REDUÇÃO DO EFEITO DE DISPOSITIVOS RETARDATÁRIOS NO APRENDIZADO FEDERADO EM CENÁRIOS CROSS-DEVICE

Kaylani Bochie

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Miguel Elias M. Campista

Rio de Janeiro
Dezembro de 2023

REDUÇÃO DO EFEITO DE DISPOSITIVOS RETARDATÁRIOS NO APRENDIZADO FEDERADO EM CENÁRIOS CROSS-DEVICE

Kaylani Bochie

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientador: Miguel Elias M. Campista

Aprovada por: Prof. Luís Henrique Maciel Kosmalski Costa
Profa. Luciana Arantes

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2023

Bochie, Kaylani

Redução do Efeito de Dispositivos Retardatários no Aprendizado Federado em Cenários Cross-Device/Kaylani Bochie. – Rio de Janeiro: UFRJ/COPPE, 2023.

XIII, 61 p.: il.; 29, 7cm.

Orientador: Miguel Elias M. Campista

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2023.

Referências Bibliográficas: p. 43 – 56.

1. Aprendizado federado. 2. Redes sem fio. 3. Redes móveis. I. Elias M. Campista, Miguel. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

REDUÇÃO DO EFEITO DE DISPOSITIVOS RETARDATÁRIOS NO APRENDIZADO FEDERADO EM CENÁRIOS CROSS-DEVICE

Kaylani Bochie

Dezembro/2023

Orientador: Miguel Elias M. Campista

Programa: Engenharia Elétrica

Esta dissertação propõe e compara o desempenho de três novas técnicas para melhorar o desempenho do aprendizado federado e reduzir a latência total de treinamento. A primeira proposta, *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 segunda proposta, *Data-Oriented Federated Learning* (DOFL), utiliza o tamanho dos conjuntos de dados locais para selecionar clientes que possam ter maior impacto no início do treinamento. Finalmente, a terceira proposta, *Hybrid Federated Learning* (Hybrid-FL), mescla os dois conceitos anteriores em uma solução híbrida, onde o tempo de treinamento e a influência de cada cliente são utilizados para melhorar o desempenho do aprendizado federado em relação ao estado da arte. Todas as propostas são avaliadas por meio de simulações utilizando distribuições de dados e configurações de clientes realistas para o cenário de aprendizado federado *cross-device*.

Os resultados obtidos demonstram que é possível obter reduções na latência de treinamento de até 35% ao obter desempenhos de classificação equiparáveis, ou até mesmo superiores, atingindo acurácias de até 98% em problemas de classificação de imagens, quando comparado a técnicas de aprendizado tradicionais. Os resultados também reduzem o consumo de recursos computacionais, garantindo que a convergência do modelo de aprendizado apresente menor variação na perda.

Finalmente, um dos cenários simulados foi executado na forma de prova de conceito, com um servidor em nuvem e clientes distribuídos entre a borda da rede e redes móveis.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

STRAGGLER MITIGATION ON CROSS-DEVICE FEDERATED LEARNING

Kaylani Bochie

December/2023

Advisor: Miguel Elias M. Campista

Department: Electrical Engineering

This dissertation proposes and compares the performance of three novel techniques to enhance federated learning performance and reduce overall latency. The first proposal, namely, Fastest-First Federated Learning (3FL), is done by selecting the faster clients at the beginning of training to reduce training latency and mitigate the impact of straggler devices. The second proposal, namely, Data-Oriented Federated Learning (DOFL), uses the size of local datasets to select clients that might have a more significant impact at the start of training. Finally, the third proposal, namely, Hybrid Federated Learning (Hybrid-FL), combines the two previous concepts into a hybrid solution where training time and each client's influence are leveraged to improve federated learning performance compared to state-of-the-art methods. All proposals are evaluated through simulations using realistic data distributions and client configurations of cross-device federated learning scenarios.

The results demonstrate the potential to achieve up to a 35% reduction in overall training latency while achieving comparable or even superior classification performances, reaching accuracies of up to 98% in image classification problems when compared to traditional learning techniques. Additionally, a reduction in computational resource consumption was observed while ensuring that the model's learning convergence exhibited less variance.

Lastly, one of the simulated scenarios was executed as a proof of concept, employing a cloud server and clients distributed amongst the network's edge and mobile networks.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

B_c	Tamanho dos *batches* locais, p. 6, 7, 28
E	Épocas de treinamento local, p. 6, 7, 28
K	Número de clientes participantes em uma rodada do aprendizado federado, p. 7
N	Número de clientes, p. 7, 28
R	Número de rodadas de treinamento federado, p. 28
S_t	Clientes selecionados para uma rodada de treinamento, p. 6
T	Tempo de treinamento total, p. 7
T^r	Tempo de treinamento de uma rodada, p. 6
T_s^{ag}	Tempo de agregamento agregação do servidor, p. 7
η	Taxa de aprendizado, p. 6, 28
$\mathcal{D}$	Conjunto de dados, p. 8
$\mathcal{I}$	Espaço de identificadores de amostras, p. 8
$\mathcal{X}$	Espaço de atributos, p. 8
$\mathcal{Y}$	Espaço de rótulos, p. 8
r	Uma rodada do aprendizado federado, p. 7
t_n	Tempo total de um cliente, p. 6
t_n^t	Tempo de treinamento de um cliente, p. 6
t_n^{cs}	Tempo de transmissão cliente-servidor de um cliente, p. 6
t_n^{sc}	Tempo de transmissão servidor-cliente de um cliente, p. 6
w	Conjunto de pesos, p. 6
w_r	Conjunto de pesos em uma rodada, p. 6

Lista de Abreviaturas

3FL	Fastest-First Federated Learning, p. iv, v, 1, 19
ACM	Association for Computing Machinery, p. 12
AI	Artificial Intelligence, p. 3
AP	Access Point, p. 13
AWS	Amazon Web Services, p. 40
CEFL	Communication-Efficient Federated Learning, p. 16
CEP	Clients' Eligibility Protocol, p. 14
CNN	Convolutional Neural Network, p. 4
CNTK	Microsoft Cognitive Toolkit, p. 24
DNN	Deep Neural Network, p. 3
DOFL	Data-Oriented Federated Learning, p. iv, v, 1, 21
FEEL	Federated Edge Learning, p. 9
FLANP	Federated Learning method with Adaptive Node Participation, p. 16
FL	Federated Learning, p. 1
FedAVG	FederatedAveraging, p. 6
GAN	Generative Adversarial Network, p. 17
HFL	Horizontal Federated Learning, p. 7
Hybrid-FL	Hybrid Federated Learning, p. iv, 1
IDS	Intrusion Detection System, p. 1
IEEE	Institute of Electrical and Electronics Engineers, p. 12

IID	Independent and Identically Distributed, p. 10
INP	In-Network Processing, p. 15
IoT	Internet of Things, p. 9
MAC	Multiply-Accumulate, p. 14
MCKP	Multiple-Choice Knapsack Problem, p. 15
MDP	Model Download Protocol, p. 15
MNIST	Modified National Institute of Standards and Technology, p. 25
MUP	Model Upload Protocol, p. 15
NLP	Natural Language Processing, p. 5
PCA	Principal Component Analysis, p. 17
PS	Parameter Server, p. 5
RNN	Recurrent Neural Network, p. 4
SOL	SBC-OpenLib, p. 12
UAV	Unmanned Aerial Vehicle, p. 15
VFL	Vertical Federated Learning, p. 8

Capítulo 1

Introdução

O volume de dados gerados por dispositivos inteligentes tem crescido vertiginosamente e isto motiva o desenvolvimento de soluções que possam se beneficiar de grandes massas de dados [4, 5], como aplicações baseadas em aprendizado profundo [1]. Paralelamente, é possível observar preocupações cada vez maiores acerca da segurança da informação e, especificamente, da privacidade dos dados [6–8]. Essas tendências impulsionam o desenvolvimento de novas soluções, nas quais dados de usuários podem ser usados para a construção de sistemas sem que haja comprometimento de privacidade.

Dentre as soluções emergentes, o Aprendizado Federado (*Federated Learning* – FL) [9] tem se apresentado como uma solução emergente que possibilita o treinamento de modelos de aprendizado de maneira distribuída, sem que os usuários participantes precisem compartilhar seus dados privados. Aplicações como Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* – IDS) [10, 11], detecção de tumores [12] e *blockchain* [13] têm se aproveitado do paradigma federado para possibilitar novas implementações que garantam privacidade dos dados.

A popularização do aprendizado federado, no entanto, é acompanhada de novos desafios. Especificamente, dispositivos retardatários, também conhecidos como *stragglers*, podem atrasar o treinamento dos modelos de aprendizado federado [14, 15], enquanto as particularidades de distribuições de dados de cenários de aprendizado federado dificultam a avaliação de novas soluções [16].

Tendo em mente os problemas mencionados, esta dissertação propõe três novas técnicas para reduzir a latência do aprendizado federado em cenários *cross-device* sem prejuízo de desempenho. As propostas *Fastest-First Federated Learning* (3FL) e *Data-Oriented Federated Learning* (DOFL) utilizam a velocidade de treinamento e o volume de dados dos clientes, respectivamente, para selecionar os clientes mais apropriados para o treinamento. Já a terceira proposta, *Hybrid Federated Learning* (Hybrid-FL) mescla as duas propostas anteriores para criar um método de seleção híbrido. Todas as três técnicas são avaliadas por meio de simulações com distribuições

de dados realistas para o aprendizado federado. Os resultados obtidos indicam uma redução de latência de até 35% em relação aos métodos tradicionais, com desempenho de classificação equiparável ou até mesmo superior. Adicionalmente, todas as três propostas são capazes de reduzir o consumo de recursos da rede através da seleção de clientes. Ademais, todas as três propostas são avaliadas em distribuições de dados realistas para o cenário de aprendizado federado *cross-device*.

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 explica alguns conceitos-chave necessários para o entendimento deste trabalho. O Capítulo 3 discorre sobre a bibliografia avaliada e sobre os trabalhos relacionados. O Capítulo 4 detalha as propostas e as contribuições deste trabalho. O Capítulo 5 explica as ferramentas utilizadas e as configurações dos experimentos realizados. O Capítulo 6 apresenta e analisa os resultados obtidos. Por fim, o Capítulo 7 conclui este trabalho e apresenta futuras direções de pesquisa.

Capítulo 2

Conceitos-Chave de Aprendizado Profundo e Aprendizado Federado

Este capítulo descreve os conceitos necessários para o entendimento desta dissertação, desde redes neurais profundas, até o aprendizado federado e suas principais características e desafios.

2.1 Redes Neurais Profundas

As Redes Neurais Profundas (*Deep Neural Networks* – DNNs) são modelos de Inteligência Artificial (*Artificial Intelligence* – AI) capazes de "aprender" a realizar tarefas, como classificação e predição, através do consumo de dados durante um processo denominado treinamento. A construção de redes neurais é inspirada nas estruturas neuronais que constituem o cérebro humano. O termo "profundo" se refere ao fato de redes neurais utilizarem diversas camadas de processamento para aumentar sua capacidade de aprendizado. A Figura 2.1 apresenta uma visão em alto nível de uma rede neural profunda. O processo de treinamento dos modelos de aprendizado é um assunto explorado extensamente na literatura [1, 17] e não é considerado o foco deste trabalho.

É necessário um grande volume de dados de treinamento para garantir que modelos de aprendizado profundo atinjam alto desempenho em tarefas complexas [1]. Usualmente, dados são enviados a um servidor centralizado responsável por treinar um modelo de aprendizado que é então distribuído para diferentes dispositivos realizarem tarefas como classificação e predição. Essa abordagem, naturalmente, requer que dispositivos interessados em contribuir para a construção de um modelo de aprendizado compartilhem seus dados com o servidor central.

A seguir, esta dissertação apresenta as duas arquiteturas de redes neurais mais usadas em soluções voltadas para redes de computadores [1].

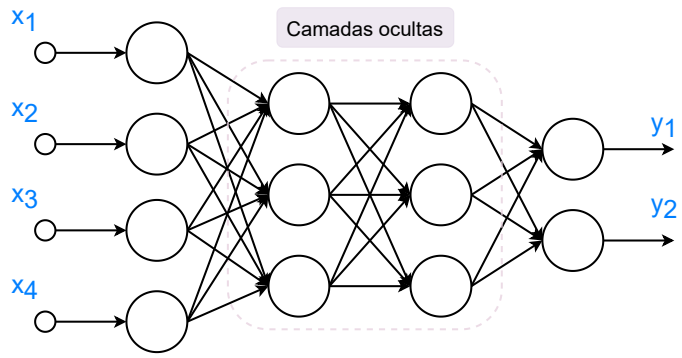


Figura 2.1: Representação simplificada de uma rede neural profunda. Os dados de entrada são transferidos da esquerda para a direita através das diferentes camadas de processamento da rede. A figura foi reproduzida de Bochie et al. [1].

2.1.1 Redes Neurais Convolucionais

Dentre os principais tipos de redes neurais, pode-se destacar as Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNNs), que obtém melhor desempenho em tarefas relacionadas ao tratamento de imagem e de vídeo. Essa propriedade se dá devido ao uso de camadas ou filtros convolucionais, responsáveis por aprender representações diferentes dos dados. Em cenários de classificação de imagem, as camadas posteriores das CNNs, então, interpretam os dados recebidos das camadas anteriores e fazem a classificação dos dados [1]. A Figura 2.2 apresenta uma visão simplificada de uma rede neural convolucional.

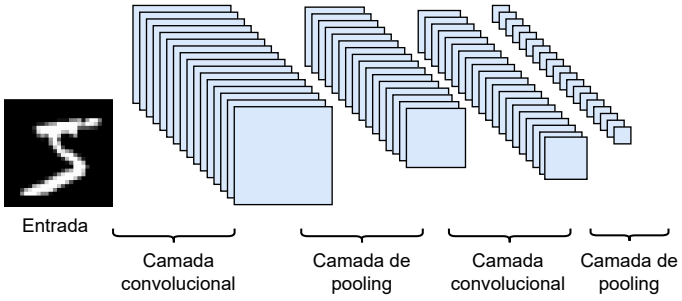


Figura 2.2: Representação simplificada de uma rede neural convolucional. Reproduzido de Bochie et al. [1].

2.1.2 Redes Neurais Recorrentes

Redes Neurais Recorrentes (*Recurrent Neural Networks* – RNNs) diferem dos outros tipos de redes neurais devido ao uso de conexões recorrentes que permitem a persistência de informação ao longo do tempo [1]. As conexões recorrentes permitem que as RNNs capturem dependências temporais nos dados, o que as tornam uma ferramenta apropriada para tarefas como predição de séries temporais e Processamento

de Linguagem Natural (*Natural Language Processing* – NLP) [1]. A Figura 2.3 apresenta uma visão simplificada de uma rede neural recorrente.

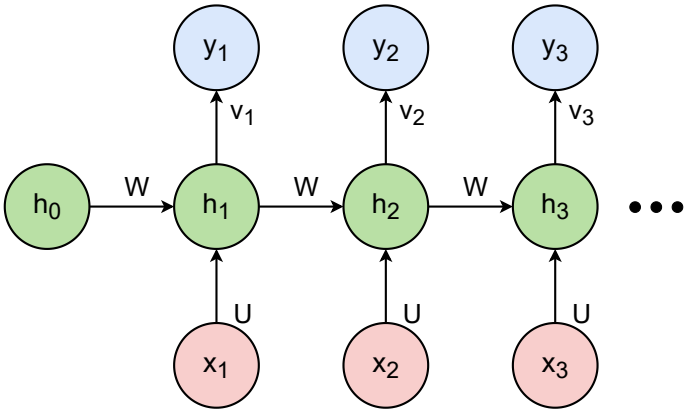


Figura 2.3: Representação simplificada de uma rede neural recorrente. Reproduzido de Bochie et al. [1].

2.2 Aprendizado Federado

Como descrito anteriormente, as redes neurais profundas foram, por muito tempo, treinadas em um servidor centralizado que recebe os dados enviados pelos usuários. O envio de dados, porém, representa possíveis brechas de segurança, visto que os dados necessários para treinar modelos de aprendizado podem conter informações pessoais ou sensíveis [1, 16]. Essas preocupações de segurança levam ao desenvolvimento de novas técnicas para preservar a privacidade de dados, como o aprendizado federado [9].

O conceito de aprendizado federado, inicialmente proposto por McMahan et al., consiste em treinar um modelo de aprendizado de forma distribuída, onde cada participante é responsável por treinar localmente o modelo de aprendizado em seu conjunto de dados [9]. Esse paradigma se dá, em contrapartida ao aprendizado de máquina "clássico" ou centralizado, onde cada dispositivo deve enviar seus dados a um servidor central que é responsável por realizar o treinamento e por difundir o modelo treinado. No aprendizado federado, os modelos de cada cliente são enviados a um servidor central, chamado por alguns autores de Servidor de Parâmetros (*Parameter Server* – PS) [18], responsável por gerar um modelo global usando como entrada os modelos recebidos dos clientes ao final de cada rodada de treinamento. O modelo global é o resultado da agregação dos modelos individuais dos clientes. O aprendizado federado atua na preservação da privacidade dos clientes, visto que torna desnecessária a troca de dados privados entre dispositivos. Uma representação simplificada do cenário de aprendizado federado pode ser vista na Figura 2.4. Os passos 1 e 2 na figura ao lado esquerdo representam a transferência do conjunto de

dados para o servidor e a transferência do modelo de aprendizado treinado para os clientes, respectivamente. Já os passos 1, 2 e 3 na figura ao lado direito representam a transferência de um modelo compartilhado para os clientes, o treinamento de cada cliente com seu conjunto de dados local e a transferência dos modelos atualizados de volta para o servidor, respectivamente.

Já o algoritmo de média federada (*FederatedAveraging* – FedAVG), originalmente proposto por McMahan et al. [9], pode ser visto no Algoritmo 2.1. O cenário do exemplo se refere aos clientes através do índice $n \in \{1, \ldots, N\}$, sendo que B_c denota o tamanho dos *batches* locais, E o número de épocas locais de treinamento, ρ_n o conjunto de dados local do n-ésimo cliente e η a taxa de aprendizado local [9]. Além disso, o parâmetro C define o número de clientes participantes de uma dada rodada de treinamento, a_n representa o número de amostras de um dado cliente n, w representa um conjunto de pesos, w_r representa o conjunto de pesos em uma rodada r e S_t representa o conjunto de clientes selecionados para uma rodada de treinamento.

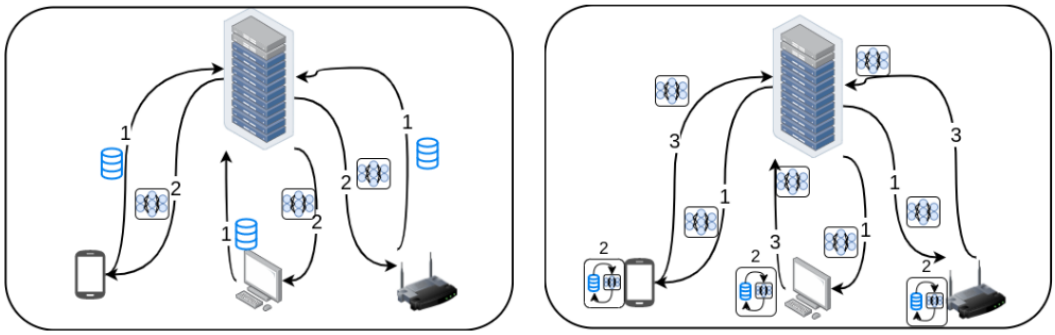


Figura 2.4: Exemplo da diferença entre o aprendizado de máquina "clássico" e o aprendizado federado. Reproduzido de Bochie et al. [2].

