorida de 32 *pixels* de largura e 32 *pixels* de altura representando 10 possíveis classes. Visto que os conjuntos de dados são imagens para classificação, a acurácia dos modelos foi utilizada como principal métrica de desempenho.

[2] `https://github.com/kaylani2/sbrc2024`.

A divisão de amostras entre os clientes é feita de forma não-IID, seguindo técnicas utilizadas na literatura [McMahan et al., 2017, Kumar et al., 2022], na qual cada cliente utiliza apenas amostras de até 2 rótulos diferentes. Essa divisão faz com que as distribuições de dados dos clientes sejam não-IID entre si. Em particular, as amostras do conjunto de dados MNIST não são balanceadas em relação aos rótulos. Dessa forma, em alguns cenários, alguns clientes recebem uma amostra adicional em relação aos outros que utilizam amostras das mesmas classes para garantir que não haja sobreposição de amostras entre os clientes.

Dois modelos de aprendizado profundo baseados em redes neurais convolucionais (*Convolutional Neural Network* – CNNs) foram utilizados para os experimentos, sendo designado um modelo para cada conjunto de dados.

4.3. Cenários avaliados e parâmetros de simulação

As simulações realizadas foram feitas inicialmente com 2, 5, 10, 15, 25 e 50 clientes para encontrar um cenário apropriado para a avaliação da proposta 3FL. Os outros hiperparâmetros utilizados foram: épocas de treinamento local (E) igual a 1; tamanho dos *batches* locais (B_c) igual a 64; número de rodadas de treinamento federado (R) igual a 500; taxa de aprendizado (η) igual a 10^{-2}; e o otimizador Adam. O número de épocas de treinamento local é reduzido para permitir uma melhor visualização do treinamento, já que as métricas escolhidas são avaliadas ao fim de cada rodada de treinamento.

5. Resultados

Primeiramente, uma avaliação do desempenho do algoritmo tradicional FedAvg [McMahan et al., 2017] foi feita em diferentes cenários a fim de encontrar uma configuração promissora para a avaliação da proposta 3FL. Em seguida, a proposta 3FL foi avaliada para diferentes configurações de hiperparâmetros usando os dois conjuntos de dados, o MNIST e o CIFAR-10.

5.1. FedAVG em dados não-IID no conjunto de dados MNIST

A Figura 3 apresenta os resultados em um cenário não-IID. A fim de manter as distribuições de dados inteiramente não-IID quando possível, os cenários com 2, 5 e 10 clientes foram configurados de forma particular. No cenário com 2 clientes, cada cliente recebe amostras de 5 rótulos diferentes. No cenário com 5 clientes, cada cliente recebe amostras de 2 rótulos diferentes. Já no cenário com 10 clientes, cada cliente recebe amostras de apenas 1 rótulo. Esse tipo de estratégia torna o problema mais desafiador, visto que os clientes buscam minimizar uma função custo de forma "egoísta". Vale destacar que mesmo em aplicações reais de aprendizado federado para classificação, os clientes podem possuir amostras de uma mesma classe [Sivek e Riley, 2022].

Naturalmente, devido à presença de 10 classes no conjunto de dados MNIST, divisões sem interseções de classes não são possíveis para cenários com menos de 10 clientes. Para os cenários com mais de 10 clientes, a divisão foi feita distribuindo apenas duas classes para cada cliente, porém cada cliente recebe apenas uma fração proporcional ao número total de clientes do conjunto de dados completo. No cenário de 50 clientes, por exemplo, cada cliente recebe apenas um décimo de suas duas respectivas classes. Essa distribuição representa mais fidedignamente as aplicações reais, sendo esses cenários o foco do resto deste trabalho.

