Seleção de Clientes Adaptativa baseada em Privacidade Diferencial para Aprendizado Federado

Vinícius R. M. Alves[1], Joahannes B. D. da Costa[1], Luis F. G. Gonzalez[1], Allan M. de Souza[1], Leandro A. Villas[1]

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

v250562@dac.unicamp.br, {jbdc, allanms, gonzalez, lvillas}@unicamp.br

Abstract. *Federated Learning (FL) is a distributed technique to training machine learning models, where data is processed locally and only local parameters are shared with an aggregation server. Despite client's data being kept locally, it's still possible for an adversary to conduct a model reconstruction attacks, for example. Therefore, this work presents PEGASUS, which leverages the guarantees of Differential Privacy (DP) to mitigate adversarial attacks in the FL environment. Additionally, PEGASUS employs a client selection strategy that dynamically adapts the number of devices training the model with the aim of dealing with the increasing loss of privacy (ϵ parameter of DP) over the course of communication rounds. Experimental evaluations show that PEGASUS significantly reduces the privacy loss (58%) of participating clients in training and maintains high levels of accuracy (97%).*

Resumo. *O Federated Learning (FL) é uma técnica distribuída para treinamento de modelos de aprendizado de máquina, em que os dados são processados localmente e apenas parâmetros locais são compartilhados com um servidor de agregação. Apesar dos dados dos clientes serem mantidos localmente, ainda é possível um adversário fazer um ataque de reconstrução de modelo, por exemplo. Sendo assim, este trabalho apresenta o PEGASUS que utiliza das garantias da privacidade diferencial (Differential Privacy (DP)) para mitigar ataques adversários no ambiente de FL. Além disso, o PEGASUS emprega uma estratégia de seleção de clientes que adapta dinamicamente a quantidade de dispositivos que treinam o modelo com o objetivo de lidar com o acréscimo da perda de privacidade (parâmetro ϵ da DP) ao decorrer das rodadas de comunicação. Avaliações experimentais mostram que PEGASUS reduz significativamente a perda de privacidade (58%) dos clientes participantes do treinamento e mantém bons níveis de acurácia (97%).*

1. Introdução

Nos últimos anos, o número de dispositivos conectados à Internet vem crescendo continuamente. Esses dispositivos possuem sensores avançados capazes de coletar inúmeros dados, tais como acelerômetro, luminosidade, posicionamento, giroscópio, dentre outros [Ficco et al. 2024]. Com isso, modelos de aprendizado de máquina, do inglês *Machine Learning (ML)*, podem ser utilizados para proposição de aplicações personalizadas para cada grupo de clientes. Na perspectiva tradicional, esses dados coletados por dispositivos móveis, por exemplo, são enviados e processados em um servidor central para obtenção de *insights* ou elaboração de modelos de inferência eficazes.

Por outro lado, coletar dados dos usuários permite que informações privadas sejam explanadas. Esses dados podem conter informações sensíveis, tais como a rotina do

usuário, o posicionamento geográfico e também o nome do portador dos dados. Para evitar que usuários se tornem vulneráveis a esse tipo de exposição, técnicas de aprendizado distribuído surgem como uma solução viável. O Aprendizado Federado, do inglês *Federated Learning (FL)*, é uma dessas técnicas promissoras onde cada usuário treina um modelo local com seus próprios dados locais e compartilha apenas os gradientes desse modelo [de Souza et al. 2024]. No entanto, apesar dos dados de treinamento continuarem nos dispositivos, o aprendizado distribuído está suscetível a ataques adversários, tais como: ataques de inversão de modelo, onde os adversários podem consultar o modelo treinado para resgatar o conjunto de dados de treinamento original [Ren et al. 2022]; discriminação da identidade do cliente através de redes generativas adversárias, onde o gerador recupera os dados privados do usuário.

Para contornar essa vulnerabilidade, existem diferentes abordagens na literatura que visam a proteção da privacidade, tais como: técnicas que produzem um dataset preservando sua privacidade (*k-anonymity*, *l-diversity* e *t-closeness*), técnicas para proteger a privacidade dos usuários durante o treinamento (*Secure Multiparty Computing* e *Homomorphic Encryption*) e a técnica de Privacidade Diferencial [Ouadrhiri and Abdelhadi 2022], do inglês Differential Privacy (DP). Sendo assim, este trabalho foca na técnica de DP, que se tornou um padrão da literatura para proteger a privacidade dos usuários. As técnicas de DP partem do princípio de adicionar ruído para proteger os dados sensíveis dos indivíduos. Porém, utilizar a DP no cenário de FL traz diversos desafios, incluindo: *(i)* custo alto de comunicação entre os dispositivos e o servidor de agregação; *(ii) trade-off* entre acurácia e privacidade; e *(iii)* acréscimo do custo de privacidade a cada rodada de treinamento.