É possível calcular o tempo necessário para o n-ésimo cliente receber o modelo de aprendizado do servidor, realizar seu treinamento local e fazer o *upload* do modelo atualizado para o servidor através da Equação 2.1. Nela, o tempo total de um cliente é definido como a soma dos tempos de transmissão e o tempo de treinamento do cliente. Onde, t_n representa o tempo total de um cliente, t_n^{sc} representa o tempo de transmissão servidor-cliente de um cliente, t_n^{cs} representa o tempo de transmissão cliente-servidor de um cliente e t_n^t representa o tempo de treinamento de um cliente.

$$t_n = t_n^{sc} + t_n^t + t_n^{cs} \tag{2.1}$$

Também é possível calcular o tempo total necessário para realizar uma rodada de treinamento através da soma do tempo de agregação dos modelos no servidor e a maior latência dentre os clientes participantes. Esse cálculo pode ser visto na Equação 2.2, onde T^r representa o tempo de treinamento de uma rodada, $\mathcal{N} = 1..N$

Algoritmo 2.1: Algoritmo FedAVG. Adaptado de McMahan et al. [9].

Entrada: w_0
para cada *rodada em* $r = 1, 2, ...$ **faça**
  $m \leftarrow max(K \cdot N, 1)$
  $S_r \leftarrow$ (conjunto aleatório de m clientes)
  para cada *cliente* $k \in S_r$ ***em paralelo*** **faça**
    $w_{r+1}^n \leftarrow$ ClientUpdate (r, w_r)
  fim
  $w_{r+1} \leftarrow \sum_{n=1}^{N} \frac{a_n}{N} w_{r+1}^n$
fim
ClientUpdate (n, w):
$\beta \leftarrow$ (divide ρ_n em *batches* de tamanho B_c)
para cada *época local* i *de* 1 *até* E **faça**
  para cada *batch* $b \in \beta$ **faça**
    $w \leftarrow w - \eta \nabla l(w; b)$
  fim
fim
retorna w para o servidor

representa o conjunto de clientes e T_s^{ag} representa o tempo de agregação do servidor.

$$T^r = \max_{\forall n \in \mathcal{N}}(t_n) + T_s^{ag} \tag{2.2}$$

Finalmente, o tempo necessário para realizar o treinamento é definido como a soma dos tempos de cada rodada. Esse cálculo pode ser visto na Equação 2.3, onde T denota o tempo de treinamento total e $\mathcal{R} = \{1, \ldots, R\}$ denota o conjunto de rodadas.

$$T = \sum_{r=1}^{R} T^r \tag{2.3}$$

As equações descritas foram utilizadas para obter os resultados de latência das propostas 3FL e DOFL, visto que o tempo real decorrido, ou tempo de relógio, pode não ser fidedigno durante as simulações, devido ao consumo de recursos computacionais por outros processos concorrentes no servidor de simulação. Os detalhes de obtenção dos resultados são discutidos no Capítulo 5.

2.2.1 Tipos de Aprendizado Federado

Aprendizado Federado Horizontal

O Aprendizado Federado Horizontal (*Horizontal Federated Learning* – HFL), também conhecido como aprendizado federado baseado em amostras, é caracterizado pelos participantes utilizarem conjuntos de dados que residem no mesmo espaço de variáveis, ou seja, as amostras de cada participante podem ser diferentes, porém suas amostras são constituídas pelas mesmas variáveis [19]. Esse paradigma ocorre

naturalmente durante a colaboração entre entidades com um objetivo em comum, como hospitais, onde aplicações tais quais detecção de tumores e predição de arritmias cardíacas são do interesse de todos os envolvidos [20, 21]. Uma definição formal do aprendizado federado horizontal pode ser vista na Equação 2.4, onde $\mathcal{X}$ se refere ao espaço de atributos, $\mathcal{Y}$ ao espaço de rótulos, $\mathcal{I}$ ao espaço de identificadores de amostras e $\mathcal{D}$ ao conjunto de dados.

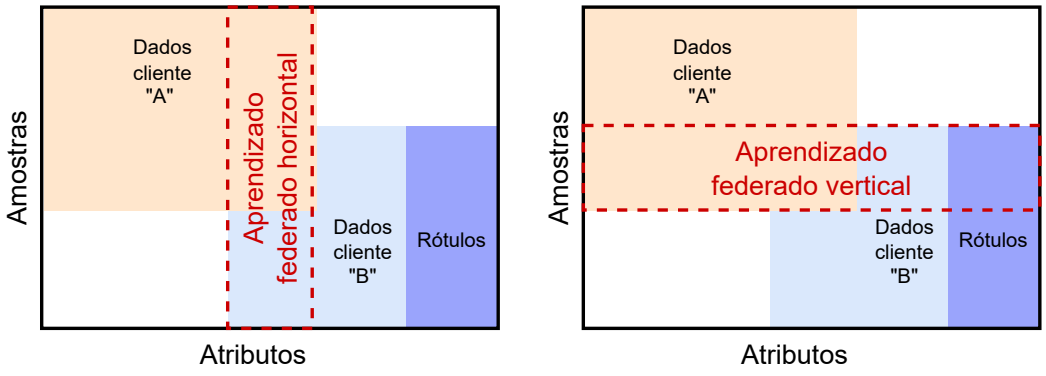
$$\mathcal{X}_i = \mathcal{X}_j, \mathcal{Y}_i = \mathcal{Y}_j, \mathcal{I}_i \neq \mathcal{I}_j, \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{2.4}$$

Aprendizado Federado Vertical

O Aprendizado Federado Vertical (*Vertical Federated Learning* – VFL), também conhecido como aprendizado baseado em atributos, é caracterizado pela presença de diferentes conjuntos de dados presentes no mesmo espaço de identificadores, porém com diferentes espaços de atributos [19]. Por exemplo, diferentes empresas podem manter registros acerca de um indivíduo, porém os dados armazenados podem variar conforme a área de atuação de cada empresa. Uma definição formal do aprendizado federado vertical pode ser vista na Equação 2.5.

$$\mathcal{X}_i \neq \mathcal{X}_j, \mathcal{Y}_i \neq \mathcal{Y}_j, \mathcal{I}_i = \mathcal{I}_j, \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{2.5}$$

A Figura 2.5 ilustra as principais diferenças entre o aprendizado federado horizontal e o aprendizado federado vertical.



(a) Aprendizado federado horizontal. (b) Aprendizado federado vertical.

Figura 2.5: Diferença entre o aprendizado federado horizontal e o aprendizado vertical. Adaptado de [3].

Aprendizado Federado *Cross-Device*

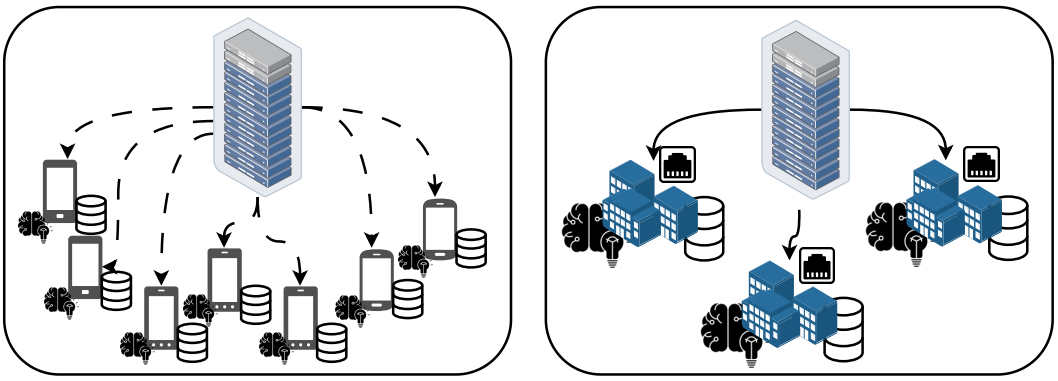
O aprendizado federado *cross-device* pode ser entendido como um cenário de aprendizado federado definido por algumas características, como um número elevado de

clientes possíveis, baixo poder de processamento, alta intermitência dos clientes ao longo do treinamento e condições de redes instáveis, como velocidades de transmissão inferiores e possíveis desconexões [14]. Naturalmente, este cenário se apresenta como o mais representativo das redes móveis e concentra grande parte do interesse de pesquisa [22–24].

Aprendizado Federado *Cross-Silo*

O aprendizado federado *cross-silo* é um tipo de aprendizado federado onde os clientes participantes são usualmente grandes organizações, como empresas ou hospitais [14]. Esse cenário possui como principais características um número reduzido de clientes participantes, a alta confiabilidade dos clientes envolvidos no treinamento e condições estáveis de rede. Uma aplicação comum deste tipo de cenário se dá quando diferentes organizações buscam treinar um modelo de aprendizado comum, porém restrições legais não permitem o compartilhamento de dados [25–27].

A Figura 2.6 ilustra as principais diferenças entre o cenário *cross-device* e o cenário *cross-silo*.



(a) Aprendizado federado *cross-device*. (b) Aprendizado federado *cross-silo*.

Figura 2.6: Diferença entre o aprendizado federado *cross-device* e o aprendizado federado cross-silo.

Aprendizado Federado na Borda da Rede

O Aprendizado Federado na Borda da Rede (*Federated Edge Learning* – FEEL) é caracterizado por seus clientes participantes serem dispositivos de borda, normalmente com capacidades computacionais heterogêneas, limitações de comunicação e conjuntos de dados distintos [28]. Aparelhos como *smartphones* ou equipamentos em redes IoT (*Internet of Things*) usualmente utilizam redes sem fio para se conectar à Internet e este tipo de conexão pode causar intermitências na comunicação que afetam o desempenho do aprendizado federado [2]. Ademais, os conjuntos de dados

nos dispositivos de borda costumam ser pequenos e terem variações entre si, fato que constitui um desafio adicional para a implementação de sistemas baseados no aprendizado federado [28]. Finalmente, visto o uso quase exclusivo de redes móveis na borda da rede, o consumo de energia em dispositivos móveis também se apresenta como uma questão a ser considerada durante o projeto de novas soluções.

É possível entender, então, que o aprendizado federado na borda da rede se apresenta como um superconjunto do aprendizado federado em redes móveis, onde as considerações feitas para contornar os efeitos negativos das redes móveis podem ser estendidas para cenários de FEEL. No entanto, o cenário FEEL introduz conceitos de redes hierárquicas como possíveis melhorias em projetos de aprendizado federado, tais qual o uso de servidores entre o núcleo e a borda da rede para realizar etapas intermediárias durante o treinamento [28–30].

2.2.2 Principais Desafios do Aprendizado Federado

Dentre os principais desafios em implementações de aprendizado federado é possível mencionar variações nas distribuições de dados dos clientes, a presença de dispositivos computacionalmente heterogêneos e falhas durante a comunicação devido a condições reais de redes de computadores [16].

Dados IID (*Independent and Identically Distributed*) são conjuntos de dados onde as amostras são independentes e coletadas a partir da mesma distribuição probabilística, ou seja, as amostras não possuem influência ou correlação sobre as outras [31]. Os conjuntos de dados que não seguem essas características são chamados de dados não-IID. Naturalmente, dados coletados por dispositivos inteligentes, como *smartphones*, seguem certos padrões, visto que cada dispositivo possui um perfil de uso distinto. A presença de dados não-IID afeta negativamente a construção de modelos de aprendizado de máquina [14] e seus efeitos no aprendizado federado não são diferentes. Em cenários de aprendizado federado, cada cliente responsável por treinar um modelo local deve ter seu conjunto de dados privado, visto que os conjuntos de dados locais são independentes entre si, os modelos locais que eventualmente serão agregados pelo servidor sofrem com essa característica e técnicas estão sendo atualmente investigadas para minimizar o impacto negativo de distribuições de dados não-IID [32, 33]. Adicionalmente, 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, este fenômeno afeta negativamente o desempenho e a convergência do modelo global [2, 34]. Ademais, a mobilidade de usuários em redes móveis também pode causar intermitências durante o treinamento, o que aumenta o tempo de treinamento total e também degrada o desempenho do modelo global [35, 36]. Finalmente, condições não ideais de

redes, como atrasos e desconexões, também impactam o tempo de convergência de modelos de aprendizado federado [2].

Dessa forma, técnicas para mitigação de retardatários (*straggler mitigation*) surgem a fim de minimizar o impacto de participantes mais lentos em tarefas de computação distribuída e, igualmente, aprendizado federado [37]. Visto que o aprendizado federado reside sob o paradigma de computação distribuída, o efeito negativo de clientes com poder computacional reduzido dá lugar a novas técnicas de treinamento [2, 38, 39].

A seguir, o Capítulo 3 discute como a literatura atual ataca os problemas explicados anteriormente.

Capítulo 3

Revisão Bibliográfica

Este capítulo revisa os artigos de maior relevância para este trabalho. Uma lista completa dos artigos considerados nesta dissertação pode ser encontrada no Apêndice A.

Revisão Sistemática

A seleção de artigos pertinentes para a revisão bibliográfica foi feita através da busca de termos relevantes em conferências renomadas na área de redes de computadores, tais quais GLOBECOM (*Global Communications Conference*, ICC (*Internal Conference on Communications*) e INFOCOM. Outras fontes como, IEEE Xplore, ACM Digital Library, SBC-OpenLib (SOL) e Google Scholar também foram utilizadas para busca com palavras-chave relevantes. A pesquisa foi limitada a artigos com até três anos, com exceção de artigos considerados "clássicos" na literatura, como aqueles creditados por cunhar conceitos importantes [9, 40], ou artigos de temas similares referenciados nos trabalhos revisados durante a etapa de elegibilidade. Uma lista completa dos artigos considerados neste trabalho, identificados através da metodologia descrita e revisados na etapa de triagem, pode ser encontrada no Apêndice A.

Uma visão detalhada da etapa de leitura feita neste trabalho pode ser vista na Figura 3.1. As palavras-chave "*straggler*", "*mitigation*", "*wireless*", "*constrained*" e "*limited*" foram utilizadas em combinação com termo-chave "*federated learning*" para identificar os artigos na etapa de identificação. O idioma das palavras-chave foi alterado segundo o idioma original dos artigos. Dentre os artigos restantes após a etapa de identificação, aqueles que abordam redes sem fio, mitigação de retardatários ou que possuem propostas singulares e inovadoras foram selecionados durante a etapa de triagem para leitura. Ademais, algumas tendências puderam ser identificadas durante a etapa de triagem, como a popularidade de propostas que atuam na camada física. Estes trabalhos, porém, não foram utilizados para modelar os cenários simulados nesta dissertação, visto que a proposta aqui descrita atua na ca-

mada de aplicação. As estratégias identificadas e os cenários avaliados foram usados como base e como inspiração para a idealização dos experimentos realizados neste trabalho.

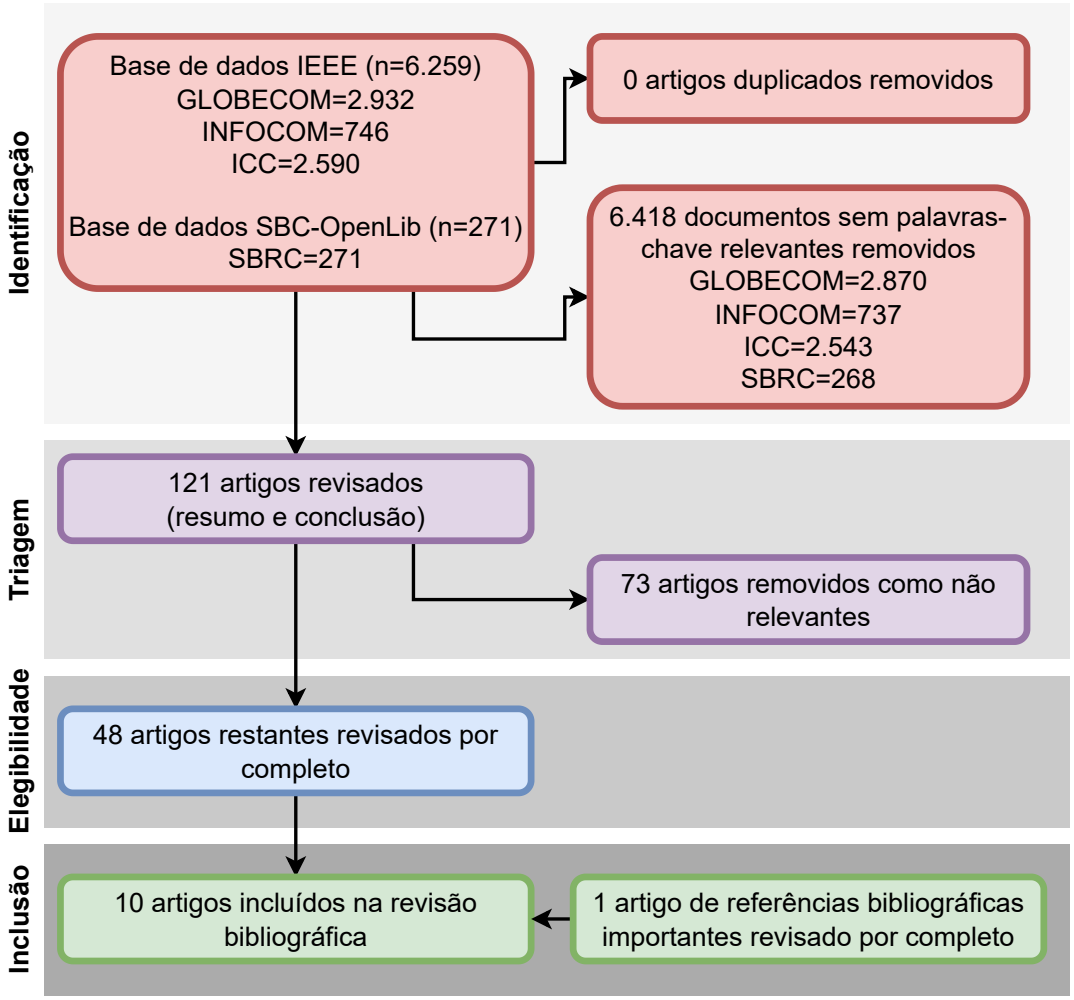


Figura 3.1: Método de seleção e revisão dos artigos.

Os trabalhos incluídos na revisão bibliográfica desta dissertação foram agrupados por tema e são descritos a seguir.

3.1 Análise dos Parâmetros de Redes Móveis

Feng et al. avaliam teoricamente como a mobilidade de usuários em uma rede sem fio hierárquica afeta o desempenho e a taxa de convergência de uma CNN no aprendizado federado [36]. A rede hierárquica avaliada consiste em um servidor de nuvem, 5 Pontos de Acesso (*Access Points* – APs) e 50 usuários móveis. Essa estrutura hierárquica permite que os APs façam agregações dos modelos de seus respectivos clientes ao invés de enviar os modelos locais diretamente para o servidor de nuvem. Os autores apontam o benefício das agregações parciais feitas pelos APs

e mostram que a troca de modelos locais entre clientes e APs possui um impacto positivo maior do que enviar os modelos locais diretamente para o servidor de nuvem. Ademais, as simulações também mostram que a alta mobilidade dos clientes afeta negativamente o desempenho do modelo e sua convergência. Esse resultado é, porém, acentuado quando as distribuições de dados são não-IID.

