

Seleção do Modelo Federado Baseada em Previsão de Desempenho e Tempo de Comunicação

Luana Gantert e Miguel Elias M. Campista

Resumo— O aprendizado federado assume que clientes e servidores se conhecem previamente e que o processo de seleção é apenas relativo aos clientes. Em uma direção diferente, este trabalho propõe a seleção de servidores a partir de estimativas futuras de desempenho dos modelos em treinamento. Cada cliente inicia dois procedimentos de treinamento em paralelo com dois servidores com dois modelos distintos e calcula, através de regressão linear, qual deles deve ser mantido. A proposta compara o desempenho dos servidores a partir da métrica AUC-ROC em uma dada rodada r e, caso a diferença seja maior ou igual a $\beta\%$, o cliente mantém o treinamento apenas com o de melhor desempenho. A proposta também considera a latência da comunicação cliente-servidor como critério de decisão, simulando a latência com valores provenientes de aplicações de monitoramento industrial. Os resultados demonstram que ao variar o parâmetro β , o melhor cenário analisado apresenta redução de aproximadamente 10,4% da latência acumulada no treinamento com maior valor de r e um aumento de 12,5% na métrica AUC-ROC ao considerar o cliente com aumento de desempenho mais expressivo.

Palavras-Chave— Aprendizado Federado, Internet das Coisas, Cidades Inteligentes.

Abstract— The federated learning assumes that the clients and servers know each other beforehand and that the selection process only concerns the clients. In a different direction, this paper proposes the server selection by future performance predictions of the models in training. Each client initiates two training procedures in parallel with two servers with two different models and calculates, using linear regression, which should be retained. The proposal compares the performance of the servers using the AUC-ROC metric in a given round r . If the difference is greater than or equal to $\beta\%$, the client continues the training in the server with the best performance. The proposal also considers client-server communication latency a decision criterion, simulating latency with values from industrial monitoring applications. The results show that the β parameter variation in the best case analyzed shows a reduction of approximately 10,4% in the accumulated latency in training with higher r value, and an increase of 12,5% in the AUC-ROC metric, based on the client with the most significant performance improvement.

Keywords— Federated learning, Internet of Things, Smart cities.

I. INTRODUÇÃO

As tecnologias baseadas em aprendizado de máquina são hoje implementadas em diversos setores de produção, desde ambientes IoT (*Internet of Things*) até cenários automotivos e a própria Indústria 4.0 [1, 2]. Em particular nas aplicações industriais, as soluções baseadas em aprendizado de máquina

destacam-se no monitoramento dos processos fabris e logísticos e controle do funcionamento do maquinário, possibilitando redução do gasto energético e planejamento da utilização de recursos computacionais [3, 4]. Nesse contexto, o aprendizado federado [5] apresenta-se como uma alternativa ao aprendizado centralizado ao possibilitar o treinamento colaborativo sem expor dados sensíveis de seus participantes. Os dispositivos/clientes treinam o modelo global fornecido pelo servidor com seus dados locais e enviam os pesos obtidos ao servidor para que estes sejam agregados e um novo modelo global seja gerado. O modelo global é atualizado e redistribuído iterativamente até que haja convergência.

Apesar da vantagem em privacidade, o aprendizado federado tem como um de seus desafios aprimorar o treinamento de modelos, dada a heterogeneidade tanto de recursos dos clientes quanto dos dados para treinamento. Para contornar esse desafio, técnicas de seleção de clientes são aplicadas para eleger e agrupar aqueles com maior contribuição à convergência do modelo global [6, 7]. Essas abordagens consideram que o servidor tem conhecimento para definir quais clientes possuem dados semelhantes e quais contribuem mais com o treinamento, sem oferecer-lhes a possibilidade de decisão sobre o modelo.

O foco comum desses trabalhos é economizar os recursos computacionais dos clientes no aprendizado federado através de algoritmos de otimização, além do uso de técnicas de poda de parâmetros e hiperparâmetros de modelos profundos [8, 9, 10, 11]. Assim, se por um lado reduz-se o tempo de treinamento e a latência global sem afetar o desempenho, por outro lado altera-se a estrutura típica dos modelos.

