Treine Menos, Preveja Mais: *plugin* de Aprendizado Federado habilita alta eficiência em dados heterogêneos

Cláudio G. S. Capanema[1], Joahannes B. D. da Costa[2], Fabrício A. Silva[3], Leandro A. Villas[2], Antonio A. F. Loureiro[1]

[1] Universidade Federal de Minas Gerais (UFMG), Brasil

[2]Universidade Estadual de Campinas (UNICAMP), Brasil

[3]Universidade Federal de Viçosa (UFV), Brasil

{claudio.capanema, loureiro}@dcc.ufmg.br, {jbdc, lvillas}@unicamp.br

fabricio.asilva@ufv.br

Abstract. *Federated learning (FL) emerged as a technique where several devices (also called clients) can learn collaboratively from the orchestration of a central server, providing scalability, privacy and low communication costs. Most research on this topic presents proposals for the model training stage in federated learning, to address various problems such as statistical data heterogeneity, which often represents increased costs (e.g., computational, storage and communication). However, the FedPredict solution was recently proposed, a plugin that operates in the prediction stage of federated learning, which when added can significantly improve the performance of several traditional solutions in data heterogeneity scenarios, without requiring any modification to their original structure or additional training. In this direction, this work presents experiments on a new discovery: the more heterogeneous the data, the less training is needed when FedPredict is added, making the learning process highly efficient.*

Resumo. *O aprendizado federado (FL) surgiu como uma técnica onde diversos dispositivos (também chamados de clientes) podem aprender de forma colaborativa a partir da orquestração de um servidor central, proporcionando escalabilidade, privacidade e baixo custo de comunicação. A maioria das pesquisas sobre este tema apresenta propostas para a etapa do treinamento de modelos no aprendizado federado, para endereçar diversos problemas como a heterogeneidade estatística de dados, o que muitas vezes representa aumento de custos (e.g., computacional, armazenamento e comunicação). No entanto, recentemente foi proposta a solução FedPredict, um plugin que opera na etapa de predição do aprendizado federado, que quando adicionado pode melhorar significativamente o desempenho de diversas soluções tradicionais em cenários de heterogeneidade de dados, sem requerer qualquer modificação na sua estrutura original ou adição de treinamento. Nesta direção, este trabalho apresenta experimentos sobre uma nova descoberta: quanto mais heterogêneos são os dados, menos treinamento é necessário quando o FedPredict é adicionado, tornando o processo de aprendizado altamente eficiente.*

1. Introdução

O aprendizado federado, do inglês *Federated Learning (FL)*, emergiu como um novo paradigma de aprendizado distribuído, capaz de trazer benefícios significativos em termos

de privacidade, eficiência (e.g., computacional e comunicação), além de ser escalável em termos da quantidade de dispositivos/clientes envolvidos. Estes benefícios são explicados pelo fato de no aprendizado federado, apenas os parâmetros de modelo de aprendizado de máquina de cada cliente serem compartilhados com o servidor central, ao invés de dados brutos. Comparativamente, estes parâmetros representam uma quantidade menor de *bytes* transmitidos na rede, além de não conterem informações privadas dos clientes.

No aprendizado federado, o treinamento dos vários dispositivos é dividido em rodadas de comunicação, onde o servidor compartilha o modelo global com um conjunto de clientes selecionados para treinamento. Em cada rodada, o fluxo de execução é o seguinte:

1. Treinamento
   (a) O cliente recebe o modelo global e realiza o treinamento com os seus dados locais. Os parâmetros atualizados são, então, enviados para o servidor, juntamente com as métricas de desempenho obtidas sobre os dados de treino e teste.
   (b) O servidor agrega os parâmetros recebidos pelos clientes para gerar o modelo global atualizado no fim da rodada de treinamento. Esta agregação é comumente uma média dos parâmetros ponderada pela quantidade de dados em que cada um foi treinado.
2. Validação/predição: ao fim de cada rodada, após a atualização do modelo global, o servidor pode enviar os novos parâmetros para um conjunto de clientes, para que estes realizem a validação do modelo atualizado sob os dados locais. Os resultados são, então, enviados ao servidor.

Um dos problemas mais comuns do aprendizado federado é a heterogeneidade estatística dos dados dos clientes, que afeta o desempenho do sistema. No processo de agregação, as atualizações do servidor avançam em direção à média do ótimo do cliente. Em um cenário de dados IID (independente e identicamente distribuídos), o modelo médio é próximo do ótimo global, uma vez que é equidistante dos ótimos locais do clientes. Porém, em cenários de dados não-IID, o ótimo global não é equidistante do ótimo local, gerando um baixo desempenho [Tan et al. 2022a, Capanema et al. 2023b]. Um exemplo de aplicação onde isso ocorre é no problema de detecção de atividades humanas (do inglês *human activity recognition - HAR*), onde cada indivíduo pode ter um padrão específico de atividades fazendo com que os dados coletados pelos sensores de mobilidade contenham dados estatisticamente diferentes dos demais clientes. Assim, o modelo global médio se torna menos eficaz para os cenários dos clientes.

Como alternativa, surgiram as soluções de aprendizado federado personalizado (do inglês *personalized federated learning - PFL*), que buscam endereçar o problema de dados não IID através do uso de diferentes tipos de técnicas como: modelo privado local, regularização personalizada, destilação do conhecimento, seleção de clientes [Tan et al. 2022a], dentre outras técnicas. Em comum, estas soluções operam na etapa de treinamento do aprendizado federado, e representam, em geral, alguma limitação, como o aumento de custos (e.g., processamento, comunicação e energético), aumento do tempo de rodada, ou baixo desempenho de novos clientes no *FL*.