Bochie et al. apresentam uma análise acerca dos impactos de parâmetros de redes móveis no desempenho de uma rede neural treinada sob o paradigma federado [2]. Os autores mostram como desconexões de dispositivos durante o treinamento afetam negativamente o desempenho do modelo de aprendizado em uma tarefa de classificação. Ademais, o trabalho também evidencia como clientes incapazes de treinar seus modelos locais em tempo hábil, ou seja, dispositivos retardatários, aumentam o tempo total de treinamento, além de sugerir uma técnica de encerramento antecipado para suavizar este fenômeno. No entanto, as análises feitas utilizam até 20 clientes, o que não caracteriza as condições esperadas para cenários de treinamento *cross-device*. Finalmente, as simulações utilizam distribuições de dados IID, o que também não reflete de forma ideal as aplicações em redes móveis.

3.2 Mitigação de Retardatários

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 [39]. 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. A implementação dos retardatários é feita por meio de limitações de operações MAC (*Multiply-Accumulate*) nos 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), onde uma entidade de confiança é responsável por eleger quais clientes devem participar do treinamento em uma dada rodada [34]. 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.

Luo et al. atacam o problema do aumento no número de dispositivos participantes em cenários *cross-device* [41]. Nesse cenário, o servidor responsável por agregar os parâmetros de treinamento durante cada rodada do treinamento federado pode se tornar sobrecarregado com o crescimento expressivo no número de clientes. Os autores propõem o *framework* INP (*In-Network Processing*) , no qual os servidores de computação em névoa na borda da rede são utilizados para agregar parâmetros de treinamento e aliviar a carga do servidor central. Novos protocolos como MDP (*Model Download Protocol*) e MUP (*Model Upload Protocol*) são introduzidos para possibilitar a pré-agregação de modelos e a eliminação de *downloads* e *uploads* desnecessários ou duplicados. As simulações realizadas indicam que o uso de processamento dentro da rede pode reduzir o volume de tráfego total em até 60% por rodada.

3.3 Agendamento de Usuários

Liu et al. desenvolvem um *framework* de aprendizado híbrido que utiliza aprendizado federado e *split learning* [42]. Com algumas características em comum, o *split learning*, em contrapartida, busca manter a privacidade dos dados através da divisão dos modelos de aprendizado, onde o cliente detentor dos dados é responsável por treinar parte do modelo de aprendizado e o servidor central é responsável por treinar a outra parte [43]. Essa técnica reduz a carga computacional no cliente, o que leva a uma solução mais apropriada para ambientes de baixo poder computacional, como na borda da rede [44]. Apesar do *split learning* e do aprendizado federado serem usados para treinar modelos de aprendizado de forma colaborativa e distribuída, as duas técnicas são consideradas paralelas e apropriadas para diferentes cenários [44]. O esquema proposto pelos autores permite que os usuários utilizem um dos dois paradigmas de acordo com suas necessidades e capacidades computacionais. Adicionalmente, também é proposto um método de agendamento e alocação dos clientes, formulado como um *Multiple-Choice Knapsack Problem* (MCKP) que é então resolvida por meio de um algoritmo guloso. A avaliação do arcabouço é feita via simulações de redes UAV (*Unmanned Aerial Vehicle*) sem fio e demonstra possível economia de energia dos dispositivos enquanto atinge desempenho de classificação similar à abordagem federada tradicional.

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 [45]. 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 [9], FedGATE [46], FedNova [47]. 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, o que não condiz com os conjuntos de dados não-IID comumente presentes em cenários *cross-device*.

3.4 Agrupamento

Albaseer et al. estudam possíveis melhorias no aprendizado federado através da implementação de um algoritmo de seleção de clientes [48]. O algoritmo proposto seleciona clientes para participar no treinamento conforme a latência de treinamento em determinadas rodadas e utiliza reaproveitamento de banda para os clientes que consomem mais tempo durante a atualização do modelo federado. Os autores também destacam a importância de separar clientes com distribuições de dados dissimilares no início do treinamento, técnica esta que acelera a taxa de convergência do modelo e reduz os custos de comunicação em um ambiente com banda limitada. Adicionalmente, a proposta avaliada utiliza informações de *hardware* e *software* dos clientes, como velocidade do processador e tamanho do conjunto de dados, respectivamente. Finalmente, o servidor pode estimar o tempo necessário para cada participante realizar uma rodada de treinamento. Após definido um tempo aceitável de treinamento, os clientes elegíveis participam do treinamento em uma dada rodada.

Chu et al. combinam o agrupamento de clientes com transferência de aprendizado em redes neurais para reduzir em até 98,5% os custos de comunicação do aprendizado federado, enquanto mantendo desempenho de classificação equiparável ao método tradicional [49]. Transferência de aprendizado é uma técnica que consiste em utilizar um modelo de aprendizado desenvolvido para uma tarefa como o ponto inicial para desenvolver um modelo de aprendizado que será treinado para realizar outra tarefa [50]. Esse método de treinamento é especialmente útil em aplicações em que o volume de dados disponível para o treinamento é reduzido e a nova tarefa é similar à tarefa original para a qual o modelo foi inicialmente treinado [51]. O *framework* proposto, chamado de CEFL (*Communication-Efficient Federated Learning*), inicialmente, treina localmente um modelo de rede neural em todos os clientes

para obter um grafo de similaridade entre os clientes a partir dos pesos de suas redes neurais resultantes. Após construído o grafo de similaridade, os clientes são agrupados em *clusters* e o cliente mais próximo do centroide de cada *cluster* é eleito como um líder que é responsável por participar do aprendizado federado e repassar o modelo para os outros membros do seu *cluster*. Durante o aprendizado federado, o servidor envia apenas as camadas iniciais da rede neural agregada para os líderes de cada *cluster*, a fim de reduzir ainda mais os custos de comunicação. Finalmente, após um número definido de rodadas de treinamento, os líderes de cada *cluster* enviam seus modelos de aprendizado aos outros membros pertencentes ao seu *cluster*, que devem utilizar seus conjuntos de dados locais para continuar o treinamento, utilizando da transferência de aprendizado para personalizar suas redes neurais.

Kim et al. desenvolvem um algoritmo de agrupamento dinâmico e avaliam seu desempenho no contexto de predição de *handover* em redes 5G [52]. Os autores se baseiam em algoritmos desenvolvidos anteriormente para desenvolver seu algoritmo de três etapas: agrupamento baseado em GANs (*Generative Adversarial Networks*) [53] para agrupar os clientes sem compartilhar dados; calibração de *clusters* [54] para realocar participantes dinamicamente; e divisão de *clusters* [55] para alterar dinamicamente a quantidade de *clusters*. A proposta é avaliada em diferentes cenários de predição de séries temporais e, quando comparada a uma proposta similar definida como referência, baseada na Análise de Componentes Principais (*Principal Component Analysis* – PCA) [56], é possível observar um ganho no desempenho de predição de até 43%.

3.5 Demais Abordagens

Outras abordagens também são exploradas atualmente, como o uso de informações acerca do canal de comunicação para agendamento de usuários [57, 58], a modelagem dos cenários de aprendizado como um problema de otimização [59, 60] e a reutilização de conceitos de teoria de jogos [61]. Essas técnicas foram consideradas tangentes a este trabalho.

Este trabalho implementa e compara três técnicas para a mitigação do impacto de dispositivos retardatários e de volumes de dados desbalanceados durante o aprendizado federado. A primeira técnica consiste em selecionar diferentes clientes no início do treinamento conforme o tempo necessário para terminarem suas rodadas de treinamento local. A segunda técnica consiste em iniciar o treinamento do modelo de aprendizado utilizando os dispositivos com maiores conjuntos de dados locais e incluir o restante dos dispositivos após uma etapa de treinamento inicial. Finalmente, a terceira técnica busca mesclar as duas técnicas anteriores em uma abordagem híbrida, que considera tanto as latências de treinamento locais quanto os volumes

de dados locais dos clientes. Diferentemente dos trabalhos anteriores, as propostas desta dissertação utilizam distribuições de dados realistas do cenário de aprendizado federado e podem ser implementadas ajustando técnicas de aprendizado federado já existentes, como o FedAVG.

Capítulo 4

Propostas e Contribuições

Como explicado no Capítulo 2, a heterogeneidade de dispositivos em cenários *cross-device* é um obstáculo para o treinamento de modelos de aprendizado federado de forma rápida e efetiva. Especialmente, técnicas como seleção de clientes e redução do impacto de dispositivos retardatários, se apresentam como possíveis soluções para melhorar o desempenho dos modelos de aprendizado federado e reduzir a latência de treinamento. Diante disso, este capítulo apresenta três propostas para reduzir a latência no aprendizado federado e melhorar o desempenho dos modelos finais. As três técnicas se baseiam na seleção de clientes de acordo com um critério em particular, seja a velocidade de treinamento ou o volume de dados, ou uma combinação dos dois.

4.1 Aprendizado Federado "*Fastest-First*"

No Aprendizado Federado *Fastest-First* (*Fastest-First Federated Learning* – 3FL), apenas os clientes com maior capacidade computacional participam das primeiras rodadas do aprendizado federado, com o intuito de reduzir drasticamente a latência nas rodadas iniciais. Após o treinamento inicial, o resto dos 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 *warm start*. A Figura 4.1 ilustra a proposta 3FL.

Duas alternativas para definir quais usuários podem ser considerados rápidos ou lentos são: (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. Devido ao desafio adicional de estimar a velocidade do treinamento de acordo com informações de *hardware* dos clientes [62] e diferentemente de trabalhos anteriores [45], esta dissertação implementa uma etapa de calibração antes do início do treinamento para a seleção de clientes. Essa decisão foi tomada a fim de buscar uma opção que não necessite de alterações nas implementações de estratégia

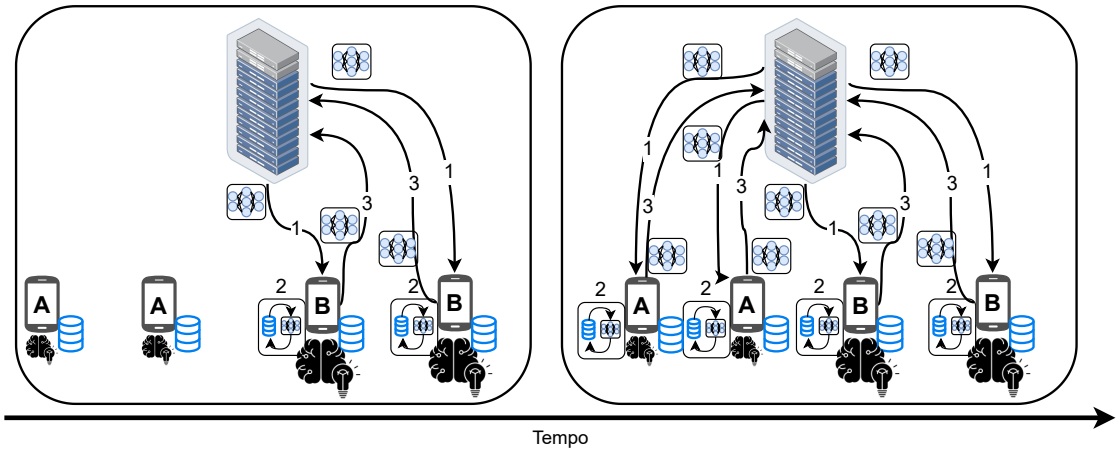


Figura 4.1: Proposta 3FL onde os dois clientes "B" possuem maior poder computacional que os dois clientes "A".

diretamente no *framework* de aprendizado federado, já que o uso de metadados é utilizado em algumas configurações do aprendizado federado, porém não faz parte do método FedAVG tradicional [63, 64].

A etapa de calibração consiste em difundir o modelo de treinamento para os usuários e definir um tempo limite para o treinamento de uma rodada. Caso nenhum usuário seja capaz de realizar o treinamento dentro do tempo limite, o tempo limite da rodada é duplicado e a etapa é repetida até que um número pré-definido de usuários responda ao treinamento. Esses usuários serão considerados como mais rápidos para o início do treinamento. Naturalmente, o tempo de resposta não depende exclusivamente da capacidade de treinamento do usuário, porém, para efeitos práticos, a latência de comunicação é uma parcela menor do tempo total de treinamento [14]. 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 pode ser visto na Figura 4.2.

Nota-se que a etapa de calibração depende de um limiar, onde determinado percentual dos clientes serão selecionados para o treinamento. A porção de clientes incluídos no início do treinamento impacta diretamente a latência, logo, deve haver um compromisso entre a redução de latência buscada e o número de clientes participantes da primeira etapa.

Finalmente, como o número de rodadas de treinamento é normalmente da ordem de centenas, o tempo necessário para realizar a etapa de calibração impacta minimamente o tempo total de treinamento. Mesmo assim, esta dissertação demonstra que a proposta 3FL ainda é capaz de reduzir o tempo total de treinamento.

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

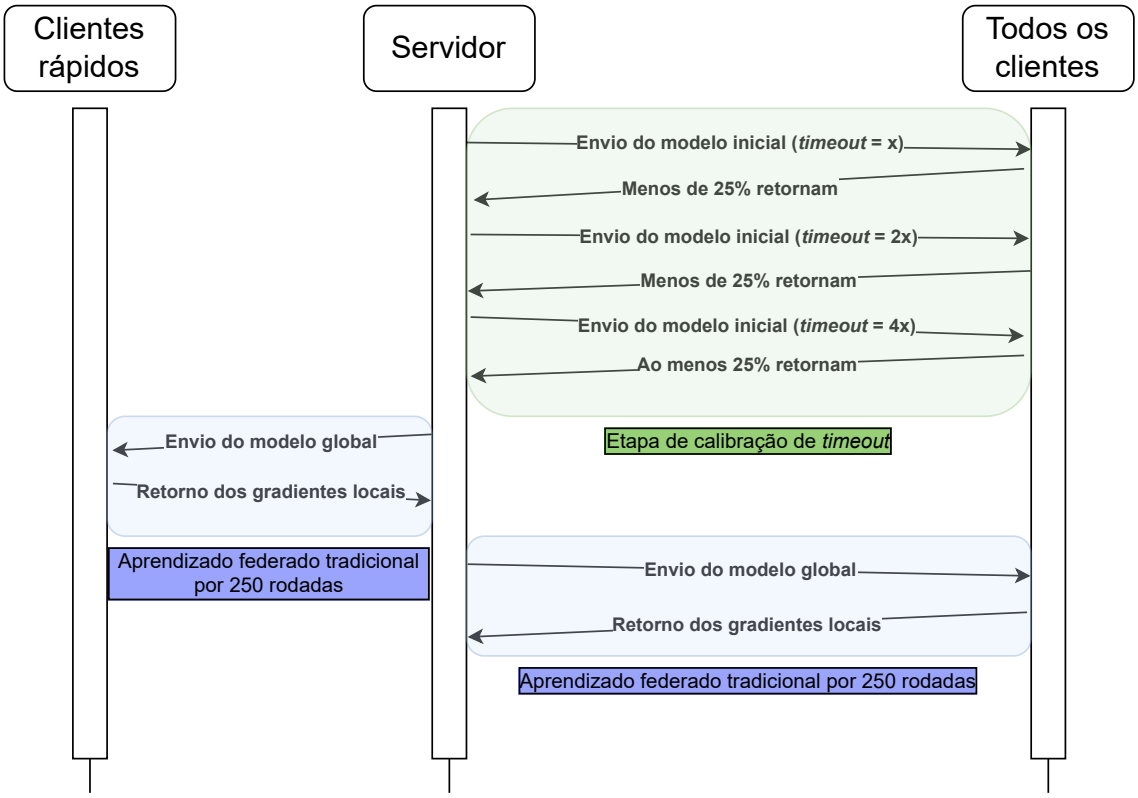


Figura 4.2: Diagrama de execução da proposta 3FL. Cenário com *timeout* inicial x selecionando ao menos 25% dos clientes totais para 250 rodadas de treinamento iniciais.

4.2 Aprendizado Federado Orientado a Dados

No Aprendizado Federado Orientado a Dados (*Data-Oriented Federated Learning* – DOFL), os clientes com maior volume de dados participam das primeiras rodadas do aprendizado federado. A expectativa desse cenário é que os clientes com acesso a uma maior massa de dados impactem de forma mais significativa o desempenho do modelo, permitindo assim reduzir a comunicação na rede para clientes com pouca influência no desempenho. Essa técnica, apesar de não atacar diretamente o problema dos retardatários no aprendizado federado, também visa melhorar o desempenho do modelo e reduzir o consumo dos recursos de rede durante o treinamento. A Figura 6.7 ilustra a proposta DOFL.

Um ponto notável desta abordagem é a necessidade de informações acerca do conjunto de dados de cada cliente, afinal, o volume de dados local é um metadado importante para a configuração apropriada do DOFL. Entretanto, o compartilhamento desse tipo de dado não é incomum em aplicações reais e seu uso não impacta a privacidade dos dados de treinamento [63, 64]. Apesar de metadados poderem ser utilizados para inferir mais informações acerca dos clientes [19], técnicas para estimar o volume de dados sem a transmissão de metadados para o servidor pode

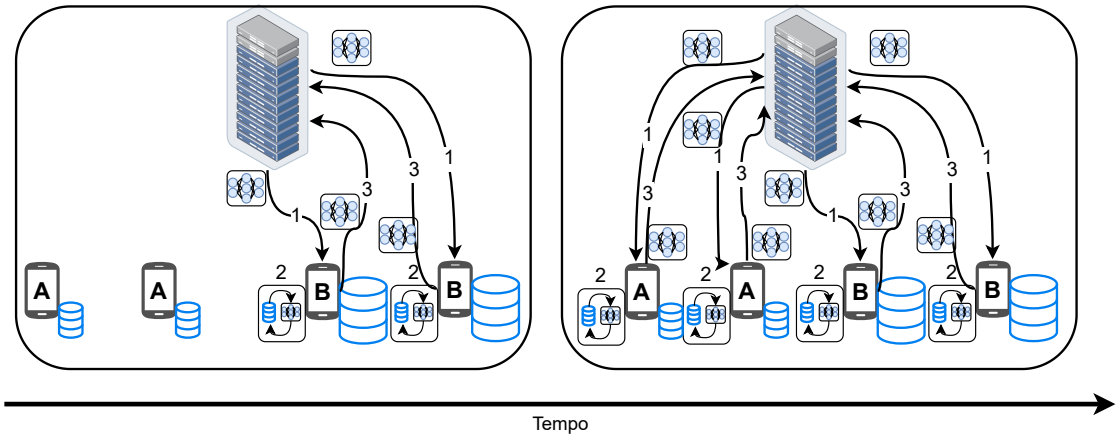


Figura 4.3: Proposta DOFL onde os dois clientes “B” possuem mais amostras em seu conjunto local que os dois clientes “A”.

levar a novas brechas para clientes maliciosos [65, 66], a abordagem DOFL avalia o uso de metadados referentes ao volume de dados dos clientes enviados ao servidor.