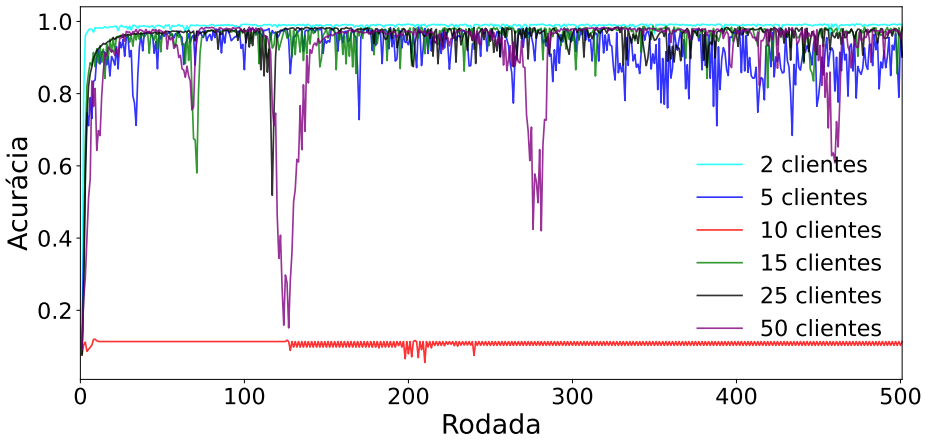


Figura 3. Desempenho de classificação do algoritmo FedAVG utilizando uma CNN personalizada no conjunto de dados MNIST com os seguintes hiperparâmetros: (i) taxa de aprendizado igual a 10^{-3}, (ii) tamanho dos *batches* locais igual a 64 amostras, (iii) épocas de treinamento local igual a 1. Cenário não-IID 2, 5, 10, 15, 25 e 50 clientes.

O alto desempenho de classificação obtido pela CNN personalizada, evidenciado na Figura 3, é resultado da capacidade de generalização do modelo de aprendizado utilizado, que não incide em sobreajuste. A exceção é aparente para o cenário com apenas 10 clientes. Porém, este resultado é compreendido ao destacar que cada cliente neste cenário possui apenas amostras de um rótulo, o que torna a função objetivo do modelo específica para cada cliente. Também é interessante apontar os vales de desempenho no cenário com 50 clientes, nas rodadas 120, 280 e 460, da Figura 3. Tais vales, porém, podem ser evitados com o uso de mecanismos como parada antecipada (*early stopping*). A aplicação desses mecanismos, porém, foi considerada tangente ao escopo deste trabalho.

Especificamente, a configuração com 25 clientes em cenário não-IID com modelo de aprendizado personalizado foi utilizada para avaliar os algoritmos propostos neste trabalho. Essa decisão foi tomada tendo em mente o cenário *cross-device*, que é tipicamente composto por dezenas ou centenas de dispositivos. Ademais, o tamanho do conjunto de dados é um fator limitante ao simular cenários com mais clientes, como observado na avaliação com 50 clientes exibida na Figura 3. Logo, o cenário com 25 clientes se mostrou aquele com a melhor relação custo-benefício para avaliação da proposta deste trabalho.

5.2. Desempenho do 3FL

Classificação com o MNIST: A Figura 4 compara o desempenho da proposta 3FL em diferentes cenários com o algoritmo FedAVG no conjunto de dados MNIST. As 250 primeiras rodadas de aprendizado foram executadas apenas com os clientes mais rápidos, enquanto o restante do treinamento foi realizado com todos os clientes. As simulações foram feitas com 25 clientes participantes. A figura mostra como o desempenho é afetado para diferentes quantidades de clientes mais rápidos na rede. Destaca-se que a curva preta pontilhada, que representa o cenário onde todos os clientes são incluídos no treinamento desde o início, se mantém sobre as outras curvas durante a etapa inicial de treinamento. No entanto, todos as configurações avaliadas do 3FL se mantêm a uma distância de no

máximo 5% de acurácia na etapa inicial, porém obtendo latências inferiores ao cenário tradicional.