Considerando estes aspectos, foi proposta a solução *FedPredict* [Capanema et al. 2023b, Capanema et al. 2024], o primeiro *plugin* de aprendizado

federado, e que cria uma nova direção de pesquisa na área. A ideia principal do *FedPredict* é que, os modelos locais dos clientes possam ser altamente assertivos em dados não-IID sem que seja necessário aplicar mudanças na etapa de treinamento (i.e., alta modularidade) e sem adição de custo computacional significativo (as operações adicionais tem custo linear). Além disso, o *FedPredict* utiliza os recursos já presentes na maioria das soluções de aprendizado federado, como os modelos global e local. Isto é feito a partir da combinação, na etapa de predição, dos modelos global (i.e., generalista) e local (i.e., personalizado) considerando diversos fatores, como o nível de evolução do modelo global e o nível de atualização do modelo local. O modelo combinado tem, portanto, um alto desempenho em dados não-IID, e se mostra mais robusto do que a utilização separada de cada um.

Em [Capanema et al. 2023b], o *FedPredict* foi combinado com a solução *FedAvg* [McMahan et al. 2017] e comparado com importantes soluções de *PFL* que utilizam modelos locais privados e/ou protótipos de classes como técnicas de personalização. No entanto, outra classe de soluções importantes que devem ser avaliadas são as técnicas de seleção de clientes. Estas técnicas são cruciais para a implantação do aprendizado federado em ambiente real, uma vez que trazem benefícios como convergência mais rápida do modelo global, economia de recursos dos dispositivos envolvidos bem como a viabilização do treinamento de clientes com menor capacidade [Fu et al. 2023].

Dessa forma, no presente trabalho a solução *FedPredict* é avaliada sob o contexto de técnicas de seleção de clientes. Neste sentido, o objetivo principal é medir o aumento de eficiência proporcionado pelo uso do *plugin* adicionado a essas técnicas. As contribuições deste trabalho são compostas por experimentos que revelam as seguintes descobertas sobre o *FedPredict*: *(i)* Alta eficiência: o *FedPredict* permite que sejam selecionados grupos ainda mais restritos de clientes (e.g., os melhores, ou seja, os que possuem maior capacidade de computação, bateria e comunicação) sem que exista um viés que degrade o desempenho do sistema. Ou seja, menos treinamento é necessário ao mesmo tempo em que a assertividade dos clientes aumenta; e *(ii)* A eficiência depende da heterogeneidade dos dados: quanto mais heterogêneos são os dados dos clientes, maior é a eficiência alcançada pelo *FedPredict*. Em outras palavras, este trabalho demonstra que o uso do *plugin* permite que se treine com menos clientes por rodada, tornando o aprendizado mais eficiente.

O restante do trabalho é organizado da seguinte maneira: a Seção 2 contém os trabalhos relacionados; a Seção 3 contém uma visão geral do aprendizado federado; na Seção 4 são apresentadas as principais características da solução *FedPredict*; nas Seções 5 e 6 são apresentados os resultados e a conclusão, respectivamente.

2. Trabalhos Relacionados

Nesta seção são apresentados os principais trabalhos da literatura referentes ao aprendizado federado personalizado e às técnicas de seleção de clientes.

2.1. Aprendizado federado personalizado

O surgimento do aprendizado federado personalizado *PFL* se deve ao fato do baixo desempenho do modelo global da solução tradicional FedAvg [McMahan et al. 2017] em dados não-IID. Este problema é explicado pelo processo de treinamento de modelos, onde

os parâmetros do modelo local do cliente são atualizados em direção ao ótimo dos dados locais. Isto é efetivo quando os dados dos clientes são semelhantes entre si, de modo que o modelo global será atualizado na direção ótima equidistante de todos os clientes. Este cenário, no entanto, não pode ser garantido e, na verdade, os dados coletados pelos clientes são comumente heterogêneos, impactando severamente no desempenho do sistema de aprendizado federado tradicional.

Alternativamente, soluções como FedAvgM e FedYogi [Hsu et al. 2019, Reddi et al. 2020] propõem modificações no cálculo do modelo global agregado, de modo que o sistema sofra menos com variações abruptas dos parâmetros recebidos dos clientes. No entanto, estas soluções representam melhorias limitadas no cenário não-IID, uma vez que ainda são muito semelhantes ao FedAvg.

Com o objetivo de intensificar a personalização para melhorar o desempenho em dados não-IID, pesquisadores projetam soluções onde parte do modelo local é privado do cliente, evitando assim que o conhecimento obtido no cenário específico local seja sobreescrito pelo modelo global na etapa de treinamento. FedPer [Arivazhagan et al. 2019] faz o emprego da abordagem "camadas base + camadas personalizadas", onde os parâmetros das camadas base são compartilhados com o servidor para aprender representações de baixo nível. Por outro lado, os parâmetros das camadas personalizadas são mantidos localmente para aprender representações específicas e endereçar melhor a distribuição dos dados locais.

Além do FedPer, diversas soluções mantêm um modelo personalizado localmente, como FedProto [Tan et al. 2022b], FedClassAvg [Jang et al. 2022] e FedKD [Wu et al. 2022]. Porém, um problema comum entre essas soluções é a queda de desempenho quando novos clientes são adicionados ao processo de FL devido ao treinamento local necessário.

Dentro do contexto de soluções de *PFL* , o *FedPredict* [Capanema et al. 2023b] surgiu como o primeiro *plugin* de personalização no aprendizado federado. Isto é feito a partir da combinação de parâmetros do modelo global e local, no lado do cliente. Esta técnica tem sua ideia originária nos modelos *ensemble* e também se assemelha às técnicas onde são combinadas as saídas das camadas de redes neurais [Capanema et al. 2021, Capanema et al. 2023a, Capanema et al. 2020]. No caso específico do *plugin*, no entanto, o objetivo é combinar parâmetros de modelos. Assim, ele é capaz de proporcionar que outras técnicas de *FL* tenham alto desempenho em cenários de dados não-IID. Além disso, o *FedPredict* oferece suporte aos novos clientes no aprendizado federado, onde o uso do *plugin* evita que o desempenho seja baixo devido ao modelo local ainda não ter sido treinado.