Diferentemente da proposta 3FL, esta proposta requer uma etapa inicial para o recebimento de metadados, mas não precisa de uma etapa de calibramento para definir quais clientes devem participar do treinamento inicial. Um diagrama de execução incluindo a etapa de transferência de metadados pode ser visto na Figura 4.4.

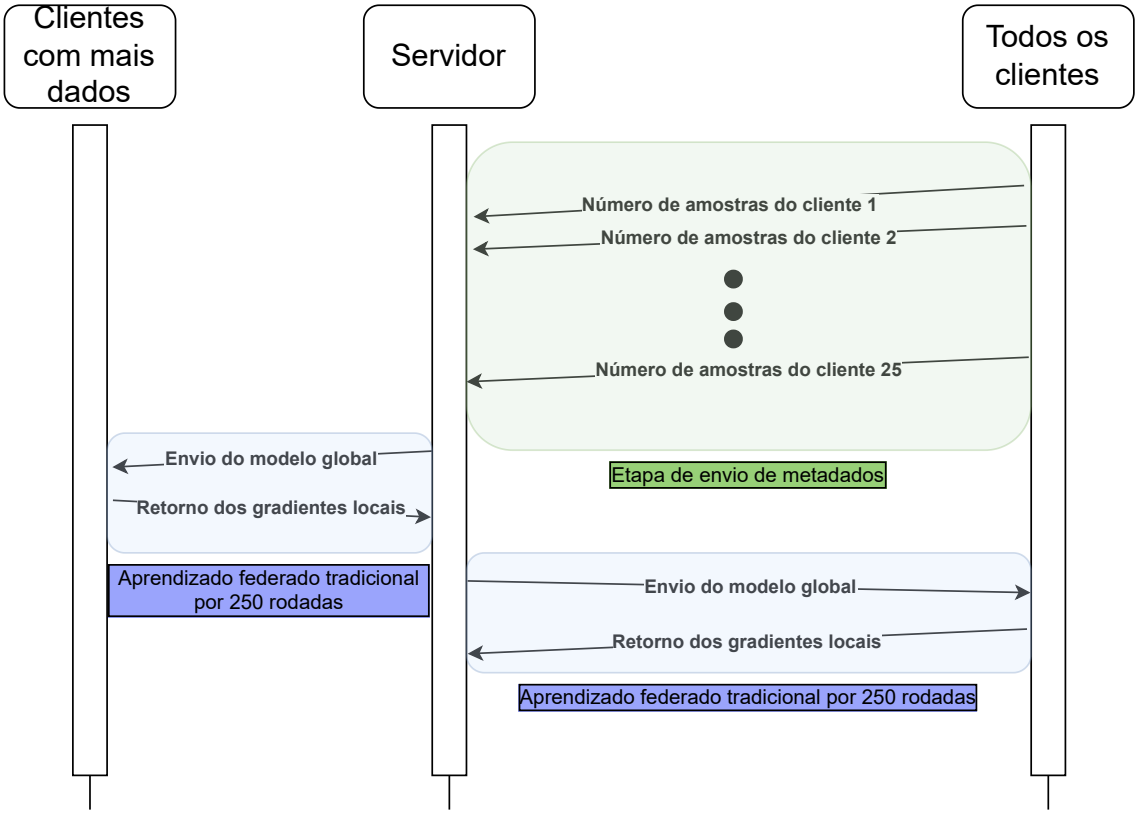


Figura 4.4: Diagrama de execução da proposta DOFL. Cenário em que ao menos 25% dos clientes totais são selecionados para 250 rodadas de treinamento iniciais.

4.3 Aprendizado Federado Híbrido

A terceira proposta, Aprendizado Federado Híbrido (*Hybrid-FL*), se difere por usar tanto o volume de dados locais quanto a latência de treinamento dos clientes como fatores de tomada de decisão para selecionar os clientes participantes da etapa inicial.

Na proposta híbrida, o tamanho do conjunto de dados local e a latência de treinamento, representada pelo tempo de treinamento para tornar a visualização mais intuitiva, são mapeadas e normalizadas em um espaço vetorial ilustrado pela Figura 4.5. Nela, observa-se que os clientes mais afastados do eixo X atingem maior velocidade de treinamento e, por consequência, menor latência. Já os clientes mais afastados do eixo Y possuem maiores conjuntos de dados. A abordagem avaliada tenta utilizar os clientes com as duas características, menor latência e maior volume de dados, que ficam posicionados mais distantemente da interseção dos eixos X e Y. Os clientes de interesse podem ser visualizados na Figura 4.5.

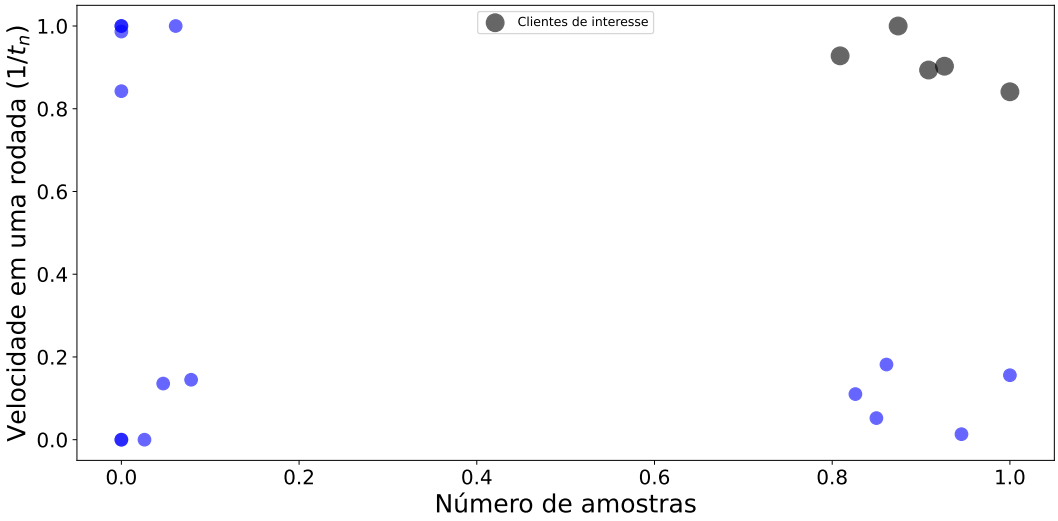


Figura 4.5: Exemplo de seleção com 20 clientes totais e 5 clientes selecionados. A velocidade de execução em uma rodada e o número de amostras de cada cliente foram normalizados.

Como apresentado na Figura 4.5, há um "raio de seleção" que deve ser apropriadamente ajustado para obter um compromisso entre desempenho de classificação, uso de recursos computacionais e latência total de treinamento. Afinal, a inclusão de mais clientes no treinamento pode apenas aumentar o tempo total de treinamento, sem que o impacto no desempenho seja diretamente previsível.

Capítulo 5

Ferramentas e Configuração dos Experimentos

Este capítulo apresenta as ferramentas utilizadas e as configurações usadas durante os experimentos.

5.1 Ferramentas Utilizadas

Software

O conjunto de ferramentas tipicamente empregado na exploração de dados em Python, como Numpy e scikit-learn, foi utilizado neste trabalho. Ademais, os seguintes *frameworks* foram usados:

- TensorFlow [67]: biblioteca de código aberto amplamente utilizada para aprendizado de máquina e inteligência artificial. Ele oferece uma abordagem flexível para a criação e treinamento de modelos de aprendizado de máquina, incluindo redes neurais profundas.
- Keras [68]: poderosa e amigável biblioteca de aprendizado de máquina e redes neurais de código aberto. O Keras fornece uma interface de alto nível para a criação e treinamento de modelos de aprendizado profundo. Embora o Keras possa ser executado sobre várias bibliotecas de *backend* de aprendizado de máquina, como TensorFlow, Theano e Microsoft Cognitive Toolkit (CNTK), ele se integra especialmente bem ao TensorFlow.
- Flower [69]: *Federated Learning Orchestrator* (Flower) é um *framework* de código aberto projetado para facilitar a implementação e o gerenciamento de treinamento federado.

Todos os resultados foram obtidos utilizando ferramentas e *frameworks* de código aberto. A fim de permitir a reprodutibilidade dos experimentos, os códigos usados estão disponíveis em um repositório GitHub[1].

Hardware

Diversos computadores 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 usados durante as simulações de latência. As configurações de *hardware* e de *software* dos computadores usados são:

1. Um servidor AMD Epyc 7452 2.35GHz com 32 núcleos de processamento executando Ubuntu 20.04.5 e 378GB RAM.

2. Um computador Intel Core i5-10400 2,90GHz com 6 núcleos de processamento executando Windows 10, 32GB RAM, hospedando um sistema operacional Ubuntu 22.04 através do *software* VMWare Workstation 17.

3. Um servidor Intel Core i5-9600k 3,70GHz com 6 núcleos de processamento executando Debian 9 e 32GB RAM.

Para o computador Intel Core i5-10400F executando Windows 10 (2), apenas 4 núcleos de processamento e 28GB de memória RAM foram compartilhados com a máquina virtual. Para fins de reprodutibilidade, todos os experimentos foram configurados para serem executados em um sistema igual ou superior ao descrito, com exceção do cenário de avaliação com 50 clientes. Entretanto, o limitante computacional é devido à memória ocupada pelos códigos clientes executando de forma concorrente, o que pode ser contornado ao utilizar múltiplos sistemas para executar grupos de clientes.

O tempo de execução médio de cada cliente no sistema 2 foi considerado ao simular os clientes mais lentos durante a análise de latência, atingindo um tempo médio de treinamento de 63 segundos. Já o tempo de execução médio de cada cliente no sistema 1 foi considerado ao simular os clientes mais rápidos durante a análise de latência, atingindo um tempo médio de treinamento de 20 segundos.

5.2 Conjuntos de Dados

Este trabalho utiliza o conjunto de dados MNIST (*Modified National Institute of Standards and Technology*) nas simulações [70]. O conjunto de dados MNIST é composto de 60.000 amostras de treinamento e 10.000 amostras de teste. Cada

[1] `https://github.com/kaylani2/mestrado`.

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. Devido a uma limitação nas camadas de entrada da rede neural utilizada, as amostras foram preenchidas com *pixels* brancos para assumir o tamanho de 32 *pixels* de largura e 32 *pixels* de altura. Visto que o conjunto de dados consiste de imagens para classificação, a acurácia dos modelos foi utilizada como principal métrica de desempenho. Algumas amostras do conjunto de dados MNIST podem ser vistas na Figura 5.1.

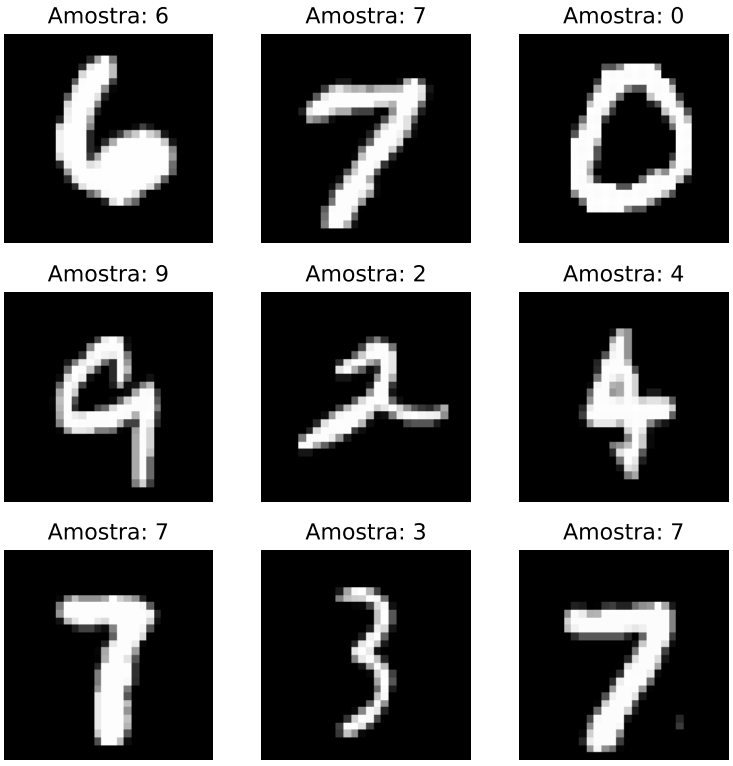


Figura 5.1: Amostras do conjunto de dados MNIST.

A divisão de amostras entre os clientes é feita de forma não-IID, seguindo técnicas utilizadas na literatura [9, 36, 39], onde cada cliente utiliza apenas amostras de até 2 rótulos diferentes. Essa divisão faz com que as distribuições de dados dos clientes não sejam internamente IID. Repare que as amostras do conjunto de dados MNIST não são balanceadas em relação aos rótulos, como visto na Tabela 5.1. De acordo, durante os experimentos, para garantir que não haja sobreposição de amostras entre os clientes, existem cenários em alguns clientes possuem uma amostra adicional em relação aos outros clientes que utilizam amostras das mesmas classes.

5.3 Modelo de Aprendizado Profundo

Inicialmente, a arquitetura MobileNetV2 foi utilizada durante as simulações [71]. Essa rede neural foi escolhida devido ao seu tamanho reduzido, com menos de 4

Tabela 5.1: Distribuição de amostras do conjunto de dados MNIST.

Rótulo	Amostras no conjunto de treino	Amostras no conjunto de teste
0	5923	980
1	6742	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009

milhões de parâmetros, em relação a outras arquiteturas, como VGG16 [72] e ResNet50 [73], que possuem aproximadamente 140 milhões de parâmetros. Apenas as camadas de entrada e de saída da rede neural foram modificadas para se adaptar ao conjunto de dados MNIST. A visualização das camadas da rede neural MobileNetV2 foi omitida devido ao número elevado de camadas de processamento, porém uma descrição detalhada de sua arquitetura pode ser encontrada em Sandler et al. [71].

Todavia, como discutido em detalhes na Seção 6.2, a arquitetura MobileNetV2 se mostrou inapropriada para a avaliação dos cenários discutidos. Como consequência, um modelo de aprendizado menor baseado em CNNs foi utilizado para o resto dos experimentos. A fim de permitir a reprodutibilidade dos experimentos realizados nesta dissertação, o modelo personalizado pode ser visto no Código 5.1.

Código 5.1: Arquitetura personalizada de rede neural convolucional.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        320
 batch_normalization (Batch  (None, 30, 30, 32)        128
 Normalization)
 conv2d_1 (Conv2D)           (None, 28, 28, 32)        9248
 batch_normalization_1 (Bat  (None, 28, 28, 32)        128
 chNormalization)
 conv2d_2 (Conv2D)           (None, 14, 14, 32)        25632
 batch_normalization_2 (Bat  (None, 14, 14, 32)        128
 chNormalization)
```

