NeuralMatch: Identificando a Similaridade de Clientes baseado em Modelos no Aprendizado Federado

Gabriel Ukstin Talasso, Allan M. de Souza, Leandro A. Villas

[1]Universidade Estadual de Campinas, Brasil

g235078@dac.unicamp.br, {allanms, lvillas}@unicamp.br

Resumo. *O aprendizado federado é uma técnica de aprendizado de máquina distribuído que permite que vários dispositivos colaborem no treinamento de um modelo de dados comum, enquanto preserva a privacidade dos dados do usuário. No entanto, o aprendizado federado apresenta desafios relacionados aos dados não identicamente distribuídos e balanceados, o que pode resultar em modelos menos precisos. Dessa forma, foi proposto o NeuralMatch, um arcabouço para identificar similaridade de modelos para aprendizado federado, capaz de identificar a similaridade entre os clientes sem o compartilhamento de dados. O arcabouço proposto pode ajudar a desenvolver soluções mais eficientes de aprendizado federado para lidar com os problemas de dados não identicamente balanceados e distribuídos.*

Abstract. *Federated learning is a distributed machine learning technique that allows multiple devices to collaborate on training a common data model, while preserving the privacy of user data. However, federated learning presents challenges related to non-identically distributed and balanced data, which can result in less accurate models. Thus, it was proposed the NeuralMatch, a framework to identify similarity of models for federated learning, capable of identifying the similarity between clients without sharing data. This proposed framework can help develop more efficient federated learning solutions to deal with the problems of not identically balanced and distributed data.*

1. Introdução

O advento da Inteligência Artificial (IA) permitiu o desenvolvimento de soluções inteligentes para melhorar o cotidiano de milhões de pessoas em todo o mundo, alterando a maneira como vivemos, trabalhamos e nos divertimos [Lim et al. 2020, Abdulrahman et al. 2021]. Essas soluções baseiam-se na coleta de dados de dispositivos (e.g., *smartphones*, dispositivos IoT, veículos, etc.) utilizando sensores físicos e/ou virtuais. Em seguida, os dados coletados são agregados e processados por uma entidade central, onde algoritmos de aprendizado de máquina e IA são aplicados, gerando conhecimento para desenvolver soluções inteligentes nas mais diversas áreas.

Neste cenário, os dispositivos compartilham os dados com servidores centrais, consequentemente gerando problemas de segurança e privacidade. Pois, os dados compartilhados podem conter dados sensíveis dos usuários finais e podem ser interceptados por indivíduos mal-intencionados ou até mesmo acessados por outras entidades [Liu et al. 2021]. Problemas de privacidade dos dados tornaram-se ainda mais relevantes com os recentes avanços de leis de proteção de dados em diversos países [Pinheiro 2020, Goddard 2017].

Tais problemas podem ser reduzidos utilizando o aprendizado federado (*Federated Learning* - FL) [Imteaj et al. 2022], uma técnica de aprendizado de máquina distribuído que

permite que vários dispositivos de computação colaborem no treinamento de um modelo de dados comum. A necessidade do aprendizado federado surge devido a privacidade dos dados do usuário, garantindo que os dados permaneçam no dispositivo do usuário e não sejam transferidos para qualquer outra entidade [Abdulrahman et al. 2021]. Porém, o aprendizado federado apresenta alguns desafios, especialmente relacionados aos dados não identicamente distribuídos e balanceados (i.e., dados não-IID). A distribuição desigual de dados em diferentes dispositivos pode levar a uma falta de representatividade, resultando em um modelo menos preciso e menos justo entre os participantes. Além disso, a natureza descentralizada do aprendizado federado pode tornar mais difícil garantir a qualidade dos dados e detectar problemas em potencial. Portanto, a seleção de algoritmos e modelos adequados é fundamental para garantir que os dados sejam adequadamente processados e os resultados sejam precisos.