Outra forma de tornar o treinamento de modelos mais eficiente é usar entradas menos complexas. No cenário industrial, o emprego de amostras sonoras e suas características espectrais demonstra que modelos menos complexos são capazes de realizar a tarefa de classificação. O conjunto de dados MIMII (*Malfunctioning Industrial Machine Investigation and Inspection*) [12] é um exemplo adotado em sistemas de monitoramento para classificação de estado fora do padrão no maquinário [13, 14]. Desse conjunto de dados, são extraídas características responsáveis por descrever o som emitido de forma concisa em comparação com abordagens baseadas em Mel-Espectrogramas [15, 16].

Esses trabalhos buscam a melhoria do desempenho de sistemas inteligentes, avaliando um conjunto de modelos distintos até obter o mais adequado. Ao treinar múltiplos modelos até o fim e apenas selecionar o melhor para uma aplicação específica, os recursos computacionais, energéticos e o tempo de treinamento são desperdiçados.

Este trabalho concentra-se no cenário industrial considerando implementações de detecção de comportamento anômalo de equipamentos através do som. É proposta a seleção do modelo mais adequado para a detecção de funcionamento anômalo de equipamentos industriais a partir de amostras sonoras [13, 14]. Diferente da literatura, este trabalho não altera a estrutura das arquiteturas de rede neural, uma vez que a adoção das características espectrais já permite que as redes neurais selecionadas sejam reduzidas. A proposta estabelece procedimentos de treinamento com dois servidores de forma exploratória, encerrando o menos promissor o mais cedo possível para evitar impacto no consumo de recursos dos clientes. A abordagem proposta permite que os clientes escolham os servidores no contexto do aprendizado federado, ao invés de apenas os servidores selecionarem o subconjunto de clientes. Além do desempenho do modelo, o encerramento antecipado também considera a latência associada às comunicações com os servidores. Para isso, os clientes iniciam conexões com dois servidores em paralelo até que seja atingida a rodada de avaliação r . Então, os clientes enviam as informações de desempenho obtidas durante o teste dos modelos da rodada 0 até a rodada r a um nó comparador que é responsável por avaliar os modelos considerando dois fatores. O primeiro é baseado na estimativa do valor da métrica AUC-ROC (*Area Under the Curve - Receiver Operating Characteristic*) na rodada $r + 1$, enquanto o segundo é baseado no tempo de comunicação entre cliente-servidor, em que o servidor com menor latência é selecionado caso a diferença no desempenho não seja significativa.

O cliente então mantém o treinamento apenas com o servidor selecionado até o final. Os experimentos considerando o conjunto de dados MIMII demonstram que a métrica AUC-ROC do modelo final é aumentada ou não comprometida em comparação ao trabalho utilizado como base.

Este trabalho está organizado da seguinte forma. A Seção II apresenta a proposta para seleção de modelos. A Seção III apresenta o conjunto de dados, parâmetros e hiperparâmetros definidos para a etapa experimental. Já a Seção IV apresenta os resultados experimentais e as análises correspondentes. Por fim, a Seção V apresenta as conclusões e os trabalhos futuros.

II. SELEÇÃO DE MODELOS BASEADA EM REGRESSÃO LINEAR E LATÊNCIA MÍNIMA

Para a seleção de modelos, o sistema proposto considera que cada cliente c se conectará a mais de um servidor s e iniciará seu treinamento federado em paralelo por r rodadas conforme ilustrado no Passo 01 na Figura 1. Em seguida, no Passo 02 os clientes compartilham com um nó comparador as métricas de desempenho (AUC-ROC e latência do servidor) obtidas nas r rodadas de comunicação iniciais em cada um dos servidores. Considerando que a métrica AUC-ROC é a selecionada para avaliar o desempenho do modelo, estes valores são utilizados como entrada do algoritmo de regressão linear para estimar o próximo valor da métrica AUC-ROC. A estimativa de AUC-ROC futura é adotada como fator de decisão para que um treinamento seja ou não prosseguido no Passo 03. Os valores de latência são considerados na decisão



Fig. 1: Proposta para seleção de servidor baseada na estimativa das métricas de desempenho em rodadas futuras e tempo de comunicação.

caso o segundo maior da métrica AUC-ROC esteja dentro de uma margem estabelecida pelo parâmetro β apresentado a seguir.