Nesse contexto, este trabalho apresenta um mecanismo que aplica **P**rivacidad**E** diferencial para miti**G**ar a perd**A** de informação **S**ensível de us**U**ário**S** no aprendizado federado, chamado **PEGASUS**. O PEGASUS é uma solução que lida com acréscimo acelerado da perda de privacidade a cada rodada de treinamento. Sendo assim, o PEGASUS seleciona os clientes de forma adaptativa baseado na privacidade do cliente para evitar uma alta perda de privacidade acumulada ao final do treinamento. Os experimentos feitos mostraram que o PEGASUS é apto à reduzir em até 58% a perda de privacidade de todos clientes comparado à solução base da literatura. Além disso, o PEGASUS atinge altos níveis de acurácia do modelo treinado, cerca de 97%, mesmo na presença de DP, por conta da estratégia de seleção de clientes empregada.

Este trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta os principais conceitos para entendimento do trabalho. A Seção 4 apresenta o PEGASUS em detalhes. A Seção 5 descreve a metodologia de avaliação e discute os resultados. Por fim, a Seção 6 apresenta as conclusões.

2. Trabalhos Relacionados

Considerando a privacidade diferencial aplicada ao cenário de FL, esta seção descreve alguns dos trabalhos relacionados e suas respectivas estratégias empregadas. Por exemplo, Talaei & Izadi [Talaei and Izadi 2024] apresentam um *framework* de DP que adiciona ruído gaussiano de forma adaptativa para modelos de aprendizado profundo (*deep learning* – DL) no ambiente de FL. Esse algoritmo é baseado na importância das características e parâmetros do modelo. Sendo que adicionar ruído em parâmetros menos importantes não afeta a precisão do mesmo modo que quando adicionado em parâmetros importantes. Portanto, esse *framework* lida com o *trade-off* entre a precisão do modelo e

a adição de ruído, se preocupando no equilíbrio entre precisão e privacidade.

Fu *et al.* [Fu et al. 2022] propôem o Adap DP-FL, um algoritmo diferencialmente privado para o FL com um recorte adaptativo de gradientes e uma redução adaptativa na escala do ruído adicionado. Os gradientes são recortados de forma adaptativa por conta da heterogeneidade das magnitudes de parâmetros ao decorrer das rodadas de treinamento. Já o ruído é gradualmente reduzido ao decorrer das rodadas, devido à convergência de gradiente. Assim, observando o valor da função de perda, o Adap DP-FL lida com a degradação da utilidade do modelo adaptando os parâmetros.

Ling *et al.* [Ling et al. 2024] apresentam o ALI-DPFL, uma abordagem do FL diferencialmente privado que realiza iterações locais adaptativas. Analisando a convergência de forma teórica, o algoritmo encontra um número ótimo de iterações locais do Gradiente Estocástico Descendente Diferencialmente Privado (DPSGD) para os clientes entre duas atualizações globais sequenciais. Assim, o ALI-DPFL é restrito tanto para o orçamento de privacidade quanto para rodadas de comunicação.

de Souza *et al.* [de Souza et al. 2023] introduzem o DEEV, um algoritmo no ambiente de FL em que os clientes são selecionados a partir do desempenho do modelo. Assim, os clientes que possuem uma acurácia menor são escolhidos, junto a um mecanismo de redução do número de clientes, o que diminui o número de clientes participando a cada rodada de comunicação. O DEEV reduz o *overhead* de comunicação e processamento gerado pelos processos envolvidos no FL. A Tabela 1 resume os trabalhos relacionados, destacando suas principais características e comparando-os com a solução proposta.

Tabela 1. Trabalhos relacionados com suas principais funcionalidades.

Trabalho	Funcionalidade		
	Privacidade Diferencial	Seleção de Clientes	Redução do ϵ
Talaei & Izadi 2024	✓	✗	✗
Fu *et al.* 2022	✓	✗	✗
Ling *et al.* 2024	✓	✗	✓
de Souza *et al.* 2023	✗	✓	✗
PEGASUS	✓	✓	✓

3. Conceitos Básicos

Esta seção traz um material de apoio que introduz alguns conceitos básicos, como a privacidade diferencial e o aprendizado federado.

3.1. Privacidade Diferencial

A privacidade diferencial, do inglês *Differential Privacy (DP)*, foi introduzida em 2006 [Dwork et al. 2006] com o objetivo de permitir o estudo de um conjunto de dados sem revelar informações individuais de cada indivíduo. Assim, a DP aplica ruído ao conjunto de dados original de modo que um adversário não possa inferir se um indivíduo específico faz parte do conjunto de dados ou não.