Uma abordagem para reduzir o problema dos dados não-IID é a identificação de similaridade das distribuições de dados dos dispositivos, para desenvolver soluções mais eficientes baseadas em: *(i)* agregação de modelos; *(ii)* agrupamento de dispositivos; e *(iii)* seleção de clientes. Assim, surge a seguinte pergunta de pesquisa: como identificar a similaridade das distribuições de dados entre os dispositivos sem o compartilhamento de dados?

Dessa forma, introduzimos **NeuralMatch**, uma abordagem para identificar similaridade entre dispositivos em um contexto de aprendizado federado. NeuralMatch utiliza os modelos compartilhados pelos clientes para identificar a similaridade, pois modelos treinados em dados semelhantes potencialmente convergem para os mesmos pesos e gradientes [Kornblith et al. 2019]. Resultados iniciais mostraram que NeuralMatch é capaz de identificar a similaridade entre os clientes sem o compartilhamento de dados, assim, tornando-o uma ferramenta adequada para desenvolver soluções mais eficientes de aprendizado federado.

O restante do trabalho é organizado da seguinte forma, a Seção 2 descreve as tecnologias utilizadas nesse trabalho. A Seção 3 apresenta o NeuralMatch, um *framework* para identificar a similaridade de dispositivos em aprendizado federado. A Seção 4 descreve a ambiente de análise, a configuração das avaliações e também os resultados iniciais do NeuralMatch. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros a serem desenvolvidos baseados nessa solução.

2. Referencial Teórico

Esta Seção descreve as tecnologias utilizadas neste trabalho. Assim, a Subseção 2.1 define o aprendizado federado destacando seu objetivo e desafios, enquanto a Subseção 2.2 descreve o problema dos dados não-IID enfrentado por soluções de aprendizado federado.

2.1. Aprendizado Federado e Federated Averaging

O Aprendizado Federado é uma técnica de aprendizado de máquina em que um modelo é treinado colaborativamente usando dados de vários dispositivos ou nós distribuídos em uma rede, sem compartilhar os próprios dados. Em vez disso, os dispositivos treinam o modelo localmente em seus próprios dados e enviam apenas atualizações para um servidor central, que agrega as atualizações e melhora o modelo. Essa abordagem preserva a privacidade e a segurança dos dados, pois os dados permanecem no dispositivo, permitindo ao mesmo tempo o uso de um conjunto de dados mais diversificado e representativo para treinamento, o que pode melhorar a precisão e robustez do modelo. O Aprendizado Federado tem sido cada vez mais utilizado em cenários onde a privacidade dos dados é uma preocupação, como na área da saúde e em aplicações móveis.

O FedAvg [McMahan et al. 2017] (Federated Averaging) é uma abordagem tradicional de aprendizado federado, o qual agrega os modelos dos clientes realizando a média ponderada dos pesos dos modelos. Considere um cenário com K clientes e T rodadas de comunicação. A cada rodada $t \in \{1, 2, 3, \ldots, T\}$ um conjunto $C \leq K$ aleatório de clientes é selecionado para participar do treinamento do modelo. Cada cliente $k \in C$ possui seu conjunto de dados locais D_k, $n_k = |D_k|$ e $n = |\bigcup_{k \in C} D_k|$. Dessa forma, cada cliente recebe os parâmetros (i.e., pesos e/ou gradientes) atuais do modelo w_t do servidor e atualiza seu modelo local para treinar o modelo em seus dados, em seguida o cliente envia os novos parâmetros do modelo após o treinamento w_{t+1}^k para o servidor. Assim o servidor agrega os modelos recebidos da seguinte forma:

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

onde w_{t+1} é o modelo global atualizado e w_{t+1}^k é o modelo treinado recebido do cliente k. Assim, o objetivo do FedAvg é minimizar a seguinte equação:

$$\min f(w) \text{ onde } f(w) = \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}(x_k, y_k; w), \tag{2}$$

onde w é o modelo global, $\mathcal{L}$ representa a função de perda do modelo, x_k é o conjunto de dados de entrada e y_k são os rótulos do conjunto de dados.