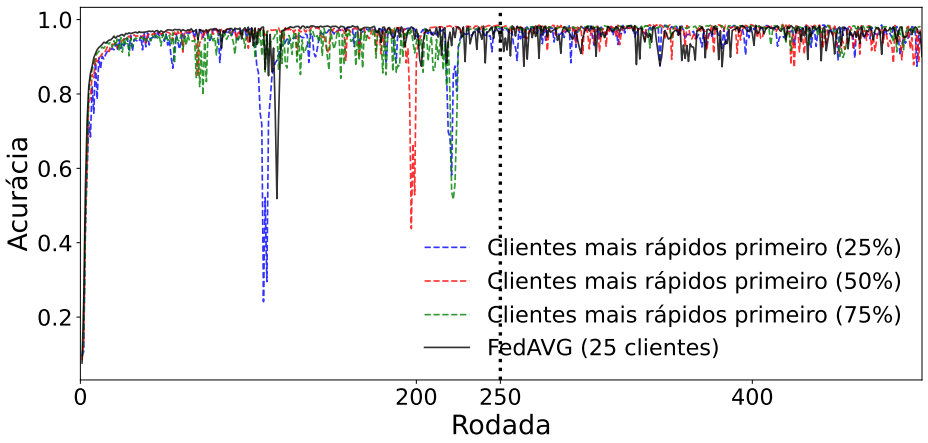


Figura 4. Desempenho de classificação do algoritmo 3FL para diferentes latências de clientes participantes no conjunto de dados MNIST.

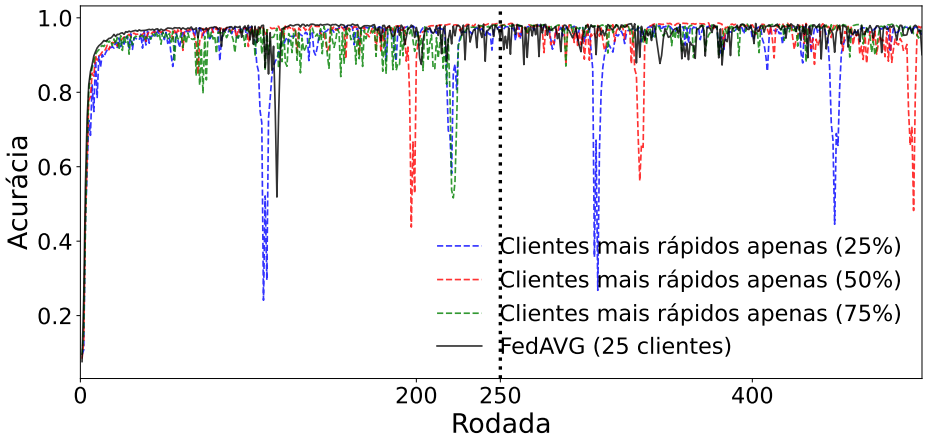


Figura 5. Treinamento com apenas os clientes mais rápidos por 500 rodadas no conjunto de dados MNIST.

Uma possível intuição ao observar a Figura 4 é que seria possível reduzir ainda mais a latência total ao manter apenas os clientes mais rápidos no treinamento, visto que a convergência do modelo parece ocorrer ainda nas primeiras rodadas. Esse experimento pode ser visto na Figura 5, em que a não incorporação dos clientes mais lentos manteve a degradação do modelo ao encontrar mínimos locais durante o treinamento. Esse resultado é evidenciado pela persistência da presença de vales na acurácia de treinamento após a rodada 250. Na Figura 4 tais vales não ocorrem mais após a incorporação de todos os clientes após a rodada 250. Ademais, fundamentalmente, o objetivo do aprendizado federado é produzir um modelo que possa ser utilizado por todos os clientes participantes. Sendo assim, é importante incluir o restante dos clientes no treinamento para que o desempenho final do modelo seja satisfatório para todos.

Classificação com o CIFAR-10: As mesmas avaliações da proposta 3FL são feitas com o conjunto de dados CIFAR-10, juntamente com a simulação do algoritmo tradicional FedAVG. A Figura 6 compara o desempenho da proposta 3FL em diferentes cenários com o algoritmo FedAVG no conjunto de dados CIFAR-10. Assim como nos experimentos anteriores, as 250 primeiras rodadas compreendem apenas os clientes mais rápidos e as 250 rodadas finais utilizam todos os clientes. Nessa figura, pode-se observar, primeiramente, que as acurácias obtidas pelos modelos são inferiores àquelas obtidas no conjunto de dados MNIST. Isso é esperado, visto que o conjunto CIFAR-10 é mais "desafiador" para ser classificado com modelos de arquiteturas similares. Tendo isso em vista, também é possível notar que apenas o desempenho obtido pela configuração com 50% dos clientes mais rápidos é superior ao do algoritmo tradicional FedAVG. Este resultado reforça a contribuição da proposta 3FL, porém também destaca que o impacto do número de clientes selecionados não é necessariamente linear em relação ao desempenho final.