Cabe ressaltar que, caso decida-se interromper um procedimento de treinamento com um dado servidor, este é feito através do encerramento da conexão solicitada pelo cliente no Passo 04. Para isso, o nó comparador envia uma mensagem sinalizando a ação a ser tomada pelo cliente.

O pseudocódigo de seleção de modelos é apresentado no Algoritmo 1 executado no nó comparador. As informações de todos os clientes $c \in C$ e servidores $s \in S$ participantes dos treinamentos são utilizadas como entrada do algoritmo. Para os experimentos deste trabalho são adotados 2 servidores. Além dos conjuntos C e S , os parâmetros β , que permite analisar a diferença entre as métricas AUC-ROC; a rodada de avaliação r ; e o vetor $\mathbf{D}_{c,s}$ com as AUC-ROC das rodadas de treinamento anteriores de cada par cliente-servidor e as latências observadas são usadas como entrada do algoritmo proposto.

Algoritmo 1 Seleção de modelos através de métrica de desempenho e latência.

```

1: Entradas:  $C, S, \beta, r, \mathbf{D}_{c,s}$ 
2: se  $r_c = r$  então
3:   Aguardar que todos os clientes em  $C$  enviem os dados de desempenho
   de treinamento em todos os servidores em  $S$ 
4:   para cada cliente  $c \in C$  faça
5:     para cada servidor  $s \in S$  faça
6:       Estimar o próximo valor de métrica  $\alpha(c, s)_{r+1}$  em cada
       servidor através de regressão linear utilizando  $\mathbf{D}_{c,s}$ 
7:     fim para
8:     Ordenar os valores  $\alpha(c, S)_{r+1}$ , sendo  $\alpha_1(c, s)_{r+1}$  o maior valor
     previsto da métrica AUC-ROC e  $\alpha_2(c, s)_{r+1}$  o segundo maior
9:     se  $\Delta_{AUC-ROC}(c)_{r+1} \geq \beta$  então
10:      Encerrar conexão de  $\alpha_2(c, s)_{r+1}$ 
11:     senão
12:      Encerrar a conexão de maior latência
13:     fim se
14:   fim para
15: fim se
    
```

Uma vez que um cliente se encontre na rodada de comunicação r_c destinada a ser a rodada de avaliação r , o próximo valor de métrica de desempenho para cada par cliente-servidor é estimado através de regressão linear. Considerando que $\alpha_1(c)_{r+1}$ é a maior AUC-ROC prevista para o cliente c e

TABELA I: Hiperparâmetros das redes neurais adotadas.

Hiperparâmetros	Rede 1 (Servidor 1)	Rede 2 (Servidor 2)
Nós por camada	(64,32,16,1)	(64,32,1)
Função de Ativação	(-, LeakyReLU, LeakyReLU, Sigmóide)	(-, ReLu, Sigmóide)
Tipo das camadas	Densas	Densas
Função de Custo	Entropia Cruzada	Entropia Cruzada
Otimizador	Adam	Adam
Total de parâmetros treináveis	4.225	3.713

$\alpha_2(c)_{r+1}$ a segunda maior, ambos os valores são comparados utilizando a Equação 1.

$$\Delta_{AUC-ROC}(c)_{r+1} = \frac{|\alpha_1(c)_{r+1} - \alpha_2(c)_{r+1}|}{\alpha_2(c)_{r+1}} \geq \beta, \quad (1)$$

onde apenas o servidor com o maior valor de AUC-ROC é selecionado para a continuação do treinamento do modelo se $\beta = 0.0$. Porém, quanto maior o valor de β , maior a diferença tolerada entre $\alpha_1(c)_{r+1}$ e $\alpha_2(c)_{r+1}$ para que a escolha do servidor seja realizada através do critério de latência.

Note que o objetivo é considerar a latência de comunicação apenas se a diferença de desempenho não for significativa conforme o parâmetro β .