Como pode ser visto, o FedAvg calcula a média ponderada dos modelos recebidos. Para cenários com distribuições IID isso não é um problema pois os modelos dos clientes irão convergir para os mesmos pesos e gradientes por possuírem distribuições semelhantes. Entretanto, para cenários com distribuições de dados não-IID, diferentes clientes podem convergir para gradientes e pesos muitos distantes. Portanto, ao realizar a média ponderadas desses modelos, o modelo global desvia do ótimo global em direção ao modelo do cliente com mais amostras de dados (i.e., o que não é o ideal, pois tal cliente pode não ser o mais representativo), consequentemente degradando a eficiência do modelo global. Dessa forma, endereçar o problema dos dados não-IID é um desafio a ser considerado para permitir soluções mais eficientes de aprendizado federado.

2.2. Dados Não Independentes e Identicamente Distribuídos

Em teoria das probabilidades e estatística, distribuição de dados é independente e identicamente distribuída (i.e., IID) se cada amostra desses dados tiver a mesma distribuição de probabilidade das demais e se todas elas forem mutuamente independentes. Considere que as distribuições de dados dos clientes assumam valores em $\mathcal{I} \in \mathcal{R}$. Portanto, duas amostras X e Y são identicamente distribuídas, se somente se $F_X(x) = F_Y(x), \forall x \in \mathcal{I}$ e são independentes, se somente se $F_{X,Y}(x, y) = F_X(x)F_Y(y), \forall x, y \in \mathcal{I}$, onde, nesse caso, F_X e F_Y representam as funções de probabilidade acumuladas (CDF) de X e de Y, ou seja, $F_X(x) = P(X \leq x)$.

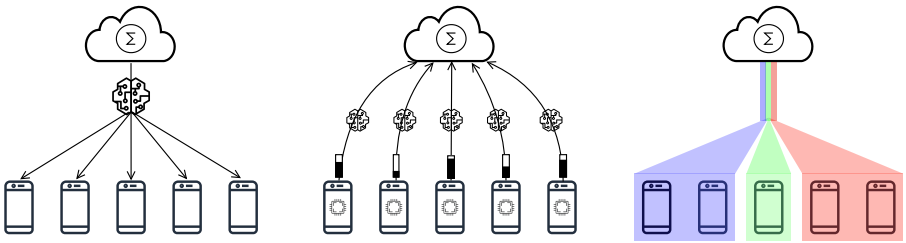
Em outras palavras, a primeira equação nos mostra que a probabilidade de se obter um valor x na distribuição de X é a mesma de obter esse x na distribuição de Y, para qualquer x no intervalo definido. Enquanto a segunda equação mostra que a probabilidade de ocorrer um evento em X e um em Y é sempre igual a multiplicação das probabilidades marginais de ambas. Portanto, dados não-IID não seguem essas características, assim introduzindo novos desafios nos treinamentos dos modelos. Pois clientes com diferentes distribuições potencialmente produzem parâmetros distintos para o modelo, assim a agregação desses modelos pode reduzir o desempenho no modelo global e causar dificuldades nas generalizações.

3. Identificando Similaridade de Clientes Sem Compartilhamento de Dados

Esta Seção descreve NeuralMatch, apresentando detalhes das abordagens utilizadas para identificar a similaridade de clientes baseado nos modelos compartilhados durante o treinamento de modelos federados. Ou seja, utilizando os modelos de clientes, mostramos que podemos quantificar a similaridade dos dados em que foram treinados. Dessa forma, quando dizemos "similaridade de clientes"nos referimos às similaridades dos dados locais e mostramos que isso implica em similaridade dos modelos compartilhados. Assim, a Subseção 3.1 apresenta uma visão geral do arcabouço proposto, enquanto a Subseção 3.2 detalha o método de similaridade utilizado.