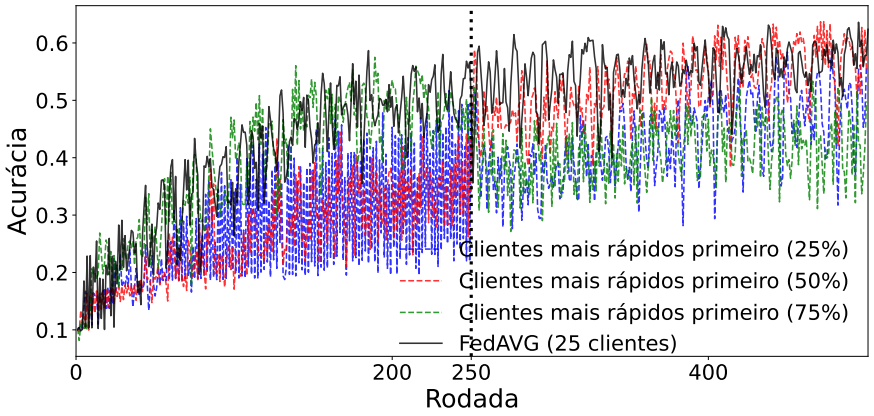
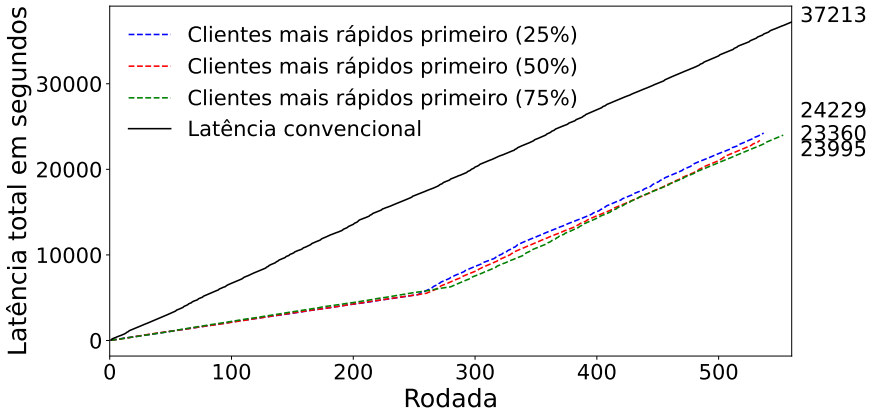


Figura 6. Desempenho de classificação do algoritmo 3FL para diferentes latências de clientes participantes no conjunto de dados CIFAR-10.

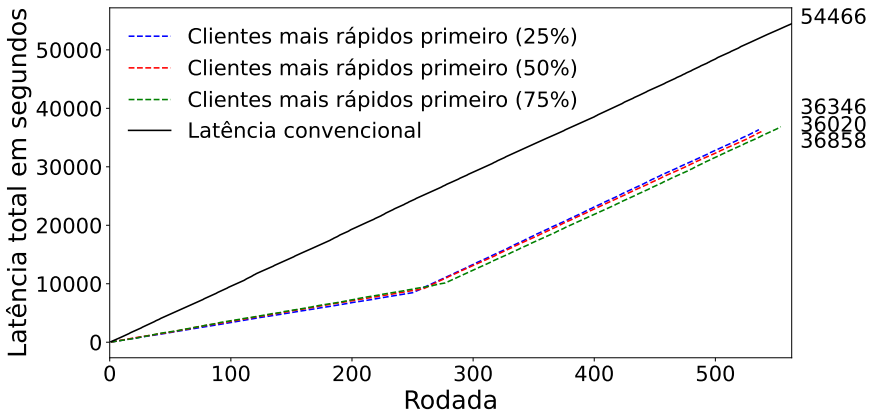
Latências com o MNIST e o CIFAR-10: Os valores de latência usados na simulação foram medidos a partir da simulação em dois sistemas com capacidades computacionais diferentes, como descrito na Seção 4.3. No conjunto de dados MNIST, um dos servidores utilizados atingiu um tempo médio de execução de 61 segundos, enquanto o outro sistema atingiu um tempo médio de execução de 23 segundos. Já no conjunto de dados CIFAR-10, um dos servidores utilizados atingiu um tempo médio de execução de 92 segundos, enquanto o outro sistema atingiu um tempo médio de 33 segundos. Os dois pares de valores observados foram usados nos experimentos como tempo médio dos clientes mais lentos e clientes mais rápidos, respectivamente.