```
 dropout (Dropout)           (None, 14, 14, 32)        0
 conv2d_3 (Conv2D)           (None, 12, 12, 64)        18496
 batch_normalization_3 (Bat  (None, 12, 12, 64)        256
 chNormalization)
 conv2d_4 (Conv2D)           (None, 10, 10, 64)        36928
 batch_normalization_4 (Bat  (None, 10, 10, 64)        256
 chNormalization)
 conv2d_5 (Conv2D)           (None, 5, 5, 64)          102464
 batch_normalization_5 (Bat  (None, 5, 5, 64)          256
 chNormalization)
 dropout_1 (Dropout)         (None, 5, 5, 64)          0
 flatten (Flatten)           (None, 1600)              0
 dense (Dense)               (None, 128)               204928
 batch_normalization_6 (Bat  (None, 128)               512
 chNormalization)
 dropout_2 (Dropout)         (None, 128)               0
 dense_1 (Dense)             (None, 10)                1290
=================================================================
Total params: 400970 (1.53 MB)
Trainable params: 400138 (1.53 MB)
Non-trainable params: 832 (3.25 KB)
_________________________________________________________________
```

5.4 Cenários Avaliados e Parâmetros de Simulação

A Tabela 5.2 apresenta os valores dos parâmetros utilizados nas simulações e seus respectivos símbolos. Notavelmente, o número de épocas de treinamento local (E) foi mantido baixo a fim de permitir uma melhor visualização do treinamento, visto que as métricas escolhidas são avaliadas entre cada rodada de treinamento, ou seja, um número menor de épocas locais permite que um número maior de rodadas seja executado dentro do mesmo período. Finalmente, o otimizador Adam [40] foi utilizado devido à sua eficiência computacional e ao seu baixo consumo de memória.

Tabela 5.2: Parâmetros das simulações de aprendizado federado.

Símbolo	**Significado**	**Valores utilizados**
N	Número de clientes	2, 5, 10, 15, 25, 50
E	Épocas de treinamento local	1
B_c	Tamanho dos *batches* locais	64
R	Número de rodadas de treinamento federado	100, 500
η	Taxa de aprendizado	10^{-2}, 10^{-3}
-	Otimizador	Adam [40]

Capítulo 6

Análise dos Resultados

Primeiramente, se faz importante ressaltar que é possível obter excelentes desempenhos de classificação no conjunto de dados MNIST com a arquitetura MobileNetV2, especialmente ao utilizar técnicas como transferência de aprendizado [74]. No entanto, esta dissertação utiliza os resultados obtidos a partir das configurações descritas na Tabela 5.2 para o algoritmo FedAVG, como comparação para as próximas técnicas. O ajuste de hiperparâmetros, como taxa de aprendizado e o número de épocas de treinamento, para melhorar o desempenho de modelos de aprendizado de máquina é extensamente explorado na literatura [1] e foi considerado tangente a este trabalho.

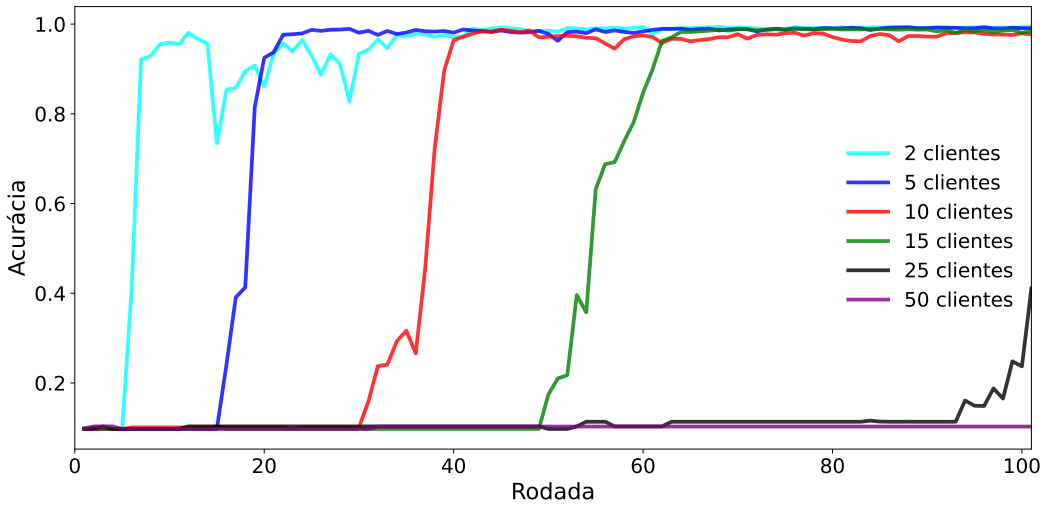
As únicas modificações feitas a hiperparâmetros foram realizadas a fim de tornar a compreensão das propostas mais claras e não para otimizar o desempenho dos modelos de aprendizado federado utilizados nesta dissertação.

Destaca-se que em todos os cenários de aprendizado federado simulados, os clientes não possuem sobreposição de amostras entre seus conjuntos de treinamento e teste locais.

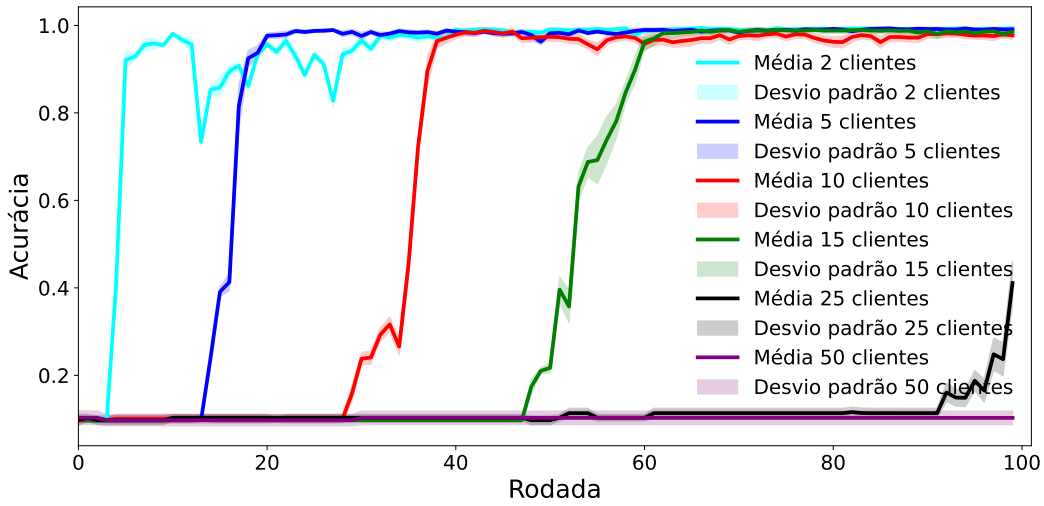
6.1 FedAVG em Dados IID

A Figura 6.1 apresenta os desempenhos do algoritmo FedAVG para diferentes configurações de clientes em um cenário de distribuição de dados IID, isto é, cada cliente obtém seu conjunto de dados local a partir da mesma distribuição de dados. Essa etapa foi executada para estabelecer um patamar de desempenho ao qual os outros algoritmos serão comparados.

A Figura 6.1 demonstra um fenômeno natural presente em cenários de aprendizado federado, onde um número elevado de clientes participantes torna o treinamento, para uma mesma configuração de hiperparâmetros, mais demorado. Como pode ser observado, aumentar o número de rodadas de treinamento é uma maneira eficaz de contornar este desafio. Outra abordagem se dá pelo aumento no número



(a) Cenário IID 2, 5, 10, 15, 25 e 50 clientes. Acurácias medidas no servidor.



(b) Cenário IID 2, 5, 10, 15, 25 e 50 clientes. Acurácias medidas nos clientes.

Figura 6.1: Desempenho de classificação do algoritmo FedAVG utilizando a arquitetura MobileNetV2 com os seguintes hiperparâmetros: (i) taxa de aprendizado igual a 10^{-2}, (ii) tamanho dos *batches* locais igual a 64 amostras, (iii) épocas de treinamento local igual a 1.

de épocas de treinamento local em cada cliente. No entanto, devido à natureza exploratória deste trabalho, apenas o número de rodadas foi estendido para garantir máxima granularidade aos resultados, visto que a avaliação de desempenho é feita entre rodadas de treinamento, não entre cada época de treinamento local.

Neste ponto, também é interessante destacar duas possíveis abordagens para avaliar o desempenho de um modelo de aprendizado federado. Em aplicações reais, apenas os clientes possuem amostras locais para avaliar o desempenho de seus modelos, logo, o servidor deve coletar as métricas de classificação de cada cliente e agregá-las de alguma forma, usualmente com uma média ponderada. No entanto, em

simulações, o conjunto de dados de teste pode ser armazenado no servidor, que age como um "oráculo" ao avaliar o desempenho do modelo agregado em seu conjunto de teste. Ao comparar as Figuras 6.1(a) e 6.1(b), pode-se notar que o desempenho medido no servidor equivale, de fato, à média dos desempenhos dos clientes. Essa visualização é normalmente utilizada por sua simplicidade, no entanto, a aferição de desempenho individual em cada cliente exibe informações adicionais.

As áreas sombreadas na Figura 6.1(b) representam o desvio padrão da acurácia nos clientes ao longo das rodadas de treinamento. É possível observar que o desvio padrão é maior em torno das regiões do gráfico onde o desempenho apresenta uma inflexão, como no cenário com 15 clientes em torno de 50 rodadas. Esse resultado pode ser interpretado como a dificuldade adicional de um processo de otimização onde mais participantes participam da tomada de decisão. Finalmente, um maior desvio representa um cenário onde nem todos os clientes atingiram um desempenho satisfatório durante o treinamento, o que pode indicar a necessidade de reconfiguração do cenário de aprendizado. Apesar desse cenário não ter ocorrido durante as simulações no cenário de distribuição de dados IID, uma ilustração exacerbada desse fenômeno criado de forma artificial pode ser visto no Anexo B.

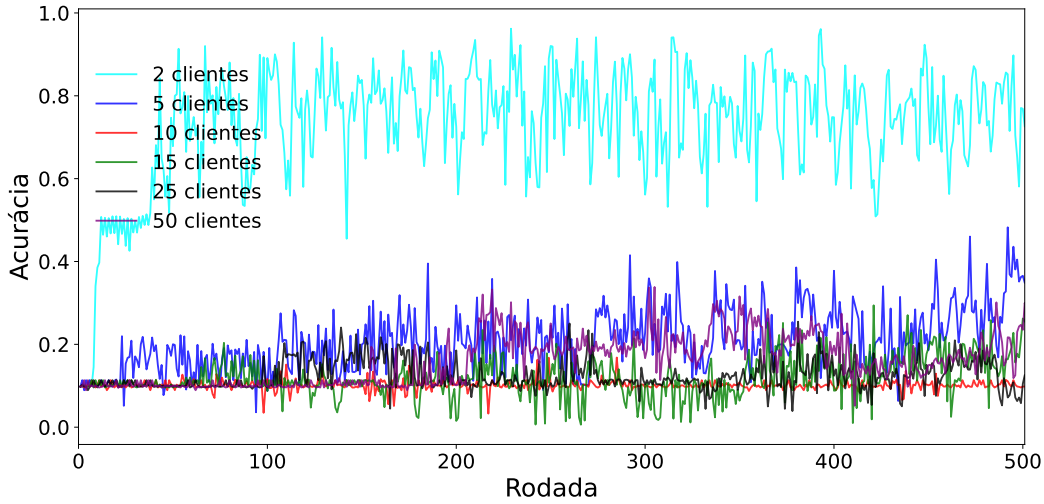
6.2 FedAVG em Dados Não-IID

A Figura 6.2 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 [75, 76]. Adicionalmente, para facilitar a visualização dos resultados, apenas os cenários com 2 e 10 clientes são apresentados na Figura 6.2(b).

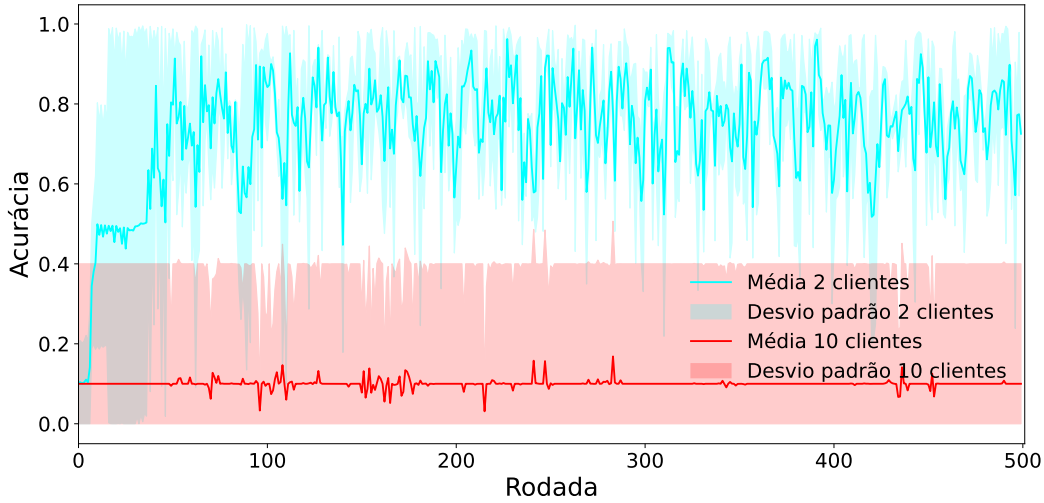
O valor elevado do desvio padrão no cenário de 10 clientes, exibido na Figura 6.2(b), demonstra um caso de sobreajuste *overfitting* [1], onde alguns clientes atingem altos desempenhos, enquanto outros obtém desempenhos inferiores. Como mencionado anteriormente, esse fenômeno se dá pela presença de apenas uma classe nos conjuntos de dados locais. A fim de manter a análise de resultados consistente e intuitiva, as acurácias medidas no servidor serão utilizadas ao invés das acurácias medidas nos clientes.

Naturalmente, devido à presença de 10 classes no conjunto de dados MNIST,

divisões sem interseções de classes não são possíveis para cenários com mais 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. Esta distribuição representa mais fidedignamente as aplicações reais e, de acordo, esses cenários constituem o foco do resto desta dissertação.



(a) Cenário não-IID 2, 5, 10, 15, 25 e 50 clientes. Acurácias medidas no servidor.



(b) Cenário não-IID 2 e 10 clientes. Acurácias medidas nos clientes.

Figura 6.2: Desempenho de classificação do algoritmo FedAVG utilizando a arquitetura MobileNetV2 com os seguintes hiperparâmetros: (i) taxa de aprendizado igual a 10^{-2}, (ii) tamanho dos *batches* locais igual a 64 amostras, (iii) épocas de treinamento local igual a 1.

A grande variação no desempenho do modelo entre rodadas consecutivas de treinamento e o desempenho reduzido para cenários com maior número de clientes,

observados nas Figuras 6.2(a) e 6.2(b), sugerem fortemente um fenômeno de sobreajuste. Esse fenômeno é justificado pela diminuição do conjunto de dados local de cada cliente, conforme a divisão de dados descrita anteriormente. Nesse ponto, técnicas como ajuste de hiperparâmetros e, especialmente, aplicação de regularização, podem ser usadas para melhorar o desempenho do modelo. No entanto, seguindo tendências da literatura [9, 45], um modelo de aprendizado personalizado com menos parâmetros foi utilizado no restante dos experimentos. Os resultados do treinamento do modelo personalizado podem ser vistos na Figura 6.3.

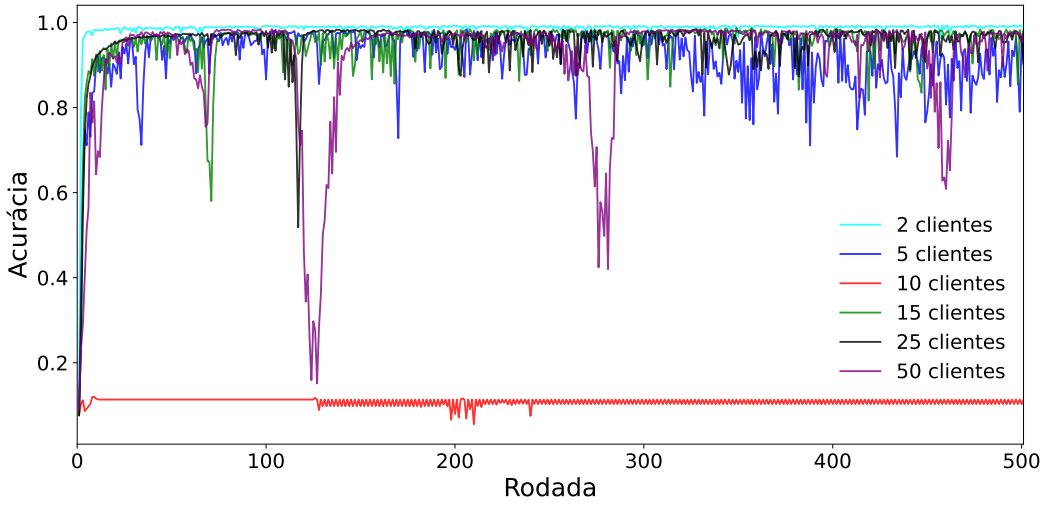


Figura 6.3: Desempenho de classificação do algoritmo FedAVG utilizando uma CNN personalizada com os seguintes hiperparâmetros: (i) taxa de aprendizado igual a 10^{-3}, (ii) tamanho dos *batches* locais igual a 64 amostras, (iii) épocas de treinamento local igual a 1. Cenário não-IID 2, 5, 10, 15, 25 e 50 clientes. Acurácias medidas no servidor.

O melhor desempenho de classificação obtido pela CNN personalizada, evidenciado pela Figura 6.3, é resultado da menor especificidade 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, na Figura 6.3. No entanto, mecanismos tipicamente utilizados, tais como parada antecipada (*early stopping*), podem ser utilizados para evitar essa degradação no desempenho. A aplicação desses mecanismos, porém, foi considerada tangente ao escopo desta dissertação.

Especificamente, a configuração com 25 clientes em cenário não-IID com modelo de aprendizado personalizado foi utilizada para avaliar os algoritmos propostos nesta dissertação. 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 6.3, logo, o cenário com 25 clientes se mostrou como o mais apropriado para avaliar as propostas desta dissertação.

6.3 Aprendizado Federado *Fastest-First*

A Figura 6.4 compara o desempenho da proposta 3FL em diferentes cenários com o algoritmo FedAVG. As 250 primeiras rodadas de aprendizado foram executadas apenas com os clientes mais rápidos, enquanto o resto do treinamento foi realizado com todos os clientes. As simulações foram feitas com 25 clientes participantes. Nela, é possível observar como o desempenho base é atingido para diferentes quantidades de clientes mais rápidos na rede. Destaca-se, porém, 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 os cenários de 3FL avaliados se mantém a uma distância de no máximo 5% de acurácia nessa etapa inicial, porém obtendo latências inferiores ao cenário tradicional.

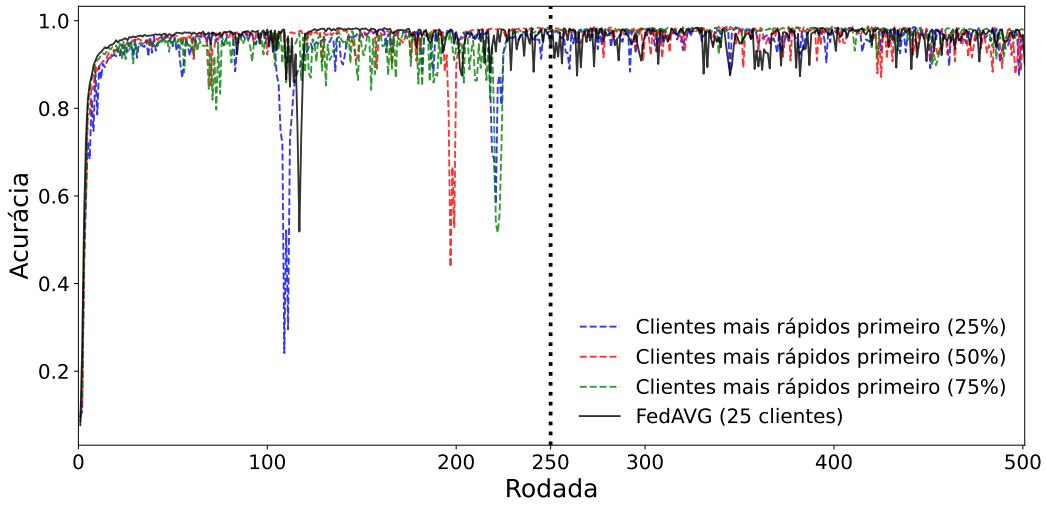


Figura 6.4: Desempenho de classificação do algoritmo 3FL para diferentes latências de clientes participantes.

Uma possível intuição ao observar a Figura 6.4 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. No entanto, como pode ser visto na Figura 6.5, os clientes mais lentos, quando incorporados ao treinamento, impedem que o modelo tenha seu desempenho degradado

ao encontrar mínimos locais durante o treinamento, o que é evidenciado pela ausência de vales na acurácia de treinamento ao incluir todos os clientes. Ademais, fundamentalmente, o objetivo do aprendizado federado é produzir um modelo que possa ser utilizado por todos os clientes participantes, ou seja, sempre será necessário incluir o resto dos clientes no treinamento a fim de garantir que o desempenho naqueles clientes também seja satisfatório.

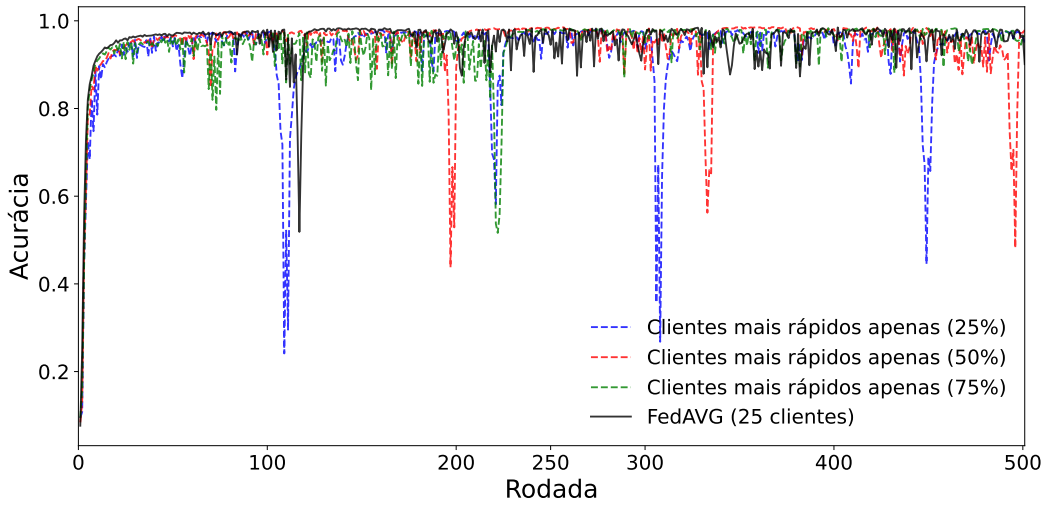


Figura 6.5: Treinamento com apenas os clientes mais rápidos por 500 rodadas.

A Figura 6.6 apresenta uma comparação entre a latência obtida no treinamento federado "tradicional" e a latência obtida através do método 3FL. Nela, os clientes com maior poder computacional foram escolhidos para treinar durante as 250 rodadas iniciais. Naturalmente, é possível observar latências iniciais menores no cenário 3FL. Este ganho em velocidade de execução é desejável, desde que seja acompanhado por um ganho de desempenho do modelo de aprendizado, ou até mesmo um desempenho equiparável. Os valores de latência usados na simulação foram medidos a partir da simulação em dois sistemas com capacidades computacionais diferentes, como descrito no Capítulo 5. Um dos sistemas atingiu um tempo médio de execução de 61 segundos, enquanto o outro sistema atingiu um tempo médio de execução de 23 segundos. Os dois valores observados foram usados nos experimentos como tempo médio dos clientes mais lentos e clientes mais rápidos, respectivamente.

6.4 Aprendizado Federado Orientado a Dados

A Figura 6.7 compara o desempenho da proposta DOFL em diferentes cenários com o algoritmo FedAVG. As 250 primeiras rodadas de aprendizado foram executadas apenas com os clientes que possuem mais dados em seu conjunto de dados local, enquanto o resto do treinamento foi realizado com todos os clientes. As simulações

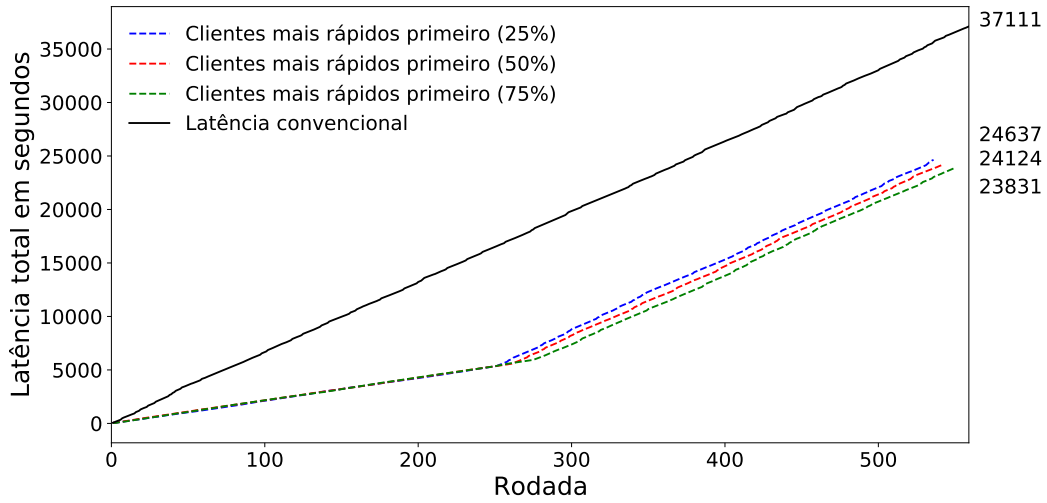


Figura 6.6: Comparação de latências nos cenários simulados. As simulações foram feitas com 25 clientes, onde diferentes combinações de clientes atingem menor latência durante uma rodada e são usados durante e etapa inicial do 3FL, enquanto o restante do treinamento é feito utilizando todos os clientes.

foram feitas com 25 clientes participantes. Em cada cenário simulado, os clientes selecionados para iniciar o treinamento, ou seja, aqueles com mais amostras em seu conjunto de dados locais, possuem o dobro de amostras em relação aos outros participantes. Isto foi atingido através do uso de *data augmentation*, visto que redistribuir amostras resultaria em um conjunto de dados desbalanceado ou forçaria o uso de amostras repetidas, as duas técnicas impediriam uma comparação justa com o cenário base representa pela curva preta.