2.2. Seleção de clientes

Dentro do aprendizado federado personalizado, existem técnicas que buscam selecionar um conjunto de clientes mais adequados para realizar o treinamento, a cada rodada, considerando diferentes métricas e restrições. O uso dessas técnicas apresenta diversos benefícios como aceleração da convergência do modelo global, viabilização da participação de dispositivos com menor capacidade ou com restrições, aprimoramento do *fairness* do aprendizado, dentre outros aspectos [Fu et al. 2023]. Esses benefícios não podem ser explorados pela seleção randômica, presente na proposta original do *FedAvg*, onde clientes

são selecionados aleatoriamente, sem qualquer restrição.

Em particular, características dos clientes e da rede envolvida como disponibilidade energética, capacidade de processamento e de comunicação são considerados no processo de seleção de clientes, o que é crucial para a implementação do aprendizado federado em ambiente real.

A solução *Power-of-Choice* (*POC*) [Cho et al. 2022] acelera a convergência do modelo a partir da seleção para treinamento dos clientes com maior *loss* nos dados de treino. No entanto, ela demanda que a cada rodada, um conjunto considerado de clientes seja avaliado, retornando a informação da *loss* gerada pelo modelo global aplicado nos dados locais, adicionando custo de processamento, energético, de comunicação e aumentando o tempo de rodada.

O método *Resource Aware Client Selection* (*RAWCS*) [Maciel et al. 2023] considera que três aspectos devem ser satisfeitos para que um dado cliente esteja apto para o treinamento: nível esperado de bateria após o treinamento, sujeito a um valor mínimo; tempo de processamento, sujeito a um valor máximo; qualidade de sinal, sujeita a um valor mínimo. Além disso, são atribuídas importâncias a cada um desses fatores, e o somatório alcançado é denotado como a utilidade do cliente. O objetivo, então, se torna alcançar o máximo de utilidade considerando os clientes selecionados.

O *FedPredict*, porém, nunca foi avaliado sob diferentes técnicas de seleção de clientes, incluindo *POC* e *RAWCS* . Este estudo é importante, dentre outros motivos, para verificar se o uso dessas técnicas combinado com o *plugin* pode tornar o aprendizado federado ainda mais eficiente.

3. Visão geral sobre aprendizado federado

Nesta seção, é apresentada uma visão geral sobre o aprendizado federado padrão, através da solução *FedAvg* [McMahan et al. 2017]. No contexto do FL tradicional, seja K o número de clientes e T o número de rodadas. A cada rodada $t = 1, 2, \ldots, T$, o servidor seleciona um subconjunto randômico de clientes $C \leq K$ e envia a eles os parâmetros atuais do modelo global w_t, como parte do treinamento. Os parâmetros do modelo global recebidos pelo cliente sobrescrevem os parâmetros do modelo local. Em seguida, é realizado o treinamento utilizando os dados locais e os parâmetros atualizados w_{t+1}^k são enviados para o servidor juntamente com as métricas de desempenho obtidas. Note que, nesta etapa, o cliente pode enviar métricas do modelo em relação aos dados de treino e de teste/validação. Ao final de cada rodada de treinamento, o servidor agrega os parâmetros recebidos dos clientes para atualizar o modelo global conforme apresentado a seguir [McMahan et al. 2017]:

$$w_{t+1} = \sum_{k \in C} \frac{n_t^k}{n_t} w_{t+1}^k, \tag{1}$$

onde w_{t+1} é o parâmetro do modelo global atualizado, w_{t+1}^k é o parâmetro atualizado do cliente k.

Opcionalmente, o servidor pode enviar os parâmetros atualizados do modelo global para um conjunto de clientes realizar a validação/teste sobre os seus dados locais.

Esta etapa é importante, pois, a combinação de parâmetros feita pelo *FedPredict* aumenta significativamente o desempenho dos clientes.

Por fim, no FL tradicional, o FedAvg visa minimizar a seguinte função [McMahan et al. 2017, Jang et al. 2022]:

$$\min f(w) \text{ onde } f(w) = \sum_{k=1}^{K} \frac{n_t^k}{n_t} \mathcal{L}(x^k, y^k; w) \tag{2}$$

onde w é o modelo global, f é o modelo de rede neural, $\mathcal{L}$ denota a função de perda, x^k são os dados de entrada e y^k é o respectivo rótulo para o conjunto de dados do cliente k.

4. FedPredict

Nesta seção, a solução *FedPredict* (*Federated Prediction*) [Capanema et al. 2023b, Capanema et al. 2024] é definida, incluindo as suas principais fórmulas e notações utilizadas. O *plugin* está disponível através de um pacote *Python*[1].

Para permitir que o cliente tenha um modelo robusto, o *FedPredict* utiliza o princípio da combinação de parâmetros para gerar um modelo generalista e, ao mesmo tempo, personalizado a partir dos parâmetros dos modelos global e local. Um ponto importante neste processo é compreender quão significativo são estes modelos ao longo do processo de aprendizado.

Os autores da solução *FedPredict* definem duas variáveis importantes: *update level* (ul) que denota o nível de evolução do modelo global; *evolution level* (el) que indica o nível de atualização do modelo local.

A variável el é definida a seguir:

$$el \leftarrow \frac{t}{T}, \text{ com } T > 0 \tag{3}$$

sendo t número da rodada atual e T o número máximo de rodadas. Note também que $el \in (0, 1]$. Quanto maior o valor de t maior o nível de evolução do modelo global (i.e., aproximação da convergência).

A variável ul é denotada pela Equação 4:

$$ul \leftarrow \frac{1}{nt}, \text{ com } nt > 0, \tag{4}$$

onde nt denota a quantidade de rodadas desde a última vez que o cliente treinou. Quanto maior o valor de nt menor o nível de atualização do modelo local.