A Figura 7(a) apresenta uma comparação entre a latência obtida no treinamento federado "tradicional" e a latência obtida com o 3FL no MNIST. Na figura, os clientes com maior poder computacional foram escolhidos para treinar durante as 250 rodadas iniciais. Como consequência, pode-se observar latências iniciais menores no cenário 3FL. Esse ganho em velocidade de execução é desejável, desde que não prejudique o desempenho do modelo de aprendizado. A Figura 7(b) apresenta resultados semelhantes para o CIFAR-10, porém com diferentes valores finais nos tempos de execução. Essa diferença é esperada, visto que os conjuntos de dados são diferentes e o tempo de treinamento de

cada cliente pode variar de acordo.



(a) Resultados com o conjunto de dados MNIST.



(b) Resultados com o conjunto de dados CIFAR-10.

Figura 7. Comparação de latências nos cenários simulados. As simulações foram feitas com 25 clientes, em que diferentes combinações de clientes atingem menor latência durante uma rodada e são usados durante etapa inicial do 3FL, enquanto o restante do treinamento é feito utilizando todos os clientes.

5.3. Comparação

A Figura 8 compara o melhor desempenho obtido em cada cenário avaliado. As acurácias apresentadas no gráfico são calculadas através da média de desempenho nas últimas 10 rodadas para cada configuração. Essa representação garante uma comparação mais "justa" entre os cenários, visto que há variações de até 5% de desempenho entre rodadas consecutivas de treinamento. Uma alternativa é utilizar o maior valor de acurácia nas últimas rodadas, similar à parada antecipada comumente implementada em sistemas reais. No entanto, como a técnica de parada antecipada não foi utilizada nos experimentos, a representação por média foi escolhida. Adicionalmente, os algoritmos FedAVGM [Hsu et al., 2019] e FedTrimmedAVG [Xiang et al., 2022], que se mostram mais apropriados para cenários com distribuições de dados não-IID, foram incluídos na avaliação final.

A Figura 8 mostra que a acurácia do 3FL é equiparável ao desempenho da técnica tradicional (FedAVG) e dos algoritmos FedAVGM e FedTrimmedAVG. Logo, a de redução de latência de treinamento e a redução do uso de recursos computacionais da rede não se apresentam em detrimento do desempenho de classificação do modelo final. Nota-se também que o número de clientes selecionados para o início do treinamento não possui um impacto óbvio nas três propostas. Para o cenário 3FL com 50% de clientes iniciais com o conjunto de dados MNIST, por exemplo, o desempenho é inferior aos cenários com o 3FL com mais e com menos clientes iniciais, enquanto o mesmo cenário com o conjunto de dados CIFAR-10 apresenta desempenho superior.