Na Figura 6.7, pode-se observar que a etapa inicial, quando apenas clientes com mais dados participam do treinamento, contém mais vales de desempenho em relação ao cenário FedAVG tradicional, o que sugere a ocorrência de sobreajustes durante o treinamento. No entanto, após incorporar o resto dos clientes ao treinamento, a proposta DOFL, para os três cenários avaliados, atinge desempenho equiparável, ou até mesmo superior, em relação ao cenário FedAVG tradicional.

A técnica DOFL proposta não garante uma redução óbvia na latência, conforme visto na proposta 3FL. Porém, para todos os cenários de aprendizado federado, um número menor de clientes pode resultar em menores latências, visto que a probabilidade de pelo menos um cliente se desconectar durante uma rodada diminui com o número de clientes, e a chance de o servidor recorrer ao tempo de *timeout* da rodada também é menor. Entretanto, a técnica DOFL reduz a utilização dos recursos de rede e dos recursos computacionais, em geral, visto que ela utiliza menos clientes durante metade do treinamento e que também reduz a necessidade da transmissão de atualizações de gradientes dos clientes ao servidor.

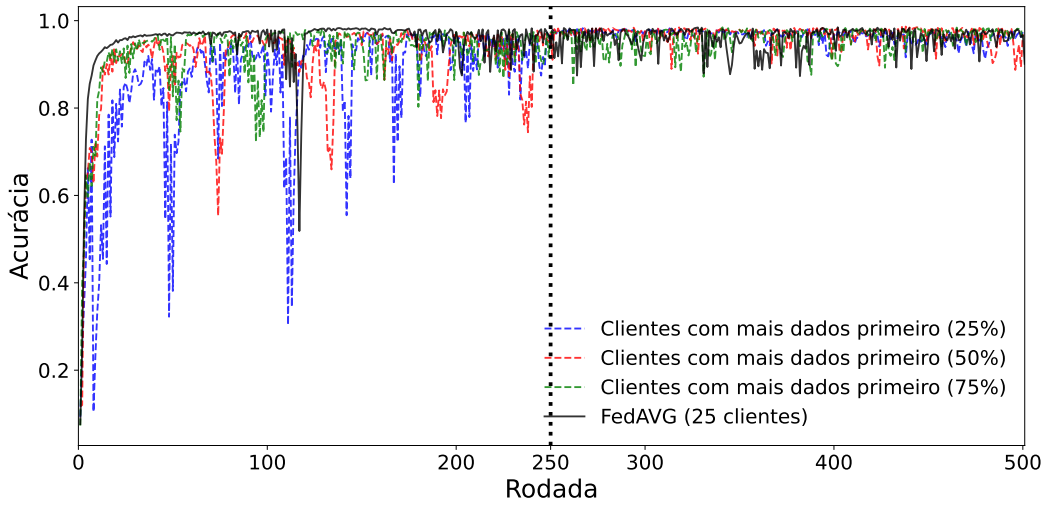


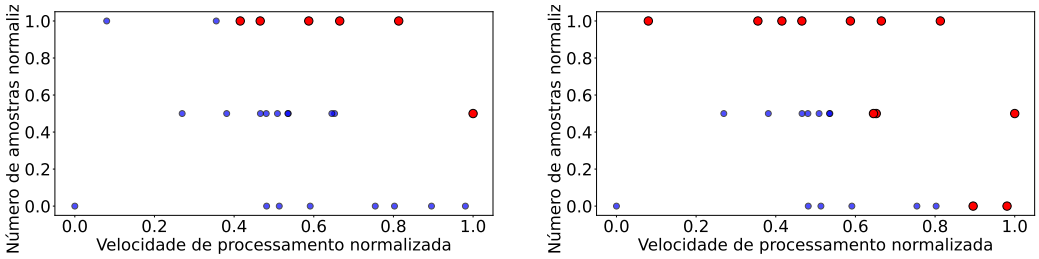
Figura 6.7: Desempenho de classificação do algoritmo DOFL para diferentes distribuições de dados de clientes participantes.

6.5 Aprendizado Federado Híbrido

Como descrito no Capítulo 4, para 25 clientes participantes, uma porção aleatória foi selecionada para atuar com um conjunto de dados maior. A fim de manter as comparações consistentes, tal qual no experimento da proposta DOFL, os clientes com mais dados usam técnicas de *data augmentation* para obter um conjunto de dados local duas vezes maior que o dos outros clientes. Após definir os conjuntos de dados de cada cliente, uma porção aleatória dos clientes foi selecionada para atuar como um cliente mais rápido. Naturalmente, para as mesmas condições computacionais, há uma relação inversa entre o volume de dados e o tempo de processamento. No entanto, devido à heterogeneidade de dispositivos em cenários *cross-device*, é possível obter menor latência em clientes com maiores conjuntos de dados. Logo, a proposta foi avaliada permitindo a existência de clientes mais rápidos e com mais dados.

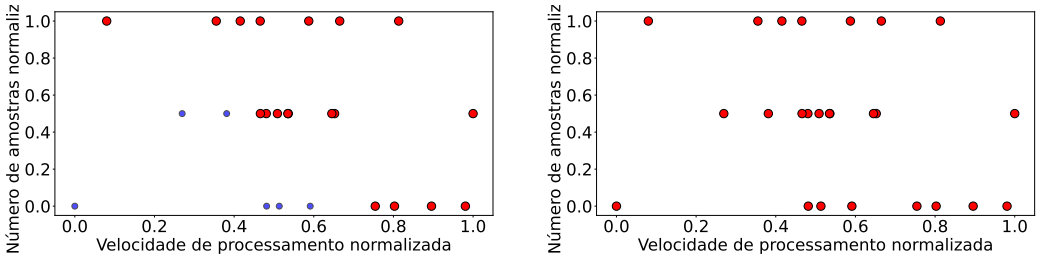
As simulações são feitas atribuindo aleatoriamente conjuntos de dados maiores a clientes e depois selecionando aleatoriamente clientes com menor tempo de execução. Depois, a proposta híbrida avalia diferentes grupos de clientes, de acordo com suas distâncias em relação à interseção dos eixos X e Y. O resultado da amostragem de clientes pode ser vista na Figura 6.8. A partir dela, simulações foram feitas com diferentes clientes participando das primeiras 250 rodadas de treinamento. Finalmente, é possível aferir o desempenho dos modelos e as latências esperadas durante cada treinamento.

A Figura 6.9 apresenta o desempenho da proposta Hybrid-FL para os diferentes cenários ilustrados na Figura 6.8. As 250 primeiras rodadas de aprendizado foram executadas com os clientes considerados de interesse, enquanto o resto do



(a) Cenário com 25 clientes e 25% dos clientes selecionados para o início do treinamento.

(b) Cenário com 25 clientes e 50% dos clientes selecionados para o início do treinamento.



(c) Cenário com 25 clientes e 75% dos clientes selecionados para o início do treinamento.

(d) Cenário com 25 clientes e todos os clientes selecionados para o início do treinamento, este cenário representa o aprendizado federado "tradicional".

Figura 6.8: Possíveis limiares de seleção de clientes do Hybrid-FL.

treinamento foi realizado com todos os clientes. As simulações foram feitas com 25 clientes participantes.

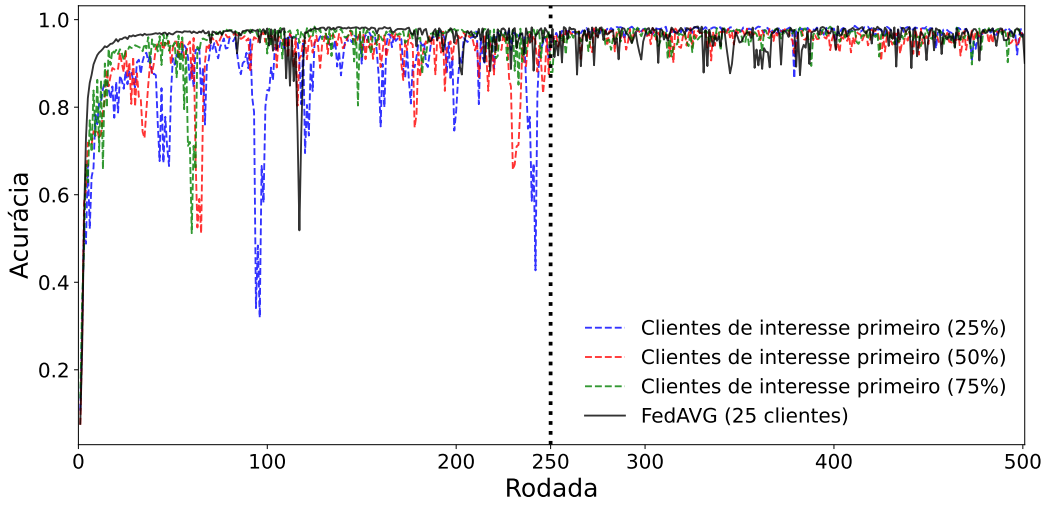


Figura 6.9: Desempenho de classificação do algoritmo híbrido para diferentes configurações dos clientes participantes.

Assim como nas propostas 3FL e DOFL, a proposta Hybrid-FL atinge desempenho equiparável à abordagem tradicional. Também é possível perceber, notavelmente no cenário com 25% de clientes selecionados primeiro, a presença de vales de desempenho. Assim como nas outras duas propostas, a presença de menos clientes

durante o treinamento pode levar o modelo ao sobreajuste. No entanto, a inclusão dos clientes restantes a partir da rodada 250 faz com que o modelo global atinja convergência para os 3 cenários simulados. Logo, a proposta Hybrid-FL se apresenta como outra alternativa para melhorar o desempenho do modelo e ainda assim reduzir a latência total de treinamento.

6.6 Comparação

A Figura 6.10 compara o melhor desempenho obtido conforme a configuração apropriada de cada cenário avaliado. As acurácias apresentadas no gráfico são calculadas através da média de desempenho nas últimas 10 rodadas para cada configuração. Esta 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.

É possível observar na Figura 6.10 que o desempenho das três técnicas propostas foi equiparável ao da técnica tradicional (FedAVG). Logo, os ganhos na forma de redução de latência e redução do uso de recursos computacionais da rede, não se apresentam em detrimento do desempenho de classificação global.

Também se nota 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, por exemplo, o desempenho é inferior aos cenários de 3FL com mais e com menos clientes iniciais.

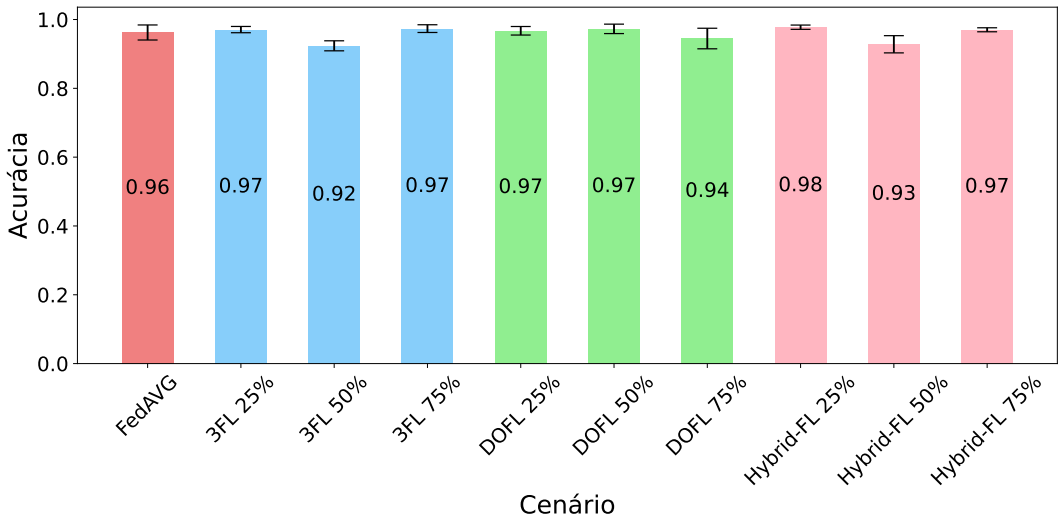


Figura 6.10: Desempenhos de classificação nos cenários avaliados.

A Figura 6.11 apresenta as latências obtidas em todos os cenários simulados. 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. Já a proposta DOFL reduz ligeiramente a latência total de treinamento, por incluir menos clientes no início do treinamento, o que resulta em menor probabilidade de *timeout*. Em contrapartida, a proposta Hybrid-FL se mostrou capaz de reduzir a latência quando inclui poucos clientes. No entanto, aumentar o raio de seleção pode fazer com que o volume de dados de clientes mais lentos seja usado para a inclusão do cliente no treinamento, aumentando assim a latência total.

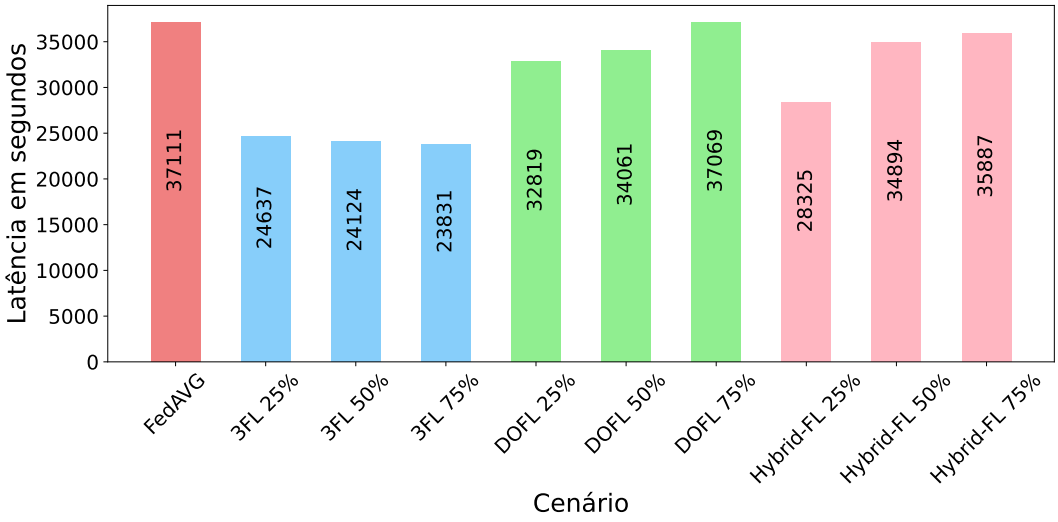


Figura 6.11: Latências totais de treinamento nos cenários avaliados.

6.7 Aprendizado Federado *Fastest-First* em Tempo Real

Como descrito na Seção 6.6, a proposta 3FL se apresentou como mais promissora para reduzir a latência total de treinamento enquanto mantém um alto desempenho de classificação. Tendo isso em mente, a proposta 3FL foi avaliada em um cenário mais similar a implementações reais.

A avaliação utilizou um servidor virtual na nuvem AWS (*Amazon Web Services*), um servidor localizado em uma rede interna na borda da rede e um computador pessoal conectado à rede 4G.

O servidor foi utilizado para simular os clientes mais rápidos, enquanto o computador pessoal, com menor capacidade computacional, foi utilizado para simular os clientes mais lentos. A Figura 6.12 apresenta o desempenho de classificação da proposta 3FL no cenário descrito. Especificamente, o cenário com 25 clientes e 50%

dos clientes mais rápidos selecionados foi avaliada.

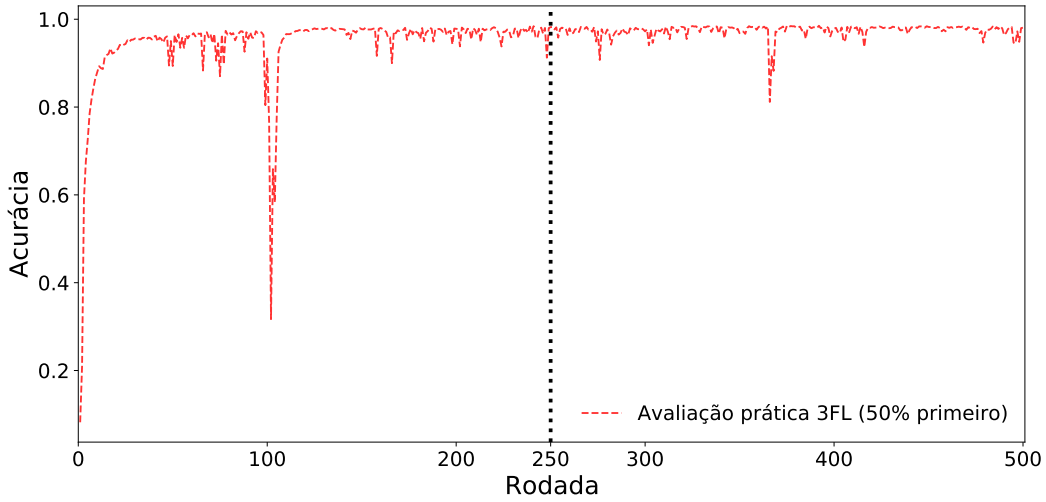


Figura 6.12: Desempenho de classificação do algoritmo 3FL com servidor na nuvem e clientes nas bordas da rede.

Assim como os resultados analisados na Seção 6.3, pode-se observar que o desempenho de classificação do modelo é capaz de convergir na análise em tempo real.

Capítulo 7

Conclusão e Trabalhos Futuros

Esta dissertação apresentou três novas formas de implementar estratégias de seleção de clientes para reduzir a latência durante o aprendizado federado. As propostas criadas foram avaliadas em um conjunto de dados representativo 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. As técnicas DOFL e Hybrid-FL também se mostraram apropriadas em reduzir o uso de recursos computacionais da rede, enquanto atingem desempenho equiparável ao das outras técnicas. Também foi demonstrado que as três técnicas avaliadas foram capazes de atingir desempenhos de classificação superiores à abordagem tradicional. Finalmente, uma avaliação da técnica 3FL foi conduzida em um cenário real, que comprovou a efetividade da proposta para sistemas em tempo real.

Como trabalho futuro, pretende-se expandir os cenários de simulação a fim validar a robustez 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. Finalmente, também pretende-se avaliar o impacto de ajustes dinâmicos dos algoritmos após um determinado número de rodadas de treinamento.

Referências Bibliográficas

[1] BOCHIE, K., GILBERT, M. S., GANTERT, L., et al. "A survey on deep learning for challenged networks: Applications and trends", *Journal of Network and Computer Applications*, v. 194, pp. 103213, Nov 2021. ISSN: 1084-8045. doi: https://doi.org/10.1016/j.jnca.2021.103213. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804521002149>.

[2] BOCHIE, K., SAMMARCO, M., DETYNIECKI, M., et al. "Análise do Aprendizado Federado em Redes Móveis". Em: *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 71–84, Porto Alegre, RS, Brasil, 2021. SBC.

[3] LIU, Y., HUANG, A., LUO, Y., et al. "FedVision: An Online Visual Object Detection Platform Powered by Federated Learning", *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 34, n. 08, pp. 13172–13179, Apr. 2020. doi: 10.1609/aaai.v34i08.7021. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/7021>.

[4] SHAFIQUE, K., KHAWAJA, B. A., SABIR, F., et al. "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios", *IEEE Access*, v. 8, pp. 23022–23040, 2020. doi: 10.1109/ACCESS.2020.2970118.

[5] NAEEM, M., JAMAL, T., DIAZ-MARTINEZ, J., et al. "Trends and Future Perspective Challenges in Big Data". Em: Pan, J.-S., Balas, V. E., Chen, C.-M. (Eds.), *Advances in Intelligent Data Analysis and Applications*, pp. 309–325, Singapore, 2022. Springer Singapore. ISBN: 978-981-16-5036-9.

[6] SILVA, P., MONTEIRO, E., SIMÕES, P. "Privacy in the Cloud: A Survey of Existing Solutions and Research Challenges", *IEEE Access*, v. 9, pp. 10473–10497, 2021. doi: 10.1109/ACCESS.2021.3049599.

[7] CHENG, P., ROEDIG, U. "Personal Voice Assistant Security and Privacy—A Survey", *Proceedings of the IEEE*, v. 110, n. 4, pp. 476–507, 2022. doi: 10.1109/JPROC.2022.3153167.

[8] RIMOL, M. "Gartner Identifies Top Five Trends in Privacy Through 2024", *Gartner*, 2022. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2022-05-31-gartner-identifies-top-five-trends-in-privacy-through-2024>.

[9] MCMAHAN, B., MOORE, E., RAMAGE, D., et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". Em: Singh, A., Zhu, J. (Eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, v. 54, *Proceedings of Machine Learning Research*, pp. 1273–1282, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. Disponível em: <http://proceedings.mlr.press/v54/mcmahan17a.html>.

[10] NETO, H. C., MATTOS, D., FERNANDES, N. "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*, pp. 280–293, Porto Alegre, RS, Brasil, 2021. SBC. doi: 10.5753/sbrc.2021.16727. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/16727>.

[11] DE CALDAS FILHO, F. L., SOARES, S. C. M., OROSKI, E., et al. "Botnet Detection and Mitigation Model for IoT Networks Using Federated Learning", *Sensors*, v. 23, n. 14, pp. 6305, jul. 2023. ISSN: 1424-8220. doi: 10.3390/s23146305. Disponível em: <http://dx.doi.org/10.3390/s23146305>.

[12] NAEEM, A., ANEES, T., NAQVI, R. A., et al. "A Comprehensive Analysis of Recent Deep and Federated-Learning-Based Methodologies for Brain Tumor Diagnosis", *Journal of Personalized Medicine*, v. 12, n. 2, 2022. ISSN: 2075-4426. doi: 10.3390/jpm12020275. Disponível em: <https://www.mdpi.com/2075-4426/12/2/275>.

[13] QU, Y., UDDIN, M. P., GAN, C., et al. "Blockchain-Enabled Federated Learning: A Survey", *ACM Comput. Surv.*, v. 55, n. 4, nov 2022. ISSN: 0360-0300. doi: 10.1145/3524104. Disponível em: <https://doi.org/10.1145/3524104>.

[14] KAIROUZ, P., MCMAHAN, H. B., AVENT, B., et al. "Advances and Open Problems in Federated Learning". 2021.

[15] BOCHIE, K., SAMMARCO, M., CAMPISTA, M. "An Analysis of Federated Learning Performance in Mobile Networks". Em: *VCC 2023 - IEEE Virtual Conference on Communications*, pp. 1–6, Nov 2023.

[16] LI, T., SAHU, A. K., TALWALKAR, A., et al. "Federated Learning: Challenges, Methods, and Future Directions", *IEEE Signal Processing Magazine*, v. 37, n. 3, pp. 50–60, 2020. doi: 10.1109/MSP.2020.2975749.

[17] BOCHIE, K., GILBERT, M. S., GANTERT, L., et al. "Aprendizado Profundo em Redes Desafiadoras: Conceitos e Aplicações". Em: *Minicursos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 140–189, 12 2020. doi: 10.5753/sbc.5033.7.4.

[18] BUYUKATES, B., ULUKUS, S. "Timely Communication in Federated Learning". Em: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2021. doi: 10.1109/INFOCOMWKSHPS51825.2021.9484497.

[19] YANG, Q., LIU, Y., CHEN, T., et al. "Federated Machine Learning: Concept and Applications", *ACM Trans. Intell. Syst. Technol.*, v. 10, n. 2, jan 2019. ISSN: 2157-6904. doi: 10.1145/3298981. Disponível em: <`https://doi.org/10.1145/3298981`>.

[20] FU, C., ZHANG, X., JI, S., et al. "Label Inference Attacks Against Vertical Federated Learning". Em: *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1397–1414, Boston, MA, ago. 2022. USENIX Association. ISBN: 978-1-939133-31-1. Disponível em: <`https://www.usenix.org/conference/usenixsecurity22/presentation/fu-chong`>.

[21] XU, X., QI, Z., HAN, X., et al. "Predicting anticancer drug sensitivity on distributed data sources using federated deep learning", *Heliyon*, v. 9, n. 8, pp. e18615, 2023. ISSN: 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2023.e18615. Disponível em: <`https://www.sciencedirect.com/science/article/pii/S2405844023058231`>.

[22] YANG, W., WANG, N., GUAN, Z., et al. "A Practical Cross-Device Federated Learning Framework over 5G Networks", *IEEE Wireless Communications*, v. 29, n. 6, pp. 128–134, 2022. doi: 10.1109/MWC.005.2100435.

[23] REHMAN, M. H. U., DIRIR, A. M., SALAH, K., et al. "TrustFed: A Framework for Fair and Trustworthy Cross-Device Federated Learning in IIoT", *IEEE Transactions on Industrial Informatics*, v. 17, n. 12, pp. 8485–8494, 2021. doi: 10.1109/TII.2021.3075706.