Estas duas variáveis são utilizadas para definir a quantidade de peso que será dado para cada um dos parâmetros global e local no processo de combinação. Neste sentido, é importante observar que, o modelo global tem um significativo avanço do seu aprendizado nas rodadas iniciais, enquanto nas demais a evolução nas curvas de acurácia e *loss* se tornam mais discretas de uma rodada para a outra. Considerando isto, *FedPredict* atribui maior peso ao modelo global nas primeiras rodadas e decresce o seu valor ao longo do

[1] nhttps://github.com/claudiocapanema/fedpredict. Acessado em 02/04/2024

treinamento. Ao mesmo tempo, do ponto de vista do modelo local, se este foi treinado recentemente (i.e., baixo valor de nt) o seu nível de atualização é alto, e, portanto, é necessário atribuir um peso significativo ao mesmo.

O valor do peso do modelo global gw é definido pela seguinte equação de decaimento exponencial:

$$gw \leftarrow e^{(-ul-el)}, \tag{5}$$

onde $(-ul-el) \in [-2, 0]$ e, portanto, $gw \in [0.13, 1]$. Dessa forma, garante-se que sempre será atribuída uma quantidade de peso significativa ao modelo global.

Após esta etapa, é calculado o peso do modelo local lw, como definido a seguir:

$$lw \leftarrow 1 - gw. \tag{6}$$

Por fim, os parâmetros dos modelos global e local são ponderados por gw e lw, respectivamente, e somados [Capanema et al. 2023b].

5. Resultados

Nesta seção, o *FedPredict* é avaliado quando adicionado a três técnicas de seleção de clientes: Randômico, *POC* e *RAWCS*. Como o *FedPredict* é um *plugin* para ser usado no topo de uma solução, o *FedAvg* é utilizado como a solução base necessária. Assim, é avaliado o seu desempenho com e sem o *plugin*.

5.1. Configuração dos experimentos

A Tabela 1 exibe a configuração padrão dos experimentos. Os *datasets* utilizados são *CIFAR-10* [Krizhevsky et al. 2009], que contém 10 classes de imagens de diferentes temas, e *GTSRB* [Stallkamp et al. 2011], que contém 43 classes de imagens de placas de trânsito. Ambos os *datasets* são amplamente adotados na literatura e contêm diferentes quantidades de classes O grau prático não-IID dos dados é variado usando o método [Lin et al. 2020], onde a distribuição de *Dirichlet* separa os dados em conjuntos de treinamento disjuntos para os clientes considerando valores de α 0,1 e 1,0, a quantidade total de rodadas é $T = 100$, o modelo de rede neural utilizado contém duas camadas convolucionais e uma *fully connected*, uma época de treinamento é executada a cada participação do cliente, e o total de clientes é $K = 20$. A quantidade de clientes selecionados C em cada rodada de treinamento varia em três níveis: alto, médio e baixo.

Tabela 1. Configuração padrão

Datasets	**α**	**T**	**Modelo**	**Épocas**	**K**	**C**
CIFAR-10 e *GTSRB*	0,1-1,0	100	2 camadas convolucionais + 1 *fully connected*	1	20	Separado em níveis alto, médio e baixo de acordo com o funcionamento de cada solução

Os métodos Randômico e *POC* selecionam uma quantidade constante de clientes por rodada, denotada pelo valor de C. A técnica *RAWCS*, por outro lado, varia esta quantidade a cada rodada, de acordo com os recursos de cada cliente e da rede de comunicação. *RAWCS* utiliza três hiperparâmetros: (1) ql, o valor mínimo da qualidade do *link* de comunicação; (2) l, a latência máxima; (3) b, o nível mínimo de bateria, em joules, que o cliente deve ter para participar do treinamento. A Tabela 2 apresenta as descrições de cada nível de seleção de clientes de acordo com cada tipo de solução.

Tabela 2. Níveis de C

Nível de seleção	Solução	Descrição
Alto	Randômico e *POC* *RAWCS*	$C = 14$ (70% de $K = 20$) $ql = 0,01$, $l = 7$ e $b = 0,05$
Médio	Randômico e *POC* *RAWCS*	$C = 10$ (50% de $K = 20$) $ql = 0,05$, $l = 0,4$ e $b = 0,35$
Baixo	Randômico e *POC* *RAWCS*	$C = 6$ (30% de $K = 20$) $ql = 0,5$, $l = 0,5$ e $b = 0,45$

As métricas utilizadas nos experimentos são acurácia e eficiência, sendo esta última calculada através da relação acurácia e quantidade de clientes selecionados para treinamento, como apresentado a seguir:

$$\text{eficiência} = \frac{\text{acurácia}}{C_t}, \tag{7}$$

onde C_t é a quantidade de clientes selecionados para treinamento na rodada t. Note que a eficiência não indica que o crescimento de C_t corresponde a um incremento proporcional na acurácia.

5.2. Resultados

A Figura 1 e a Tabela 3 exibem os resultados de acurácia e a Figura 2 e a Tabela 4 exibem os resultados da eficiência das soluções para o *dataset CIFAR-10*. Para o *dataset GTSRB*, os resultados de acurácia são representados pela Figura 3 e pela Tabela 5, e os valores da métrica de eficiência são representados pela Figura 4 e pela Tabela 6. Os resultados mostram que, quanto menor o valor de α maior é o ganho de acurácia obtido pela adição do *plugin* à solução original. Para o *dataset* os valores de ganho são de até 98.7%, 110,5% e 139.3% para os níveis de seleção de clientes alto, médio e baixo respectivamente. Analogamente, os ganhos de acurácia para o *dataset GTSRB* são de até 48.1%, 51,5% e 58.7%.