III. EXPERIMENTOS

Para etapa experimental, foi utilizada uma máquina com CPU Intel Core i7-4790 e 16 GB de RAM. A linguagem Python foi adotada para os experimentos de aprendizado de máquina. Esta seção apresenta os equipamentos selecionados do conjunto de dados MIMII, as características espectrais extraídas do conjunto das amostras sonoras, os parâmetros e hiperparâmetros adotados e latências simuladas nos experimentos.

Conjunto de dados: O conjunto de dados MIMII contém originalmente amostras sonoras de máquinas industriais em funcionamento normal e fora do padrão esperado. O conjunto reúne quatro tipos de máquinas e as amostras possuem 10 segundos de duração. Além disso, cada tipo de máquina contém amostras de quatro modelos individuais identificados por 00, 02, 04 e 06. Os tipos de máquinas são: ventiladores industriais, bombas hidráulicas, trilhos deslizantes e válvulas solenoides. No conjunto original, as amostras estão disponíveis nas relações sinal-ruído: 6 dB, 0 dB e -6 dB. Para a etapa experimental, foram selecionados 12 máquinas no total, sendo 3 de cada um dos 4 tipos de máquinas diferentes e apenas as amostras com relação sinal-ruído de 6 dB.

Características espectrais: As características do espectro sonoro são extraídas dos sons originais através da biblioteca librosa [17] para descrever de forma concisa as amostras do conjunto original. Para isso, as seguintes características foram adotadas: Croma (*Chroma*), Coeficientes Mel-cepstrais (*Mel-frequency Cepstral Coefficients*), Centroid Espectral (*Spectral Centroid*), Largura Espectral (*Spectral Band*), Decaimento Espectral (*Spectral roll-off*) e Taxa de Cruzamento por Zero (*Zero Crossing Rate*).

Ambiente do aprendizado federado: A proposta foi implementada utilizando a biblioteca Flower [18] para treinamento do aprendizado federado. O conjunto de dados original possui

um número de amostras de estado normal maior que o de estados anômalos, dificultando o ajuste dos modelos de classificação. Para contornar este desafio, a técnica SMOTE (*Synthetic Minority Over-Sampling Technique* – Técnica de Sobreamostragem Minoritária Sintética) [19] é adotada na etapa de treinamento para balancear o número de amostras.

Parâmetros e hiperparâmetros das redes neurais: Conforme mencionado anteriormente, a etapa experimental é realizada utilizando 12 clientes do conjunto original. Dois servidores são selecionados, cujos hiperparâmetros são apresentados na Tabela I. Estes hiperparâmetros são utilizados considerando trabalhos anteriores. O critério de parada final é o número máximo de rodadas, r_{ct} , definido como 50. Além disso, os experimentos foram realizados com valores da rodada de avaliação r iguais a 5, 10 e 15. Isso significa que os modelos serão analisados em 10%, 20% e 30% do total de rodadas pré-determinado. Por fim, o parâmetro β assume os valores de 0, 0,05 e 0,01. Desta forma, o servidor de menor latência é escolhido em cenários em que o valor da maior métrica AUC-ROC $\alpha_1(c, s)_{r+1}$ é 0%, 5% e 10% superior ao segundo valor previsto $\alpha_2(c, s)_{r+1}$, respectivamente.

Latências dos servidores: Para simular os valores de latência foi considerado os requisitos de aplicações de monitoramento de segurança e alarmes de controle no contexto industrial definidos como entre 5ms e 100ms [20]. Para isso, este intervalo foi dividido em dois: (A) de 5ms a 52ms, e (B) de 53ms a 100ms. O servidor 1 possui a rede neural com mais camadas e parâmetros treináveis e possui latências na faixa B e o servidor 2 possui latências na faixa A. O objetivo é simular um contexto onde modelos mais complexos podem necessitar de mais recursos computacionais e adotar servidores mais distantes. Desta forma, o servidor mais complexo tende a ter resultados melhores, mas o servidor mais simples é privilegiado quando a latência é considerada. Na inicialização do treinamento, a latência das conexões é definida de forma aleatória.