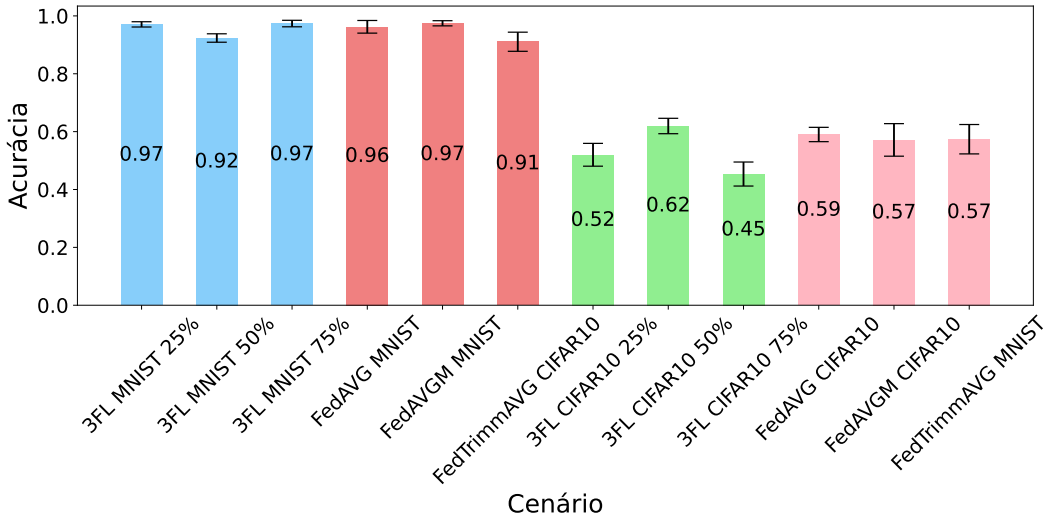


Figura 8. Desempenho da classificação nos cenários avaliados.

A Figura 9 apresenta as latências obtidas em todos os cenários simulados. Visto que os algoritmos FedAVGM e FedTrimmed não fazem alegações sobre reduções de latência de treinamento [Hsu et al., 2019, Xiang et al., 2022], seus resultados foram considerados afins aos do algoritmo original FedAVG e foram devidamente omitidos. Nela, nota-se a efetividade da proposta 3FL em reduzir a latência total de treinamento para os diferentes cenários em que apenas os clientes mais rápidos são incluídos.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma nova forma de implementar estratégias de seleção de clientes para reduzir a latência de treinamento durante o aprendizado federado. A proposta criada foi avaliada em dois conjuntos de dados com distribuições representativas de cenários de aprendizado federado *cross-device*. Os experimentos realizados demonstraram que é possível obter latências até 35% menores durante o treinamento ao selecionar clientes mais rápidos com a técnica 3FL. Também foi demonstrado que a técnica avaliada foi capaz de atingir desempenhos de classificação superiores à abordagem tradicional.

Como trabalho futuro, pretende-se expandir a validação da proposta para situações onde a intermitência do meio sem fio se mostra como um desafio extra. Adicionalmente, também pretende-se implantar os algoritmos propostos em uma rede de computadores real, onde clientes sem fio contribuem para o treinamento federado de forma não simulada.

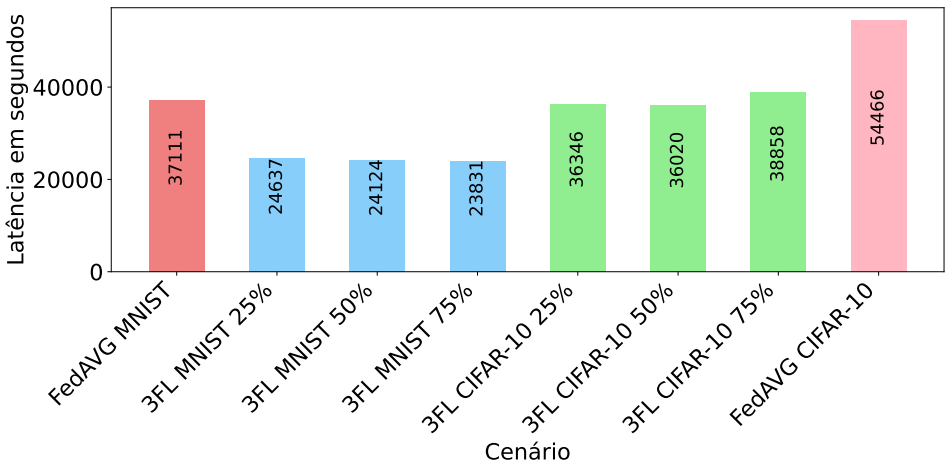


Figura 9. Latências totais de treinamento nos cenários avaliados.

Finalmente, também pretende-se avaliar o impacto de ajustes dinâmicos dos algoritmos após um determinado número de rodadas de treinamento.