Em particular, sobre os diferentes níveis de seleção de clientes, as curvas do *FedPredict* estão sempre mais próximas do que as curvas da solução original *FedAvg*. Isto indica que, com uma quantidade menor de treinamento (e.g., nível baixo) o *plugin* obtém desempenho similar a quando se treina com muitos clientes (i.e., nível alto). Este resultado é traduzido pela métrica de eficiência, expressa pelas Figuras 2 e 4, e pelas Tabelas 4 e 6. As curvas dos gráficos indicam que o maior nível de eficiência é alcançado quando o nível de seleção de clientes é baixo, seguido pelo nível médio e, posteriormente, pelo nível alto. Em particular, para as técnicas Randômico e *POC*, a eficiência do *FedPredict* no nível alto é superior até do nível baixo do *FedAvg* para o *dataset CIFAR-10* e $\alpha = 0,1$.

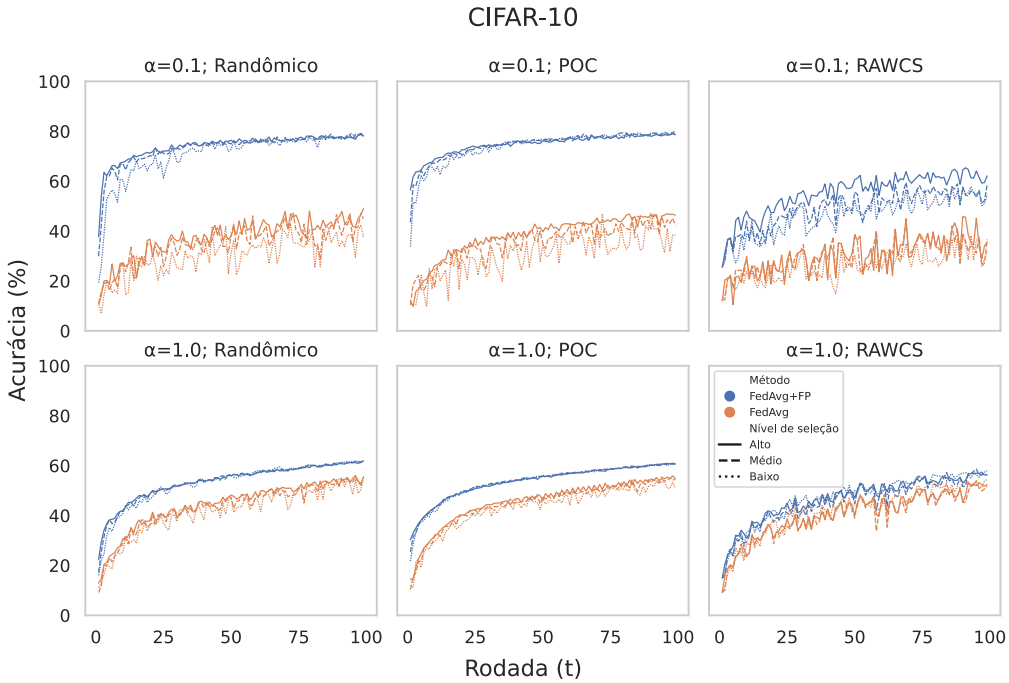

Figura 1. Acurácia das soluções no *dataset CIFAR-10*

Tabela 3. Acurácia média das soluções para o *dataset CIFAR-10*

Nível de seleção	Tipo de seleção	Método	α 0,1 Acurácia	α 0,1 Ganho	α 1,0 Acurácia	α 1,0 Ganho
Alto	Randômico	FedAvg+FP	**73,9±1,1**	**↑98,7%**	**53,5±1,5**	**↑20,8%**
		FedAvg	37,2±1,6	-	44,3±1,9	-
	POC	FedAvg+FP	**74,7±0,8**	**↑96,1%**	**53,5±1,3**	**↑17,6%**
		FedAvg	38,1±1,7	-	45,5±1,8	-
	RAWCS	FedAvg+FP	**55,2±1,6**	**↑82,2%**	**46,8±1,8**	**↑15,3%**
		FedAvg	30,3±1,4	-	40,6±1,8	-
Médio	Randômico	FedAvg+FP	**73,2±1,4**	**↑109,1%**	**53,4±1,6**	**↑23,3%**
		FedAvg	35,0±1,5	-	43,3±1,9	-
	POC	FedAvg+FP	**74,5±1,1**	**↑110,5%**	**53,6±1,4**	**↑20,2%**
		FedAvg	35,4±1,6	-	44,6±1,8	-
	RAWCS	FedAvg+FP	**49,0±1,5**	**↑65,0%**	**45,6±1,8**	**↑13,7%**
		FedAvg	29,7±1,3	-	40,1±1,9	-
Baixo	Randômico	FedAvg+FP	**70,7±2,0**	**↑133,3%**	**52,9±1,8**	**↑26,9%**
		FedAvg	30,3±1,5	-	41,7±2,0	-
	POC	FedAvg+FP	**73,7±1,4**	**↑139,3%**	**53,3±1,5**	**↑23,7%**
		FedAvg	30,8±1,7	-	43,1±1,8	-
	RAWCS	FedAvg+FP	**46,4±1,5**	**↑77,8%**	**46,9±1,9**	**↑12,5%**
		FedAvg	26,1±1,2	-	41,7±2,0	-

Os ganhos de eficiência alcançados (veja as Tabelas 4 e 6) são de até 120%, 130,3% e 151% para o *dataset CIFAR-10*, considerando respectivamente os níveis de

seleção alto, médio e baixo. Na mesma ordem, os resultados de ganho de eficiência para o *dataset GTSRB* são 48,%, 51,5% e 58,7%.