[24] KARIMIREDDY, S. P., JAGGI, M., KALE, S., et al. "Breaking the centralized barrier for cross-device federated learning". Em: Ranzato, M., Beygelzimer, A., Dauphin, Y., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 34, pp. 28663–28676. Curran Associates, Inc., 2021. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2021/file/f0e6be4ce76ccfa73c5a540d992d0756-Paper.pdf>.

[25] TANG, M., WONG, V. W. "An Incentive Mechanism for Cross-Silo Federated Learning: A Public Goods Perspective". Em: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, 2021. doi: 10.1109/INFOCOM42981.2021.9488705.

[26] HUANG, Y., CHU, L., ZHOU, Z., et al. "Personalized Cross-Silo Federated Learning on Non-IID Data", *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 35, n. 9, pp. 7865–7873, May 2021. doi: 10.1609/aaai.v35i9.16960. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/16960>.

[27] MARFOQ, O., XU, C., NEGLIA, G., et al. "Throughput-Optimal Topology Design for Cross-Silo Federated Learning". Em: Larochelle, H., Ranzato, M., Hadsell, R., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 33, pp. 19478–19487. Curran Associates, Inc., 2020. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2020/file/e29b722e35040b88678e25a1ec032a21-Paper.pdf>.

[28] TAK, A., CHERKAOUI, S. "Federated Edge Learning: Design Issues and Challenges", *IEEE Network*, v. 35, n. 2, pp. 252–258, 2021. doi: 10.1109/MNET.011.2000478.

[29] LI, H., WANG, R., WU, J., et al. "Federated Edge Learning via Reconfigurable Intelligent Surface with One-Bit Quantization". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 1055–1060, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10001550.

[30] ZHANG, F., GE, J., WONG, C., et al. "Optimizing Federated Edge Learning on Non-IID Data via Neural Architecture Search". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685909.

[31] HSIEH, K., PHANISHAYEE, A., MUTLU, O., et al. "The Non-IID Data Quagmire of Decentralized Machine Learning". Em: III, H. D., Singh, A. (Eds.), *Proceedings of the 37th International Conference on Machine Learning*, v.

119, *Proceedings of Machine Learning Research*, pp. 4387–4398. PMLR, 13–18 Jul 2020. Disponível em: <https://proceedings.mlr.press/v119/hsieh20a.html>.

[32] MO, K., CHEN, C., LI, J., et al. "Two-Dimensional Learning Rate Decay: Towards Accurate Federated Learning with Non-IID Data". Em: *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2021. doi: 10.1109/IJCNN52387.2021.9533708.

[33] SOUZA, L., CAMILO, G., SAMMARCO, M., et al. "Aprendizado Federado com Agrupamento Hierárquico de Clientes para Aumento da Acurácia". Em: *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 545–558, Porto Alegre, RS, Brasil, 2022. SBC. doi: 10.5753/sbrc.2022.222371. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/21196>.

[34] ASAD, M., OTOUM, S., SHAUKAT, S. "Resource and Heterogeneity-aware Clients Eligibility Protocol in Federated Learning". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 1140–1145, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000884.

[35] CAO, Y., MAGHSUDI, S., OHTSUKI, T. "Mobility-Aware Routing and Caching: A Federated Learning Assisted Approach". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500804.

[36] FENG, C., YANG, H. H., HU, D., et al. "Federated Learning with User Mobility in Hierarchical Wireless Networks". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 01–06, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685129.

[37] KARAKUS, C., SUN, Y., DIGGAVI, S., et al. "Straggler Mitigation in Distributed Optimization Through Data Encoding". Em: Guyon, I., Luxburg, U. V., Bengio, S., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 30. Curran Associates, Inc., 2017. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/663772ea088360f95bac3dc7ffb841be-Paper.pdf>.

[38] VU, T. T., NGO, D. T., NGO, H. Q., et al. "Straggler Effect Mitigation for Federated Learning in Cell-Free Massive MIMO". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500541.

[39] KUMAR, S., SCHLEGEL, R., ROSNES, E., et al. "Coding for Straggler Mitigation in Federated Learning". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 4962–4967, May 2022. doi: 10.1109/ICC45855.2022.9838986.

[40] KINGMA, D. P., BA, J. "Adam: A Method for Stochastic Optimization". 2017.

[41] LUO, S., FAN, P., XING, H., et al. "Eliminating Communication Bottlenecks in Cross-Device Federated Learning with In-Network Processing at the Edge". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 4601–4606, 2022. doi: 10.1109/ICC45855.2022.9838381.

[42] LIU, X., DENG, Y., MAHMOODI, T. "Energy Efficient User Scheduling for Hybrid Split and Federated Learning in Wireless UAV Networks". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 1–6, May 2022. doi: 10.1109/ICC45855.2022.9882277.

[43] GUPTA, O., RASKAR, R. "Distributed learning of deep neural network over multiple agents", *Journal of Network and Computer Applications*, v. 116, pp. 1–8, 2018. ISSN: 1084-8045. doi: https://doi.org/10.1016/j.jnca.2018.05.003. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804518301590>.

[44] THAPA, C., MAHAWAGA ARACHCHIGE, P. C., CAMTEPE, S., et al. "SplitFed: When Federated Learning Meets Split Learning", *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 36, n. 8, pp. 8485–8493, Jun. 2022. doi: 10.1609/aaai.v36i8.20825. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/20825>.

[45] REISIZADEH, A., TZIOTIS, I., HASSANI, H., et al. "Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity", *IEEE Journal on Selected Areas in Information Theory*, v. 3, n. 2, pp. 197–205, 2022. doi: 10.1109/JSAIT.2022.3205475.

[46] HADDADPOUR, F., KAMANI, M. M., MOKHTARI, A., et al. "Federated Learning with Compression: Unified Analysis and Sharp Guarantees". Em: Banerjee, A., Fukumizu, K. (Eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, v. 130, *Proceedings of Machine Learning Research*, pp. 2350–2358. PMLR, 13–15 Apr 2021. Disponível em: <https://proceedings.mlr.press/v130/haddadpour21a.html>.

[47] WANG, J., LIU, Q., LIANG, H., et al. "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization". Em: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN: 9781713829546.

[48] ALBASEER, A., ABDALLAH, M., AL-FUQAHA, A., et al. "Client Selection Approach in Support of Clustered Federated Learning over Wireless Edge Networks". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685938.

[49] CHU, D., JAAFAR, W., YANIKOMEROGLU, H. "On the Design of Communication-Efficient Federated Learning for Health Monitoring". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 1128–1133, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10001077.

[50] AGARWAL, N., SONDHI, A., CHOPRA, K., et al. "Transfer Learning: Survey and Classification". Em: Tiwari, S., Trivedi, M. C., Mishra, K. K., et al. (Eds.), *Smart Innovations in Communication and Computational Sciences*, pp. 145–155, Singapore, 2021. Springer Singapore. ISBN: 978-981-15-5345-5.

[51] PAN, S. J., YANG, Q. "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, pp. 1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

[52] KIM, Y., HAKIM, E. A., HARALDSON, J., et al. "Dynamic Clustering in Federated Learning". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500877.

[53] MUKHERJEE, S., ASNANI, H., LIN, E., et al. "ClusterGAN: Latent Space Clustering in Generative Adversarial Networks". Em: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN: 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33014610. Disponível em: <`https://doi.org/10.1609/aaai.v33i01.33014610`>.

[54] MANSOUR, Y., MOHRI, M., RO, J., et al. "Three Approaches for Personalization with Applications to Federated Learning", *CoRR*, v. abs/2002.10619, 2020. Disponível em: <`https://arxiv.org/abs/2002.10619`>.

[55] SAVARESI, S. M., BOLEY, D. L., BITTANTI, S., et al. "Cluster selection in divisive clustering algorithms". Em: *Proceedings of the 2002 SIAM International Conference on Data Mining (SDM)*, pp. 299–314, Virginia, USA, SIAM International Conference on Data Mining (SDM), 2002. doi: 10.1137/1.9781611972726.18. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9781611972726.18>.

[56] HYNDMAN, R. J., WANG, E., LAPTEV, N. "Large-Scale Unusual Time Series Detection". Em: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1616–1619, 2015. doi: 10.1109/ICDMW.2015.104.

[57] JIANG, B., DU, J., JIANG, C., et al. "Communication-Efficient Device Scheduling via Over-the-Air Computation for Federated Learning". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 173–178, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000727.

[58] SHI, W., ZHOU, S., NIU, Z. "Device Scheduling with Fast Convergence for Wireless Federated Learning". Em: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2020. doi: 10.1109/ICC40277.2020.9149138.

[59] XUE, D., LUO, J., JIANG, C., et al. "Cost-Aware Hierarchical Federated Learning via Over-the-Air Computing". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 4728–4733, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000988.

[60] TIAN, Y., ZHANG, Z., YANG, Z., et al. "Hierarchical Federated Learning with Adaptive Clustering on Non-IID Data". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 627–632, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000749.

[61] LIU, J., CHANG, Z., MIN, G., et al. "Incentive Mechanism Design For Federated Learning in Multi-access Edge Computing". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 3454–3459, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000933.

[62] CHEN, Y., NING, Y., SLAWSKI, M., et al. "Asynchronous Online Federated Learning for Edge Devices with Non-IID Data". Em: *2020 IEEE International Conference on Big Data (Big Data)*, pp. 15–24, 2020. doi: 10.1109/BigData50022.2020.9378161.

[63] ZHANG, Q., PALACHARLA, P., SEKIYA, M., et al. "Demo: A Blockchain Based Protocol for Federated Learning". Em: *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pp. 1–2, 2020. doi: 10.1109/ICNP49622.2020.9259388.

[64] HIESSL, T., SCHALL, D., KEMNITZ, J., et al. "Industrial Federated Learning – Requirements and System Design". Em: De La Prieta, F., Mathieu, P., Rincón Arango, J. A., et al. (Eds.), *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness. The PAAMS Collection*, pp. 42–53, Cham, 2020. Springer International Publishing. ISBN: 978-3-030-51999-5.

[65] CAO, X., FANG, M., LIU, J., et al. "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping". Em: *The Network and Distributed System Security Symposium (NDSS) 2021*, 01 2021. doi: 10.14722/ndss.2021.24434.

[66] ZHANG, Z., CAO, X., JIA, J., et al. "FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients". Em: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, p. 2545–2555, New York, NY, USA, 2022. Association for Computing Machinery. ISBN: 9781450393850. doi: 10.1145/3534678.3539231. Disponível em: <`https://doi.org/10.1145/3534678.3539231`>.

[67] ABADI, M., AGARWAL, A., BARHAM, P., et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems". 2015. Disponível em: <`https://www.tensorflow.org/`>. Software available from tensorflow.org.

[68] CHOLLET, F., OTHERS. "Keras". `https://keras.io`, 2015.

[69] BEUTEL, D. J., TOPAL, T., MATHUR, A., et al. "Flower: A Friendly Federated Learning Research Framework", *arXiv preprint arXiv:2007.14390*, 2020.

[70] LECUN, Y., BOTTOU, L., BENGIO, Y., et al. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791.

[71] SANDLER, M., HOWARD, A., ZHU, M., et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". Em: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.

4510–4520, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00474. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474>.

[72] SIMONYAN, K., ZISSERMAN, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition". Em: Bengio, Y., LeCun, Y. (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. Disponível em: <http://arxiv.org/abs/1409.1556>.

[73] HE, K., ZHANG, X., REN, S., et al. "Deep Residual Learning for Image Recognition". Em: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[74] WANJIKU, R. N., NDERU, L., KIMWELE, M. "Improved transfer learning using textural features conflation and dynamically fine-tuned layers", *PeerJ Computer Science*, v. 9, 2023. doi: 10.7717/peerj-cs.1601.

[75] XU, Z., ZHANG, Y., ANDREW, G., et al. "Federated Learning of Gboard Language Models with Differential Privacy". Em: Sitaram, S., Beigman Klebanov, B., Williams, J. D. (Eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 629–639, Toronto, Canada, jul. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.60. Disponível em: <https://aclanthology.org/2023.acl-industry.60>.

[76] SIVEK, G., RILEY, M. "Spatial Model Personalization in Gboard", *Proc. ACM Hum.-Comput. Interact.*, v. 6, n. MHCI, sep 2022. doi: 10.1145/3546737. Disponível em: <https://doi.org/10.1145/3546737>.

[77] JI, Y., KOU, Z., ZHONG, X., et al. "Client Selection and Bandwidth Allocation for Federated Learning: An Online Optimization Perspective". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 5075–5080, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10001492.

[78] WANG, F., GURSOY, M. C., VELIPASALAR, S. "Communication-Efficient and Privacy-Preserving Feature-based Federated Transfer Learning". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 3875–3880, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000612.

[79] GAO, Y., YE, Z., YU, H., et al. "Multi-Resource Allocation for On-Device Distributed Federated Learning Systems". Em: *GLOBECOM 2022 - 2022*

IEEE Global Communications Conference, pp. 160–165, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10000935.

[80] GUO, W., LI, R., HUANG, C., et al. "Optimized Device Selection and Power Control for Wireless Federated Learning". Em: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 4710–4715, Dec 2022. doi: 10.1109/GLOBECOM48099.2022.10001306.

[81] ALBELAIHI, R., SUN, X., CRAFT, W. D., et al. "Adaptive Participant Selection in Heterogeneous Federated Learning". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685077.

[82] WANG, D., WANG, B., ZHANG, J., et al. "CFL-HC: A Coded Federated Learning Framework for Heterogeneous Computing Scenarios". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685962.