3.1. Descrição do Arcabouço

A Figura 1 descreve o processo completo aplicado pelo NeuralMatch. Como pode ser visto na Figura 1(a), inicialmente o servidor envia o modelo global para todos os clientes. Em seguida, os clientes realizam o treinamento do modelo com seus dados locais e o compartilham com o servidor, enviando também a acurácia obtida após o treinamento, descrito na Figura 1(b). Assim, na Figura 1(c), o servidor recebe os modelos dos clientes, realiza a agregação fazendo uma média ponderada dos modelos, ordena de maneira crescente e também calcula a similaridade dos modelos utilizando a técnica *Centered Kernel Alignment (CKA)* [Kornblith et al. 2019]. Dessa forma, com a identificação das similaridades dos clientes, é possível derivar soluções mais eficientes para endereçar os problemas de *(i)* dados não IID; *(ii)* agregação de modelos; e até mesmo *(iii)* seleção de clientes. No exemplo apresentado na Figura 1, é possível identificar as similaridades dos clientes de acordo com a organização das cores.



(a) Passo 1 (b) Passo 2 (c) Passo 3

Figura 1. Visão geral do NeuralMatch

3.2. Similaridade entre Modelos

A similaridade é calculada através da técnica *Centered Kernel Alignment (CKA)*, baseada em Análise de Correlações Canônicas (CCA) [Kornblith et al. 2019], a qual é capaz de verificar a correlação entre matrizes com valores entre $[0, 1]$. Assim, NeuralMatch utiliza as matrizes que representam os modelos de cada cliente (i.e., w_i) e calcula esse coeficiente da seguinte forma:

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\,\mathrm{HSIC}(L, L)}}, \tag{3}$$

Assumindo que HSIC representa o Hilbert-Schmidt Independence Criterion:

$$\mathrm{HSIC}(K, L) = \frac{1}{(n-1)^2}\,\mathrm{tr}(KHLH), \tag{4}$$

e H é a matriz de centralização $H_n = I_n - \frac{1}{n}\underset{\sim}{\mathbf{1}} \times \underset{\sim}{\mathbf{1}}^{\mathrm{T}}$, sendo $\underset{\sim}{\mathbf{1}}^{\mathrm{T}} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$, de tamanho n e I_n a identidade de ordem n, enquanto $K = k(w_{c_i}^j, w_{c_i}^k)$ e $L = l(w_{c_i}^j, w_{c_i}^k)$ são

dois *kernels* aplicados nas matrizes de ativação da camada $c_i \in w^j$ do modelo do cliente j em relação as ativações da camada $c_i \in w^k$ do modelo do cliente k. Portanto, considerando a camada $c_i \in w$, cada linha da matriz é a representação de um dado (uma imagem) após sua passagem por c_i . Sendo assim, a matriz terá o número de linhas igual ao tamanho do conjunto de dados e número de colunas igual ao número de neurônios na camada $c_i \in w$.

O *kernel* aplicado é linear, ou seja, $K = k(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T\mathbf{x}$ e $L = l(\mathbf{y}, \mathbf{y}) = \mathbf{y}^T\mathbf{y}$. Dessa forma a métrica é calculada com finalidade de entender a similaridade entre as ativações de camadas de diferentes redes. A ideia é que redes treinadas em dados parecidos terão uma alta similaridade em suas ativações, tornando possível a identificação de clientes treinados em dados distintos e portanto não-IIDs.