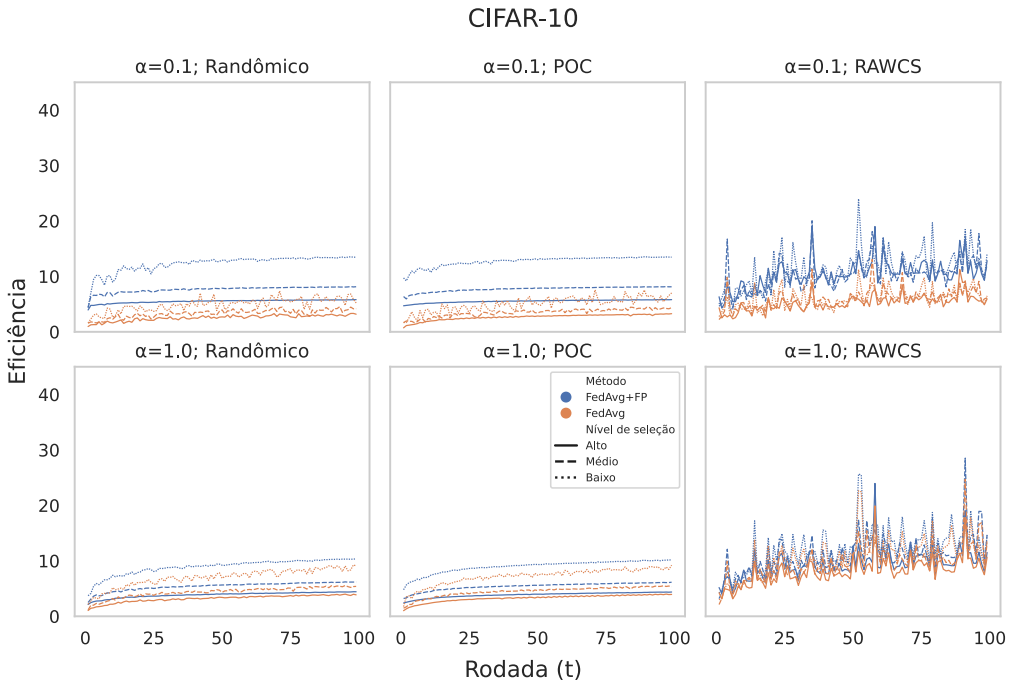

Figura 2. Eficiência das soluções no *dataset CIFAR-10*

Tabela 4. Eficiência média das soluções para o *dataset CIFAR-10*

Nível de seleção	Tipo de seleção	Método	α 0,1 Eficiência	α 0,1 Ganho	α 1,0 Eficiência	α 1,0 Ganho
Alto	Randômico	FedAvg+FP	**5,5±0,1**	**↑120,0%**	**3,9±0,1**	**↑21,9%**
		FedAvg	2,5±0,1	-	3,2±0,1	-
	POC	FedAvg+FP	**5,5±0,1**	**↑103,7%**	**3,9±0,1**	**↑18,2%**
		FedAvg	2,7±0,1	-	3,3±0,1	-
	RAWCS	FedAvg+FP	**10,4±0,5**	**↑100,0%**	**9,4±0,5**	**↑16,0%**
		FedAvg	5,2±0,3	-	8,1±0,5	-
Médio	Randômico	FedAvg+FP	**7,6±0,1**	**↑130,3%**	**5,4±0,1**	**↑22,7%**
		FedAvg	3,3±0,1	-	4,4±0,2	-
	POC	FedAvg+FP	**7,7±0,1**	**↑120,0%**	**5,4±0,1**	**↑20,0%**
		FedAvg	3,5±0,1	-	4,5±0,2	-
	RAWCS	FedAvg+FP	**10,3±0,6**	**↑71,7%**	**11,0±0,7**	**↑14,6%**
		FedAvg	6,0±0,3	-	9,6±0,6	-
Baixo	Randômico	FedAvg+FP	**12,3±0,3**	**↑151,0%**	**8,9±0,3**	**↑27,1%**
		FedAvg	4,9±0,2	-	7,0±0,3	-
	POC	FedAvg+FP	**12,7±0,2**	**↑154,0%**	**9,0±0,2**	**↑25,0%**
		FedAvg	5,0±0,3	-	7,2±0,3	-
	RAWCS	FedAvg+FP	**11,3±0,7**	**↑88,3%**	**12,2±0,8**	**↑13,0%**
		FedAvg	6,0±0,4	-	**10,8±0,7**	-