[83] WAN, S., LU, J., FAN, P., et al. "Convergence analysis and Design principle for Federated learning in Wireless network". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 01–06, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685504.

[84] WANG, P., LI, L., WANG, D., et al. "Enabling Efficient Scheduling Policy in Intelligent Reflecting Surface Aided Federated Learning". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 01–05, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685096.

[85] ZHENG, X.-Y., LEE, M.-C., HONG, Y.-W. P. "Knowledge Caching for Federated Learning". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685861.

[86] NGUYEN, H. T., MORABITO, R., KIM, K. T., et al. "On-the-fly Resource-Aware Model Aggregation for Federated Learning in Heterogeneous Edge". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685893.

[87] PRAKASH, P., DING, J., WU, M., et al. "To Talk or to Work: Delay Efficient Federated Learning over Mobile Edge Devices". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685793.

[88] HAMDI, R., CHEN, M., BEN SAID, A., et al. "User Scheduling in Federated Learning over Energy Harvesting Wireless Networks". Em: *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2021. doi: 10.1109/GLOBECOM46510.2021.9685801.

[89] HU, Q., LI, F., ZOU, X., et al. "Correlated Participation Decision Making for Federated Edge Learning". Em: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Dec 2020. doi: 10.1109/GLOBECOM42002.2020.9321981.

[90] YIN, B., CHEN, Z., TAO, M. "Joint User Scheduling and Resource Allocation for Federated Learning over Wireless Networks". Em: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Dec 2020. doi: 10.1109/GLOBECOM42002.2020.9348225.

[91] QOLOMANY, B., AHMAD, K., AL-FUQAHA, A., et al. "Particle Swarm Optimized Federated Learning For Industrial IoT and Smart City Services". Em: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Dec 2020. doi: 10.1109/GLOBECOM42002.2020.9322464.

[92] ALI, M. I., GAO, F., MILEO, A. "CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets". Em: *In proceedings of ISWC 2015 - 14th International Semantic Web Conference*, pp. 374–389, Bethlehem, PA, USA, 2015. W3C.

[93] AZURE. "Azure AI Gallery". `https://gallery.azure.ai/`, 2023. Acessado em 02/10/2023.

[94] HE, Y., REN, J., YU, G., et al. "Resource Allocation for Wireless Federated Edge Learning based on Data Importance". Em: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Dec 2020. doi: 10.1109/GLOBECOM42002.2020.9322155.

[95] ZHANG, H., TAO, M., SHI, Y., et al. "Federated Multi-Task Learning with Non-Stationary Heterogeneous Data". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 4950–4955, May 2022. doi: 10.1109/ICC45855.2022.9838703.

[96] AYGÜN, O., KAZEMI, M., GÜNDÜZ, D., et al. "Hierarchical Over-the-Air Federated Edge Learning". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 3376–3381, May 2022. doi: 10.1109/ICC45855.2022.9839230.

[97] LIU, S., YU, G., CHEN, X., et al. "Joint User Association and Resource Allocation for Wireless Hierarchical Federated Learning with Non-IID Data". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 74–79, May 2022. doi: 10.1109/ICC45855.2022.9839164.

[98] ZHOU, X., DENG, Y., XIA, H., et al. "Resource Allocation for Time-triggered Federated Learning over Wireless Networks". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 2810–2815, May 2022. doi: 10.1109/ICC45855.2022.9838329.

[99] LOTFI, F., SEMIARI, O., SAAD, W. "Semantic-Aware Collaborative Deep Reinforcement Learning Over Wireless Cellular Networks". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 5256–5261, May 2022. doi: 10.1109/ICC45855.2022.9839122.

[100] LIU, C.-H., FENG, K.-T., WEI, L., et al. "Spatio-Temporal Federated Learning for Massive Wireless Edge Networks". Em: *ICC 2022 - IEEE International Conference on Communications*, pp. 2816–2821, May 2022. doi: 10.1109/ICC45855.2022.9838401.

[101] LIU, Y.-J., FENG, G., WANG, J., et al. "Access Control for RAN Slicing based on Federated Deep Reinforcement Learning". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500611.

[102] WANG, Z., ZHANG, Z., WANG, J. "Asynchronous Federated Learning over Wireless Communication Networks". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–7, June 2021. doi: 10.1109/ICC42927.2021.9500860.

[103] OTOUM, S., GUIZANI, N., MOUFTAH, H. "Federated Reinforcement Learning-Supported IDS for IoT-steered Healthcare Systems". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500698.

[104] CHEN, M., SHLEZINGER, N., POOR, H. V., et al. "Joint Resource Management and Model Compression for Wireless Federated Learning". Em: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500815.

[105] YANG, L., LI, L., GUO, X., et al. "Saddle Point Approximation Based Delay Analysis for Wireless Federated Learning". Em: *ICC 2021 - IEEE*

International Conference on Communications, pp. 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500298.

[106] CHEN, M., POOR, H. V., SAAD, W., et al. "Convergence Time Minimization of Federated Learning over Wireless Networks". Em: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2020. doi: 10.1109/ICC40277.2020.9148815.

[107] YOSHIDA, N., NISHIO, T., MORIKURA, M., et al. "Hybrid-FL for Wireless Networks: Cooperative Learning Mechanism Using Non-IID Data". Em: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, June 2020. doi: 10.1109/ICC40277.2020.9149323.

[108] HAO, M., LI, H., XU, G., et al. "Privacy-aware and Resource-saving Collaborative Learning for Healthcare in Cloud Computing". Em: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2020. doi: 10.1109/ICC40277.2020.9148979.

[109] LEI, W., WANG, S., ZHANG, N., et al. "Adaptive Decision-Making Framework for Federated Learning Tasks in Multi-Tier Computing". Em: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–2, May 2022. doi: 10.1109/INFOCOMWKSHPS54753.2022.9798193.

[110] XIN, S., ZHUO, L., XIN, C. "Online Node Cooperation Strategy Design for Hierarchical Federated Learning". Em: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, May 2022. doi: 10.1109/INFOCOMWKSHPS54753.2022.9798016.

[111] ZHENG, J., NI, W., TIAN, H., et al. "Semi-Federated Learning: An Integrated Framework for Pervasive Intelligence in 6G Networks". Em: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, May 2022. doi: 10.1109/INFOCOMWKSHPS54753.2022.9798021.

Apêndice A

Artigos Avaliados Através da Metodologia PRISMA

As Tabelas A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9 e A.10 apresentam os artigos revisados nesta dissertação. As colunas "conjunto de dados" e "clientes" foram usadas para definir a configuração dos experimentos descritos no Capítulo 5.

Tabela A.1: Artigos da conferência GLOBECOM do ano 2022.

Clientes	**Artigo**	**Conjunto de Dados**	**Cenário**
100	Client Selection and Bandwidth Allocation for Federated Learning: An Online Optimization Perspective [77]	MNIST	Redes sem fio
6.250	Communication-Efficient and Privacy-Preserving Feature-based Federated Transfer Learning [78]	CIFAR-10 e ImageNet	Transferência de aprendizado
100	Communication-Efficient Device Scheduling via Over-the-Air Computation for Federated Learning [57]	MNIST	Redes sem fio
5 – 40	Cost-Aware Hierarchical Federated Learning via Over-the-Air Computing [59]	MNIST	FEEL
100	Hierarchical Federated Learning with Adaptive Clustering on Non-IID Data [60]	EMNIST	FEEL
1 – 1.000	Incentive Mechanism Design For Federated Learning in Multi-access Edge Computing [61]	Simulado	FEEL
Modelado por Poisson	Multi-Resource Allocation for On-Device Distributed Federated Learning Systems [79]	Simulado	FEEL
67	On the Design of Communication-Efficient Federated Learning for Health Monitoring [49]	MobiAct	Saúde
20	Optimized Device Selection and Power Control for Wireless Federated Learning [80]	MNIST	Redes sem fio
100	Resource and Heterogeneity-aware Clients Eligibility Protocol in Federated Learning [34]	CIFAR-10 e MNIST	Mitigação de retardatários

Tabela A.2: Artigos da conferência GLOBECOM do ano 2021.

Clientes	Artigo	Conjunto de Dados	Cenário
200	Adaptive Participant Selection in Heterogeneous Federated Learning [81]	Não definido	Redes sem fio
4	CFL-HC: A Coded Federated Learning Framework for Heterogeneous Computing Scenarios [82]	MNIST	CFL
100	Client Selection Approach in Support of Clustered Federated Learning over Wireless Edge Networks [48]	FEMNIST	FEEL e agrupamento
50, 100	Convergence analysis and Design principle for Federated learning in Wireless network [83]	MNIST e CIFAR-10	Redes sem fio
20, 40	Enabling Efficient Scheduling Policy in Intelligent Reflecting Surface Aided Federated Learning [84]	MNIST	FEEL
50	Federated Learning with User Mobility in Hierarchical Wireless Networks [36]	MNIST	Redes sem fio
20	Knowledge Caching for Federated Learning [85]	Simulado	Redes sem fio
1 – 50	On-the-fly Resource-Aware Model Aggregation for Federated Learning in Heterogeneous Edge [86]	MNIST	5G e EdgeAI
10	To Talk or to Work: Delay Efficient Federated Learning over Mobile Edge Devices [87]	MNIST e CIFAR-10	Redes sem fio
10	User Scheduling in Federated Learning over Energy Harvesting Wireless Networks [88]	MNIST	Redes sem fio

Tabela A.3: Artigos da conferência GLOBECOM do ano 2020.

Clientes	Artigo	Conjunto de Dados	Cenário
2 – 16	Correlated Participation Decision Making for Federated Edge Learning [89]	MNIST	FEEL
30	Joint User Scheduling and Resource Allocation for Federated Learning over Wireless Networks [90]	MNIST	Redes sem fio
5	Particle Swarm Optimized Federated Learning For Industrial IoT and Smart City Services [91]	City Pulse EU FP7 [92] e Microsoft Azure Intelligence Gallery [93]	IoT
6	Resource Allocation for Wireless Federated Edge Learning based on Data Importance [94]	CIFAR-10	FEEL

Tabela A.4: Artigos da conferência ICC do ano 2022.

Clientes	Artigo	Conjunto de Dados	Cenário
25	Coding for Straggler Mitigation in Federated Learning [39]	MNIST e Fashion-MNIST	Redes sem fio
100	Energy Efficient User Scheduling for Hybrid Split and Federated Learning in Wireless UAV Networks [42]	MNIST	Redes UAV
1.000	Federated Multi-Task Learning with Non-Stationary Heterogeneous Data [95]	Simulado	Redes sem fio D2D
20	Hierarchical Over-the-Air Federated Edge Learning [96]	MNIST e CIFAR-10	FEEL
30	Joint User Association and Resource Allocation for Wireless Hierarchical Federated Learning with Non-IID Data [97]	CIFAR-10	Redes sem fio
20	Resource Allocation for Time-triggered Federated Learning over Wireless Networks [98]	MNIST	Redes sem fio
10	Semantic-Aware Collaborative Deep Reinforcement Learning Over Wireless Cellular Networks [99]	Não aplicável	Redes de telefonia
1.000	Spatio-Temporal Federated Learning for Massive Wireless Edge Networks [100]	Simulado	FEEL

Tabela A.5: Artigos da conferência ICC do ano 2021.

Clientes	Artigo	Conjunto de Dados	Cenário
1 – 30	Access Control for RAN Slicing based on Federated Deep Reinforcement Learning [101]	Simulado	Redes 5G
100	Asynchronous Federated Learning over Wireless Communication Networks [102]	MNIST	Redes sem fio
30, 149	Dynamic Clustering in Federated Learning [52]	Handover, Italy power demand, Pendigit e Melbourne pedestrian	Redes sem fio
20	Federated Reinforcement Learning-Supported IDS for IoT-steered Healthcare Systems [103]	CICIDS2017	IoT e IDS
15	Joint Resource Management and Model Compression for Wireless Federated Learning [104]	MNIST	Redes sem fio
6.040	Mobility-Aware Routing and Caching: A Federated Learning Assisted Approach [35]	MovieLens 1M	Redes *small-cell*
30	Saddle Point Approximation Based Delay Analysis for Wireless Federated Learning [105]	MNIST	Redes sem fio
Simulados	Straggler Effect Mitigation for Federated Learning in Cell-Free Massive MIMO [38]	Simulado	*Cell-Free Massive MIMO*

Tabela A.6: Artigos da conferência ICC do ano 2020.

Clientes	Artigo	Conjunto de Dados	Cenário
15	Convergence Time Minimization of Federated Learning over Wireless Networks [106]	MNIST	Redes sem fio
20	Device Scheduling with Fast Convergence for Wireless Federated Learning [58]	MNIST	Redes sem fio
1000	Hybrid-FL for Wireless Networks: Cooperative Learning Mechanism Using Non-IID Data [107]	CIFAR-10	Redes sem fio
10, 20, 30, 40, 50	Privacy-aware and Resource-saving Collaborative Learning for Healthcare in Cloud Computing [108]	CIFAR-10	Saúde

Tabela A.7: Artigos da conferência INFOCOM do ano 2022.

Clientes	Artigo	Conjunto de Dados	Cenário
10	Adaptive Decision-Making Framework for Federated Learning Tasks in Multi-Tier Computing [109]	Simulado	FEEL (multi-tier)
50	Online Node Cooperation Strategy Design for Hierarchical Federated Learning [110]	MNIST	FEEL
10	Semi-Federated Learning: An Integrated Framework for Pervasive Intelligence in 6G Networks [111]	MNIST	Redes 6G

Tabela A.8: Artigos da conferência SBRC do ano 2022.

Clientes	Artigo	Conjunto de Dados	Cenário
10	Aprendizado Federado com Agrupamento Hierárquico de Clientes para Aumento da Acurácia [33]	CIFAR-10	Dados não-IID

Tabela A.9: Artigos da conferência SBRC do ano 2021.

Clientes	Artigo	Conjunto de Dados	Cenário
100	FedSA: Arrefecimento Simulado Federado para a Aceleração da Detecção de Intrusão em Ambientes Colaborativos [10]	Simulado	IDS
5, 10, 20	Análise do Aprendizado Federado em Redes Móveis [2]	CIFAR-10	Redes móveis

Tabela A.10: Artigos de outras fontes.

Clientes	**Artigo**	**Conjunto de Dados**	**Cenário**
20, 50, 100	Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity [45]	CIFAR-10, MNIST e simulado	Mitigação de retardatários

Apêndice B

Efeito de um Cliente Ruim na Visualização do Desempenho do Modelo

A Figura B.1 apresenta o resultado de duas simulações de aprendizado federado com 5 e com 10 clientes. Em cada uma das simulações, um dos clientes foi substituído por um preditor com acurácia fixa igual a 70% para exemplificar o impacto de um cliente com desempenho diferente dos demais na visualização dos resultados.

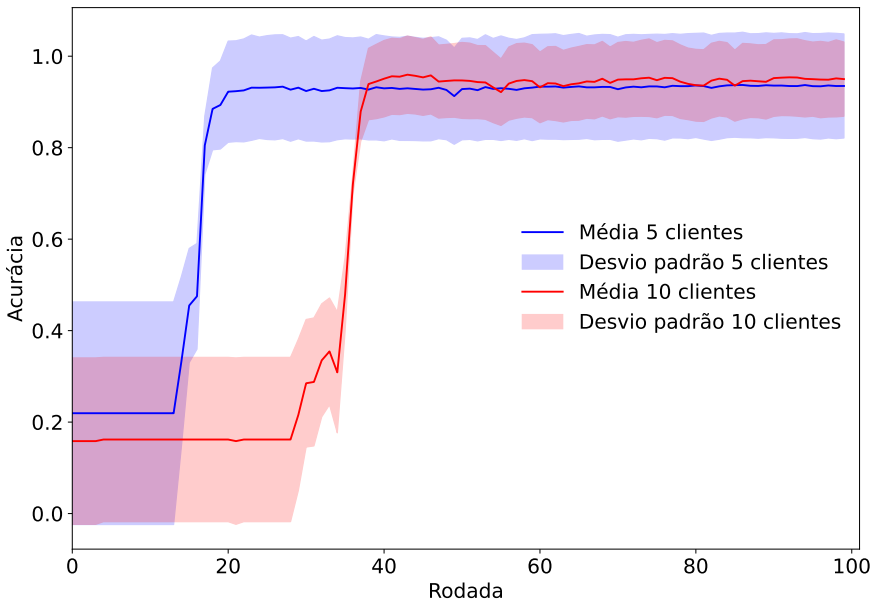


Figura B.1: Nos dois cenários um dos clientes foi substituído por um preditor com desempenho fixo de 70% de acurácia.