Para fins de verificação da funcionalidade do método, podemos utilizar uma métrica para identificar se os dados dos clientes eram realmente diferentes em suas distribuições, e comparar isso com o valor do CKA obtido. A métrica EMD (Earth Mover's Distance) [Rubner et al. 1998] é uma medida de distância entre duas distribuições de probabilidade. A ideia por trás é imaginar as duas distribuições como "montes de terra", onde cada ponto da distribuição representa uma quantidade de terra. A métrica EMD calcula a quantidade de trabalho necessário para transformar um monte de terra no outro, onde o trabalho é medido pelo custo de mover cada ponto de terra de um monte para o outro. Ela pode ser calculada de forma geral como a Distância de Wasserstein de primeira ordem $(p = 1)$ [Levina and Bickel 2001] :

$$W_p(P, Q) = \left(\frac{1}{n} \sum_{i=1}^{n} \left\| X_{(i)} - Y_{(i)} \right\|^p \right)^{1/p} \tag{5}$$

Onde P e Q são as duas distribuições a serem comparadas e X_i e Y_i são amostras de tamanho n dessas distribuições, obtidas empiricamente.

4. Análise de Desempenho

Esta Seção apresenta a análise de desempenho do NeuralMatch, onde a Subseção 4.1 apresenta a metodologia utilizada para a avaliação. A Subseção 4.2 apresenta uma análise da similaridade dos modelos por camada considerando dados IID e não-IID. A Subseção 4.3 faz uma comparação entre a similaridade obtida pelo NeuralMatch e a similaridade da distribuição real dos dados dos clientes. Por fim, a Subseção 4.4 analisa o processo de treinamento federado do modelo para validar se NeuralMatch permite o treinamento federado eficiente.

4.1. Metodologia de Avaliação

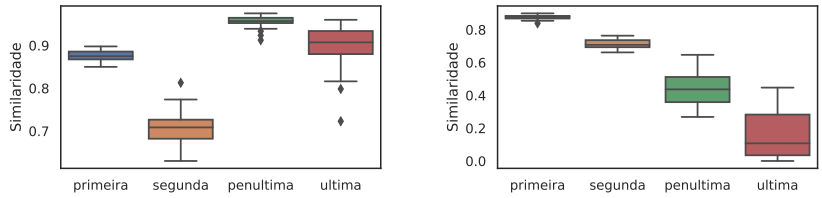
Para avaliar o NeuralMatch, consideramos os seguintes casos: *(i)* avaliar o processo de aprendizado federado para garantir que os clientes estão treinando o modelo de forma colaborativa para um problema específico; e *(ii)* verificar a eficiência do método de similaridade baseada em modelos. Dessa forma, foi desenvolvido um ambiente para simulação utilizando o *framework* de aprendizado federado Flower [Beutel et al. 2020], e a base de dados MNIST com o objetivo principal de treinar um modelo federado para a classificação de imagens. O modelo foi desenvolvido utilizando TensorFlow[1] versão 2.11. O modelo é um *Multi Layer Perceptron - MLP*, composto por três camadas ocultas, com 128 unidades por camada oculta [Vaizman et al. 2017] e mais uma camada de *softmax* para identificar a classe da imagem predita. Para o ajuste do MLP, foram utilizados os seguintes parâmetros: o *Stochastic*

[1] `https://www.tensorflow.org/`

Gradient Descent (SGD) como método para atualizar os pesos do modelo MLP e o *Sparse Categorical Crossentropy* como função de perda. Por fim, o treinamento federado é realizado com 10 rodadas de comunicação, onde cada cliente realiza 10 épocas de treinamento local antes do compartilhamento do modelo treinado com o servidor para ser agregado.

Neste cenário, as seguintes métricas foram analisadas: *(i) Loss*, a qual é a função de perda para analisar o aprendizado do modelo; *(ii)* Acurácia, para avaliar o desempenho do modelo para classificação de imagens; *(iii)* Similaridade entre os modelos utilizando NeuralMatch; e *(iv)* similaridade entre os dados dos clientes utilizando EMD para validação.

4.2. Análise da Similaridade das Camadas do Modelo com Dados IID e não-IID



(a) Todas classes no cliente. (b) Duas classes no cliente.