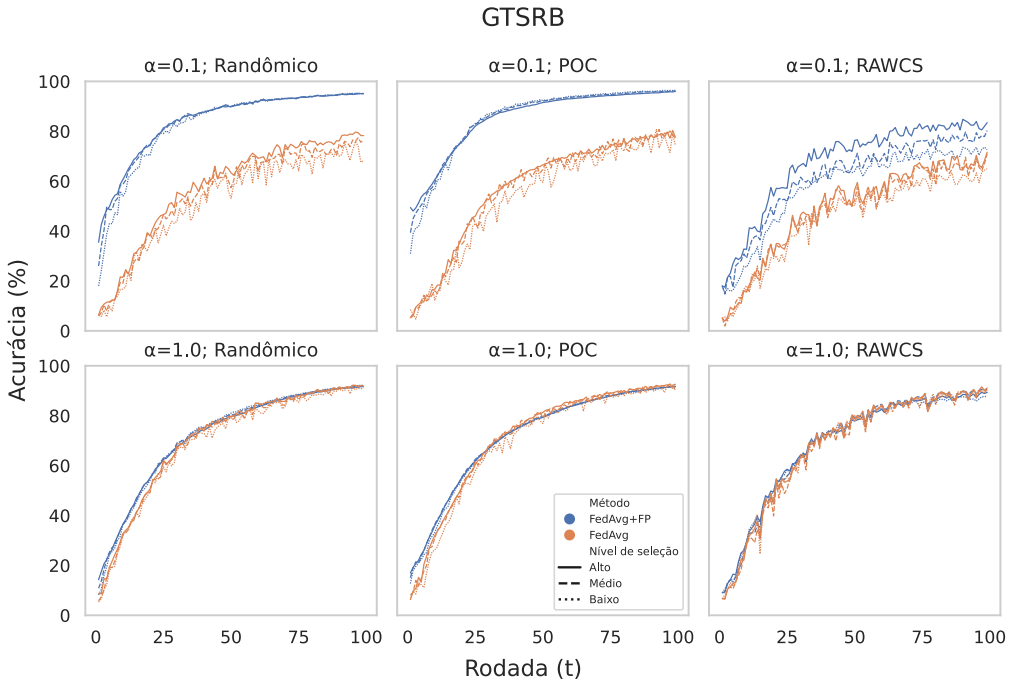

Figura 3. Acurácia das soluções no *dataset GTSRB*

Tabela 5. Acurácia média das soluções para o *dataset GTSRB*

Nível de seleção	Tipo de seleção	Método	α 0,1 Acurácia	0,1 Ganho	1,0 Acurácia	1,0 Ganho
Alto	Randômico	FedAvg+FP	**84,2±2,8**	**↑45,7%**	**71,9±4,1**	**↑1,7%**
		FedAvg	57,8±4,1	-	**70,7±4,5**	-
	POC	FedAvg+FP	**85,3±2,6**	**↑48,1%**	**71,8±4,1**	**↑0,7%**
		FedAvg	57,6±4,3	-	**71,3±4,7**	-
	RAWCS	FedAvg+FP	**67,6±3,6**	**↑38,0%**	**68,7±4,5**	**↑0,6%**
		FedAvg	49,0±3,7	-	**68,3±4,7**	-
Médio	Randômico	FedAvg+FP	**83,8±3,0**	**↑51,5%**	**72,2±4,2**	**↑2,8%**
		FedAvg	55,3±4,0	-	**70,2±4,6**	-
	POC	FedAvg+FP	**85,6±2,8**	**↑50,7%**	**72,0±4,1**	**↑1,4%**
		FedAvg	56,8±4,4	-	**71,0±4,7**	-
	RAWCS	FedAvg+FP	**61,7±3,4**	**↑28,5%**	**68,9±4,7**	**↑2,2%**
		FedAvg	48,0±3,6	-	**67,4±4,8**	-
Baixo	Randômico	FedAvg+FP	**82,5±3,4**	**↑58,0%**	**72,1±4,3**	**↑4,3%**
		FedAvg	52,2±3,9	-	**69,1±4,6**	-
	POC	FedAvg+FP	**85,2±3,1**	**↑58,7%**	**71,4±4,2**	**↑3,3%**
		FedAvg	53,7±4,3	-	**69,1±4,8**	-
	RAWCS	FedAvg+FP	**57,7±3,4**	**↑27,9%**	**68,4±4,4**	**↑0,1%**
		FedAvg	45,1±3,4	-	**68,3±4,7**	-

É importante notar que os métodos Randômico e *POC* não são avaliados dentro do cenário de dispositivos heterogêneos com restrições de capacidade, diferentemente do *RAWCS*. Dessa forma, *RAWCS* seleciona uma quantidade menor de clientes por rodada, o

que justifica a sua menor acurácia ao mesmo tempo que possui picos de maior eficiência. Em particular, *RAWCS* seleciona em média $C = 4,4$ clientes por rodada (i.e., 22% de $K = 20$).