IV. RESULTADOS

Para avaliação da proposta, a latência acumulada ao longo do processo de treinamento dos modelos é observada. A latência acumulada é a soma do tempo de latência de cada servidor para cada cliente durante o treinamento federado. Este valor é fixo para cada par cliente-servidor, definido de maneira aleatória, considerando o intervalo de valores apresentado na seção anterior e determinado no início do treinamento. Além disso, os valores de AUC-ROC dos modelos finais pelos clientes. Os resultados da proposta, considerando a estimativa do próximo valor de AUC-ROC, são comparados com o

trabalho anterior dos próprios autores, chamado de “Ranqueamento” [21]. No “Ranqueamento”, apenas a métrica AUC-ROC em r é considerada para decisão sobre a manutenção da conexão com os servidores. A principal diferença entre ambas é a estimativa do próximo valor através de regressão linear e considerar uma margem de avaliação da métrica para considerar a latência como fator de decisão.

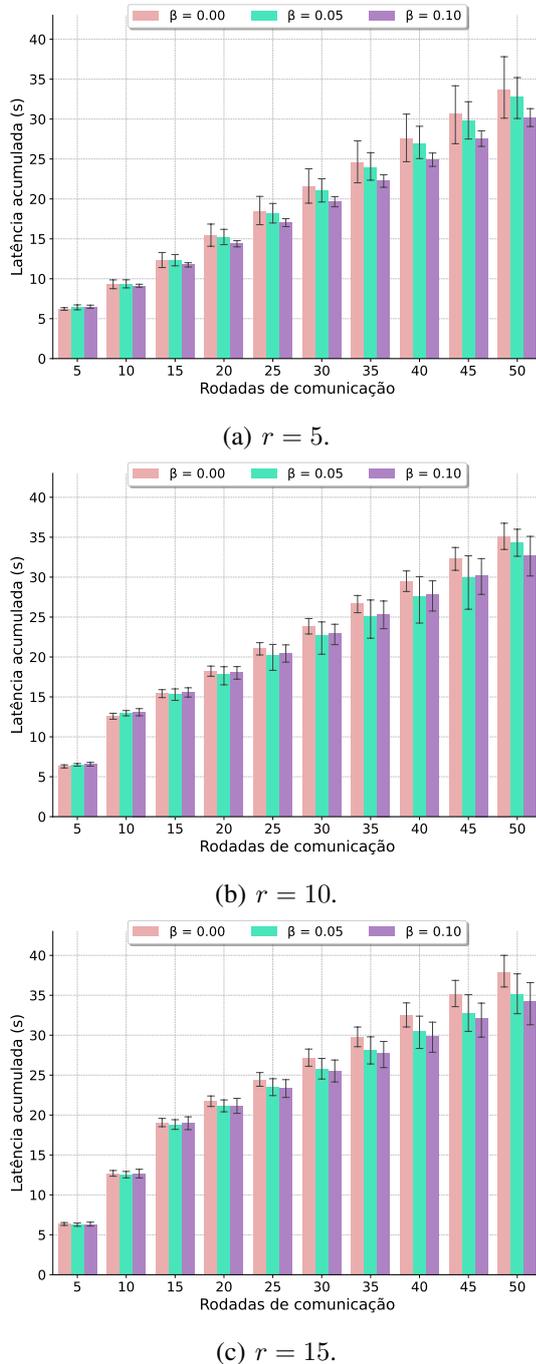


Fig. 2: Latência acumulada durante as rodadas de comunicação de treinamento para os distintos valores de r e β .

e as propostas com distintos valores de β . Os resultados são apresentados com intervalo de confiança de 95%, cada cenário foi repetido 10 vezes de forma independente e a figura sintetiza o resultado obtido por todos os 12 clientes.