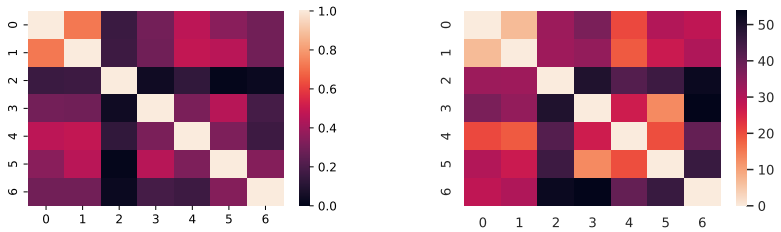
Figura 2. Comparação entre as similaridades das redes de clientes.

O objetivo dessa avaliação é analisar a similaridade das camadas do modelo de cada cliente considerando distribuições de dados IID e também não-IID. Assim, a Figura 2(a) apresenta a similaridade da métrica CKA com clientes com dados IID (i.e., todos os clientes possuem todas as classes com alta probabilidade). Como esperado, todas as camadas do modelo possuem uma similaridade superior a 70% uma vez que os clientes possuem distribuições semelhantes de dados. Por outro lado, ao considerar dados não-IID na Figura 2(b), onde os clientes recebem apenas 2 classes diferentes, verificamos uma queda na similaridade das últimas camadas nesse exemplo quando comparado com o exemplo anterior, onde as distribuições eram semelhantes. As primeiras camadas mantiveram-se em níveis proporcionais aos dados IID. Assim, é possível concluir que as últimas camadas dos modelos têm um poder discriminatório maior e são a melhor opção a serem utilizadas para identificar clientes non-IID.

4.3. Similaridade de Modelos vs. Similaridade de Dados

O objetivo desta análise é verificar se é possível identificar a similaridade dos clientes apenas utilizando modelos e se de fato a similaridade identificada é coerente com a distribuição dos dados. Portanto, foi analisada a similaridade dos modelos com NeuralMatch e também a similaridade dos dados com EMD. Além disso, para garantir uma distribuição não-IID para os clientes, definimos um subconjunto de sete clientes com as seguintes configurações: **Clientes 0 e 1**: uma amostra do conjunto completo (todas as classes); **Clientes 2 e 3**: classes 0 e 1 para o cliente 2 e classes 2 e 3 para o cliente 3; **Cliente 4**: classes 2, 3, 4 e 5; **Cliente 5**: classes 2, 3, 4 e 5 nas respectivas proporções, 50%, 30%, 10%, e 10%; e **Cliente 6**: classes 6, 7, 8 e 9 nas respectivas proporções, 50%, 30%, 10%, e 10%.

A Figura 3 apresenta a similaridade calculada pelo NeuralMatch e pela similaridade dos dados dos clientes (i.e., EMD). De acordo com a análise da similaridade das camadas do modelo apresentada na Figura 3, verificamos que a última camada é a que melhor ilustra a divergência dos dados dos clientes. Portanto, para esta análise consideramos os dados



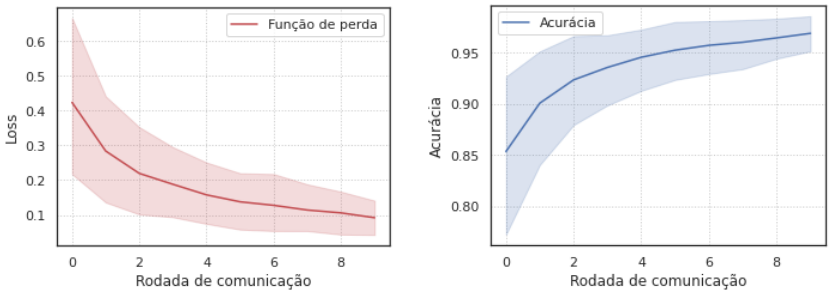
(a) NeuralMatch (b) EMD

Figura 3. Comparação entre as similaridades das redes de clientes na última camada e a distância (EMD) entre as distribuições (normalizada).

das últimas camadas dos clientes. Como pode ser visto, a similaridade observada do NeuralMatch (Figura 3(a)) é semelhante a similaridade apresentada pelos dados (Figura 3(b)). Com isso, podemos concluir que clientes mais heterogêneos possuem similaridades menores, confirmando novamente a eficácia do NeuralMatch.