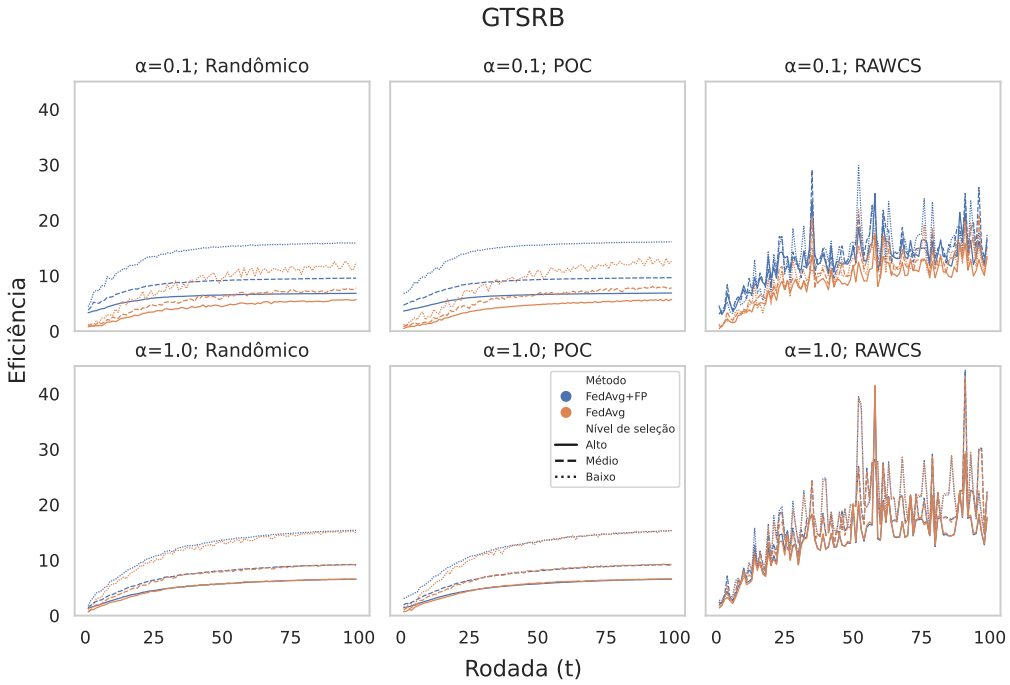

Figura 4. Eficiência das soluções no *dataset* *GTSRB*

Tabela 6. Eficiência média das soluções para o *dataset* *GTSRB*

Nível de seleção	Tipo de seleção	Método	α 0,1 Eficiência	0,1 Ganho	1,0 Eficiência	1,0 Ganho
Alto	Randômico	FedAvg+FP	**6,1±0,2**	**↑45,2%**	**5,2±0,3**	**↑2,0%**
		FedAvg	4,2±0,3	-	**5,1±0,3**	-
	POC	FedAvg+FP	**6,2±0,2**	**↑47,6%**	**5,2±0,3**	**↑0,0%**
		FedAvg	4,2±0,3	-	**5,2±0,3**	-
	RAWCS	FedAvg+FP	**12,7±0,8**	**↑41,1%**	**13,8±1,1**	**↑0,7%**
		FedAvg	9,0±0,7	-	**13,7±1,1**	-
Médio	Randômico	FedAvg+FP	**8,6±0,3**	**↑53,6%**	**7,3±0,4**	**↑2,8%**
		FedAvg	5,6±0,4	-	**7,1±0,4**	-
	POC	FedAvg+FP	**8,7±0,3**	**↑50,0%**	**7,3±0,4**	**↑1,4%**
		FedAvg	5,8±0,4	-	**7,2±0,5**	-
	RAWCS	FedAvg+FP	**13,4±0,9**	**↑28,8%**	**16,5±1,4**	**↑2,5%**
		FedAvg	10,4±0,9	-	**16,1±1,4**	-
Baixo	Randômico	FedAvg+FP	**14,0±0,5**	**↑59,1%**	**12,2±0,7**	**↑4,3%**
		FedAvg	8,8±0,6	-	**11,7±0,7**	-
	POC	FedAvg+FP	**14,4±0,5**	**↑60,0%**	**12,1±0,7**	**↑3,4%**
		FedAvg	9,0±0,7	-	**11,7±0,8**	-
	RAWCS	FedAvg+FP	**13,6±1,0**	**↑28,3%**	**17,9±1,4**	**↑0,6%**
		FedAvg	10,6±0,9	-	**17,8±1,5**	-

Os elevados ganhos de eficiência proporcionados pelo *plugin* mostram que a ideia de se transferir o conhecimento atualizado do modelo global para o modelo local na etapa de predição para se evitar a necessidade de treinamento frequente é uma nova e promissora direção de pesquisa. No primeiro passo dado pela solução *FedPredict* esta transferência de conhecimento foi feita a partir da combinação de parâmetros, no entanto, outras possibilidades podem ser investigadas.

6. Conclusão

Neste trabalho, o *FedPredict* é avaliado sob a perspectiva de diferentes técnicas de seleção de clientes no aprendizado federado. Os resultados mostram que, dentre os métodos avaliados, o *FedPredict* não só obtém ganhos significativos de acurácia, mas de eficiência de treinamento. Em particular, o *plugin* permite que a solução original tenha maior assertividade/acurácia mesmo que menos clientes sejam selecionados para treinamento a cada rodada. Este benefício tem impacto sobre a maioria das soluções de aprendizado federado, uma vez que o *plugin* tem baixo custo computacional e pode ser combinado com diferentes técnicas.

Como trabalhos futuros, pretende-se estender a avaliação considerando outras técnicas de seleção de clientes (e.g., que consideram a qualidade dos dados locais dos clientes como fator de seleção), e considerar o uso de outros tipos de redes neurais e *datasets* que envolvam tarefas de regressão e outros tipos de dados (e.g., séries temporais).

Agradecimentos

Este projeto foi apoiado pelo programa PPI Softex, Acordo de Parceria nº 126/2022, financiado pelo Ministério da Ciência, Tecnologia e Inovações com recursos da Lei nº 8.248, de 23 de outubro de 1991 [01245.013778/2020-21]. Os autores agradecem às agências de pesquisa CAPES, CNPq, FAPEMIG e bolsas 15/24494-8 & 18/23064-8, Fundação de Amparo à Pesquisa de São Paulo (FAPESP).

Referências

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. (2019). Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Capanema, C. G., de Oliveira, G. S., Silva, F. A., Silva, T. R., and Loureiro, A. A. (2023a). Combining recurrent and graph neural networks to predict the next place's category. *Ad Hoc Networks*, 138:103016.

Capanema, C. G., de Souza, A. M., Silva, F. A., Villas, L. A., and Loureiro, A. A. (2023b). Fedpredict: Combining global and local parameters in the prediction step of federated learning. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 17–24. IEEE.

Capanema, C. G., de Souza, Joahannes B D da Costa, F. A., Villas, L. A., and Loureiro, A. A. L. (2024). A modular plugin for concept drift in federated learning. In *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE.

Capanema, C. G., Silva, F. A., Silva, T. R., and Loureiro, A. A. (2021). Poi-rgnn: Using recurrent and graph neural networks to predict the category of the next point of interest.

In *Proceedings of the 18th acm symposium on performance evaluation of wireless ad hoc, sensor, & ubiquitous networks*, pages 49–56.

Capanema, C. G. S., Silva, F. A., and Silva, T. R. d. M. B. (2020). Mfa-rnn: Uma rede neural recorrente para predição de próximo local de visita com base em dados esparsos. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuidos*, pages 127–140. SBC.

Cho, Y. J., Wang, J., and Joshi, G. (2022). Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR.

Fu, L., Zhang, H., Gao, G., Zhang, M., and Liu, X. (2023). Client selection in federated learning: Principles, challenges, and opportunities. *IEEE Internet of Things Journal*.

Hsu, T.-M. H., Qi, H., and Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Jang, J., Ha, H., Jung, D., and Yoon, S. (2022). Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proceedings of the 51st International Conference on Parallel Processing*, pages 1–10.

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

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363.

Maciel, F., De Souza, A. M., Bittencourt, L. F., and Villas, L. A. (2023). Resource aware client selection for federated learning in iot scenarios. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 1–8. IEEE.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. (2022a). Towards personalized federated learning. *IEEE Trans. on Neural Networks and Learning Systems*.

Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., and Zhang, C. (2022b). Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence*, volume 1, page 3.

Wu, C., Wu, F., Lyu, L., Huang, Y., and Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032.