Referências

Asad, M., Otoum, S. e Shaukat, S. (2022). Resource and heterogeneity-aware clients eligibility protocol in federated learning. Em *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, p. 1140–1145.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Kwing, H. L., Parcollet, T., Gusmão, P. P. d. e Lane, N. D. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Bochie, K., Gilbert, M. S., Gantert, L., Barbosa, M. S., Medeiros, D. S. e Campista, M. E. M. (2021a). A survey on deep learning for challenged networks: Applications and trends. *Journal of Network and Computer Applications*, 194:103213.

Bochie, K., Sammarco, M., Detyniecki, M. e Campista, M. (2021b). Análise do aprendizado federado em redes móveis. Em *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 71–84, Porto Alegre, RS, Brasil. SBC.

Cao, Y., Maghsudi, S. e Ohtsuki, T. (2021). Mobility-aware routing and caching: A federated learning assisted approach. Em *ICC 2021 - IEEE International Conference on Communications*, p. 1–6.

Chen, Y., Ning, Y., Slawski, M. e Rangwala, H. (2020). Asynchronous online federated learning for edge devices with non-iid data. Em *2020 IEEE International Conference on Big Data (Big Data)*, p. 15–24.

Hsieh, K., Phanishayee, A., Mutlu, O. e Gibbons, P. (2020). The non-IID data quagmire of decentralized machine learning. Em III, H. D. e Singh, A., editores, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, p. 4387–4398. PMLR.

Hsu, H., Qi, H. e Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. Em *Neurips Workshop on Federated Learning*.

Kairouz et al. (2021). Advances and open problems in federated learning.

Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. *University of Toronto*.

Kumar, S., Schlegel, R., Rosnes, E. e Amat, A. G. i. (2022). Coding for straggler mitigation in federated learning. Em *ICC 2022 - IEEE International Conference on Communications*, p. 4962–4967.

Lecun, Y., Bottou, L., Bengio, Y. e Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

McMahan, B., Moore, E., Ramage, D., Hampson, S. e y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Em Singh, A. e Zhu, J., editores, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, p. 1273–1282, Fort Lauderdale, FL, USA. PMLR.

Naeem, A., Anees, T., Naqvi, R. A. e Loh, W.-K. (2022a). A comprehensive analysis of recent deep and federated-learning-based methodologies for brain tumor diagnosis. *Journal of Personalized Medicine*, 12(2).

Naeem, M., Jamal, T., Diaz-Martinez, J., Butt, S. A., Montesano, N., Tariq, M. I., De-la Hoz-Franco, E. e De-La-Hoz-Valdiris, E. (2022b). Trends and future perspective challenges in big data. Em Pan, J.-S., Balas, V. E. e Chen, C.-M., editores, *Advances in Intelligent Data Analysis and Applications*, p. 309–325, Singapore. Springer Singapore.

Neto, H. C., Mattos, D. e Fernandes, N. (2021). Fedsa: Arrefecimento simulado federado para a aceleração da detecção de intrusão em ambientes colaborativos. Em *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 280–293, Porto Alegre, RS, Brasil. SBC.

Qu, Y., Uddin, M. P., Gan, C., Xiang, Y., Gao, L. e Yearwood, J. (2022). Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.*, 55(4).

Reisizadeh, A., Tziotis, I., Hassani, H., Mokhtari, A. e Pedarsani, R. (2022). Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity. *IEEE Journal on Selected Areas in Information Theory*, 3(2):197–205.

Rimol, M. (2022). Gartner identifies top five trends in privacy through 2024. *Gartner*.

Sivek, G. e Riley, M. (2022). Spatial model personalization in gboard. *Proc. ACM Hum.-Comput. Interact.*, 6(MHCI).

Souza, L., Camilo, G., Sammarco, M., Campista, M. e Costa, L. (2022). Aprendizado federado com agrupamento hierárquico de clientes para aumento da acurácia. Em *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 545–558, Porto Alegre, RS, Brasil. SBC.

Xiang, W. T., Shao, M., Fu, Y., Jia, R., Lin, F. e Zheng, Z. (2022). FEDERATED LEARNING FRAMEWORK BASED ON TRIMMED MEAN AGGREGATION RULES.

Zhang, Q., Palacharla, P., Sekiya, M., Suga, J. e Katagiri, T. (2020). Demo: A blockchain based protocol for federated learning. Em *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, p. 1–2.