4.4. Análise do Treinamento Federado

Por fim, analisamos o processo de treinamento federado do NeuralMatch. A Figura 4 apresenta os resultados da função de perda e de acurácia do modelo federado em relação às rodadas de comunicação. Como pode ser visto, o ambiente desenvolvido permite o treinamento colaborativo do modelo, atingindo uma perda menor que 0.1 e uma acurácia superior a 95%. Dessa forma, podemos concluir que o NeuralMatch permite o treinamento federado de mode-



(a) Loss do modelo federado (b) Acurácia do modelo federado

Figura 4. Análise de desempenho do modelo federado no NeuralMatch.

los de forma eficiente e também identifica a similaridade de clientes considerando apenas os modelos compartilhados pelos clientes. Consequentemente, preservando a privacidade dos clientes (i.e., o que é um requisito essencial no aprendizado federado), uma vez que os dados não são compartilhados com um servidor central.

5. Considerações Finais

Neste trabalho introduzimos NeuralMatch, uma abordagem para identificar a similaridade de clientes sem o compartilhamento de dados em cenários de aprendizado federado. NeuralMatch é capaz de identificar a similaridade dos clientes apenas explorando os modelos compartilhados a cada rodada de comunicação. Para isso, NeuralMatch utiliza a técnica CKA que é baseada em correlações de canônicas.

Para avaliar NeuralMatch, foi desenvolvido um ambiente de aprendizado federado onde os clientes treinam o modelo localmente, em seguida o modelo é compartilhado com o servidor para realizar a agregação e também para calcular as similaridades. Além disso, uma análise das similaridades entre as camadas de diferentes redes foi realizada explorando tanto dados IID quanto não-IID. Os resultados mostraram que NeuralMatch, permite um treinamento federado de modelos e também a identificação de similaridade dos clientes apenas explorando os modelos compartilhados, consequentemente NeuralMatch ainda mantém a privacidade dos clientes que é um requisito essencial em aprendizado federado. Como trabalhos futuros, uma vez mostrada a eficácia do método, serão estudadas formas de utilizar a similaridade fornecida pelo NeuralMatch para desenvolver soluções mais eficientes de seleção inteligente de clientes, o que auxilia no desempenho de todo o sistema de aprendizado federado, agregação customizada de modelos e compartilhamento de modelos em agrupamentos de clientes, tendo em vista a melhor generalização dos modelos globais e aumento da performance de ambientes federados.

6. Agradecimentos

Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovação, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado como Agentes inteligentes para plataformas móveis baseados em tecnologia de Arquitetura Cognitiva (processo 01245.013778/2020-21).

Referências

Abdulrahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C., and Guizani, M. (2021). A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., and Lane, N. D. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Goddard, M. (2017). The eu general data protection regulation (gdpr): European regulation that has a global impact. *International Journal of Market Research*, 59(6):703–705.

Imteaj, A., Thakker, U., Wang, S., Li, J., and Amini, M. H. (2022). A survey on federated learning for resource-constrained iot devices. *IEEE Internet of Things Journal*, 9(1):1–24.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR.

Levina, E. and Bickel, P. (2001). The earth mover's distance is the mallows distance: some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256 vol.2.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 22(3):2031–2063.

Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., and Lin, Z. (2021). When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2).

McMahan, B., Moore, E., Ramage, D., Hampson, S., , and Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data.

Pinheiro, P. P. (2020). *Proteção de Dados Pessoais: Comentários à Lei n. 13.709/2018-LGPD*. Saraiva Educação SA.

Rubner, Y., Tomasi, C., and Guibas, L. (1998). A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66.

Vaizman, Y., Ellis, K., and Lanckriet, G. (2017). Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Computing*, 16(4):62–74.