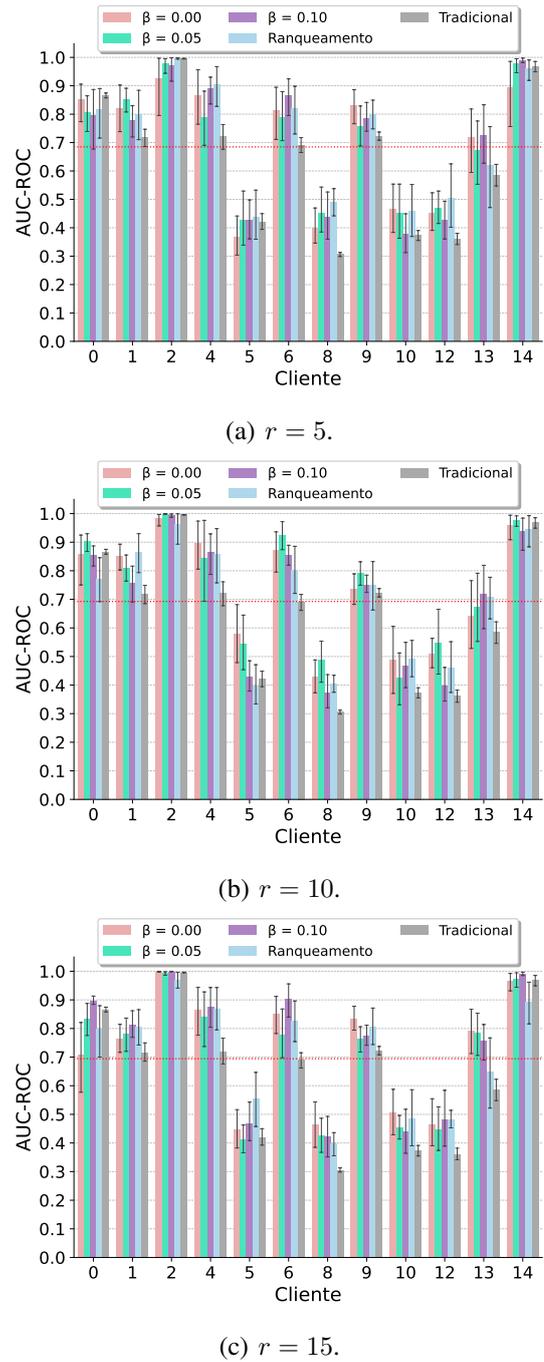


Fig. 3: Métrica de desempenho AUC-ROC obtida pelos modelos finais na proposta para distintos valores de β em comparação com trabalho relacionado.

A latência acumulada em segundos durante o treinamento dos modelos é apresentada na Figura 2a para $r = 5$, na Figura 2b para $r = 10$ e na Figura 2c para $r = 15$. Conforme o esperado, a latência acumulada observada para todos os valores de β permanece igual até sua respectiva rodada de decisão r . Em todas as variações de r o valor da latência acumulada ao final das 50 rodadas de comunicação é o maior alcançado com $\beta = 0,0$. Este fato indica que o servidor com menor latência é descartado pela maioria dos clientes

quando a escolha considera apenas a métrica $\alpha_1(c, s)_{r+1}$ ao invés de analisar quão maior $\alpha_1(c, s)_{r+1}$ é de $\alpha_2(c, s)_{r+1}$. Comparando $\beta = 0,0$ e $\beta = 0,15$ observa-se a redução da latência acumulada é de aproximadamente 10,4% em $r = 5$, 6,9% em $r = 10$ e 9,6% em $r = 15$. Conforme o esperado, quanto antes a avaliação é realizada e maior a tolerância entre os valores da métrica AUC-ROC, menor o tempo de latência observado. Desta forma, além da redução do gasto de recursos em rodadas de comunicação de servidores menos promissoras, o modelo final é gerado mais rapidamente.

A Figura 3 apresenta a métrica de desempenho AUC-ROC obtida nos modelos finais para cada um dos clientes. A proposta definida como Tradicional é baseada em aprendizado federado sem que ocorra a seleção dos modelos. Desta forma, todos os clientes permanecem em ambos os servidores. A linha pontilhada em vermelho representa o valor médio de todos os clientes em todos os modelos. É possível perceber que metade dos clientes não contribuem para a convergência dos modelos ao se manterem abaixo da média. Na maioria dos cenários avaliados a proposta baseada em estimativa de próximo valor e escolha por latência mínima obtém valor de métrica próximo ou superior ao Ranqueamento, com o adicional de avaliar a redução de latência. No cenário de maior melhoria percentual a métrica, a proposta aumentou em 45,55% em relação ao trabalho baseado em Ranqueamento. No cenário de maior melhoria absoluta da métrica o aumento foi de aproximadamente 15%.

V. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta para seleção de modelos de aprendizado federado considerando a estimativa de próximo valor da métrica AUC-ROC obtido pelo nó comparados através dos dados dos clientes e a menor latência de comunicação cliente-servidor. Os clientes treinam os modelos paralelamente até a rodada de avaliação r quando apenas a conexão menos promissora é encerrada. O objetivo é reduzir a latência acumulada sem comprometer o desempenho. A proposta estende trabalhos anteriores onde apenas o valor observado na rodada de avaliação r é utilizado como fator de decisão, sem avaliar latência ou tendência de cada cliente durante o treinamento.

Os resultados da proposta demonstram um aumento da métrica AUC-ROC do modelo final, em comparação com o trabalho base que considera o ranqueamento simples das métricas, de 0,80% para 0,92% no caso mais expressivo, significando um aumento de 15%. Além disso, a latência acumulada para treinamento do modelo foi reduzida em 10,4%.

Como trabalhos futuros, pretende-se adotar maior número de servidores e diferentes tipos de clientes. Além disso, métricas adicionais de desempenho dos modelos devem ser inseridas.

REFERÊNCIAS

- [1] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467–3501, 2019.
- [2] K. Bochie, M. S. Gilbert, L. Gantert, M. S. Barbosa, D. S. Medeiros, and M. E. M. Campista, "A survey on deep learning for challenged networks: Applications and trends," *Journal of Network and Computer Applications*, vol. 194, p. 103213, 2021.
- [3] S. Wang, Y. Liang, W. Li, and X. Cai, "Big data enabled intelligent immune system for energy efficient manufacturing management," *Journal of cleaner production*, vol. 195, pp. 507–520, 2018.
- [4] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *Journal of cleaner production*, vol. 252, p. 119869, 2020.
- [5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.
- [6] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things Journal*, 2023.
- [7] L. A. C. de Souza, G. F. Camilo, M. Sammarco, M. E. M. Campista, and L. H. M. Costa, "Aprendizado federado com agrupamento hierárquico de clientes para aumento da acurácia," in *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 545–558, SBC, 2022.
- [8] N. Huang, M. Dai, Y. Wu, T. Q. Quek, and X. Shen, "Wireless federated learning with hybrid local and centralized training: A latency minimization design," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 248–263, 2022.
- [9] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [10] R. Lin, Y. Xiao, T.-J. Yang, D. Zhao, L. Xiong, G. Motta, and F. Beaufays, "Federated pruning: improving neural network efficiency with federated learning," *arXiv preprint arXiv:2209.06359*, 2022.
- [11] B. Alhalabi, S. Basurra, and M. M. Gaber, "Fednets: Federated learning on edge devices using ensembles of pruned deep neural networks," *IEEE Access*, vol. 11, pp. 30726–30738, 2023.
- [12] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *arXiv preprint arXiv:1909.09347*, 2019.
- [13] B. Natesha and R. M. R. Guddeti, "Fog-based intelligent machine malfunction monitoring system for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7923–7932, 2021.
- [14] L. Gantert, T. Zeffiro, M. Sammarco, and M. E. M. Campista, "Multi-class classification of faulty industrial machinery using sound samples," *Engineering Applications of Artificial Intelligence*, vol. 136, p. 108943, 2024.
- [15] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 271–275, IEEE, 2020.
- [16] B. Tao, C. Chen, and H. Chen, "Communication efficient federated learning via channel-wise dynamic pruning," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2023.
- [17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, pp. 18–25, 2015.
- [18] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.
- [19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [20] Y. Lu, L. Yang, S. X. Yang, Q. Hua, A. K. Sangaiah, T. Guo, and K. Yu, "An intelligent deterministic scheduling method for ultralow latency communication in edge enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1756–1767, 2022.
- [21] L. Gantert and M. Campista, "Seleção de modelo de aprendizado federado baseado em busca e poda para detecção de defeitos industriais," in *Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, (Porto Alegre, RS, Brasil), pp. 1106–1119, SBC, 2024.