Definição 1 $((\epsilon, \delta) - DP)$**.** *um mecanismo M é chamado (ϵ, δ)-diferencialmente privado se para todos os conjuntos de dados vizinhos D, $D' \in D^n$ [Dwork et al. 2006]. Para*

todos conjuntos $S \subseteq Y$, onde Y é o conjunto de todas as possíveis saídas, se tem:

$$Pr[M(D) \in S] \leq e^{\epsilon} Pr[M(D') \in S] + \delta \quad (1)$$

Isso significa que quando o mecanismo M é aplicado a D, sua saída é semelhante à saída quando M é aplicado a D'. Assim, $(\epsilon, \delta) - DP$ fornece um critério forte na proteção de privacidade para sistemas distribuídos [Wei et al. 2020]. Sendo que o parâmetro $\epsilon > 0$ denota o limite distinguível de todas as saídas do conjunto de dados vizinhos D, D' em um *dataset*. Já o parâmetro δ foi adicionado para permitir uma pequena quantidade de "vazamento" de informações, deixando o mecanismo de DP menos robusto e aumentando sua utilidade. Assim, com um δ dado, um mecanismo $(\epsilon, \delta) - DP$ com um ϵ maior, proporciona uma diversidade maior dos conjuntos de dados vizinhos D e D_i, aumentando o risco de violação da privacidade. Devido a isso, chama-se o parâmetro ϵ de perda de privacidade, orçamento de privacidade ou vazamento de privacidade.

3.2. Aprendizado Federado

O FL é uma técnica que permite que dispositivos de borda treinem colaborativamente modelos de ML de forma distribuída. Para isso, cada dispositivo (cliente) treina um modelo utilizando seus dados locais. Posteriormente, esse modelo é compartilhado com um servidor que é responsável em agregar esses modelos recebidos dos clientes, formando um único modelo global. Por fim, o servidor de agregação compartilha o modelo global com os clientes, finalizando uma rodada de comunicação, a qual será repetida até que o modelo convirja. A primeira abordagem de FL é conhecida como FedAvg, onde a agregação dos modelos é realizada através de uma média ponderada dos modelos dos clientes participantes [McMahan et al. 2017].

O FL permite que os dispositivos distribuídos treinem de forma colaborativa um modelo global, mantendo os dados locais de treinamento no dispositivo e enviando para o servidor somente o modelo treinado localmente [de Souza et al. 2024]. Por definição, considera-se um sistema de FL que consiste em um servidor e *N* clientes. Sendo D_k um conjunto de dados local do cliente C_k, onde $k \in \{1, 2, \ldots, N\}$. No servidor o objetivo é aprender um modelo sobre os dados dos *N* clientes participantes. Cada cliente que participa do treinamento local, necessita encontrar um vetor w de um modelo de ML para minimizar certa função de perda.

4. PEGASUS

PEGASUS foi posposto para combater a vulnerabilidade do usuário de possíveis ataques adversários, utilizando da privacidade diferencial que fornece garantias matemáticas de privacidade. Além disso, o PEGASUS lida com o alto acréscimo do orçamento de privacidade quando se utiliza a DP no ambiente federado. Para isso, o PEGASUS faz uma seleção adaptativa de clientes de acordo com o seu orçamento de privacidade para eleger quais dispositivos devem realizar o treinamento na próxima rodada de comunicação e, assim, evitar que os usuários tenham uma alta perda de privacidade ao decorrer da rodadas.

4.1. Definição do Problema

Para obtenção de dados numéricos, um mecanismo gaussiano pode ser utilizado para garantir $(\epsilon, \delta) - DP$. Com isso, o mecanismo de DP que adiciona ruídos gaussianos artificiais é introduzido [Dwork and Roth 2014]. Para garantir que a distribuição de ruído

fornecida $n \sim \mathcal{N}(0, \sigma^2)$ preserve $(\epsilon, \delta) - DP$, onde $\mathcal{N}$ é a distribuição de gauss, seleciona-se uma escala de ruído $\sigma \geq \Delta s/\epsilon$ e a constante $c \geq \sqrt{2ln(1.25/\delta)}$ para $\epsilon \in (0, 1)$. Uma vez que n é o valor de uma amostra de ruído aditivo para um dado conjunto de dados, Δs é a sensibilidade da função s dada por $\Delta s = max_{D_k, D'_k} \parallel s(D_k - s(D'_k) \parallel$, e s é uma função de valor real. Também se utiliza uma técnica de recorte, para garantir que $\parallel \mathbf{w}_k \parallel \leq C$, onde $\mathbf{w}_k$ denota o vetor de parâmetros locais sem pertubações do cliente k e C é um limite de recorte para limitar $\mathbf{w}_k$.

Nesse contexto, o algoritmo *Noising before Aggregation FL (NbAFL)* treina um modelo $(\epsilon, \delta) - DP$ [Wei et al. 2020]. De início, o servidor transmite aos clientes os parâmetros de nível de privacidade necessários (ϵ, δ) e os parâmetros globais iniciais $\mathbf{w}^{(0)}$. Na agregação t, N clientes ativos irão treinar os parâmetros utilizando o *dataset* local. Após a conclusão do treinamento local, o cliente k adicionará ruído aos seus parâmetros treinados $\mathbf{w}_k^{(t)}$ e os enviará ao servidor de agregação $\tilde{\mathbf{w}}_k^{(t)}$. Assim, o servidor atualiza os parâmetros globais $\mathbf{w}^{(t)}$ com base nos parâmetros locais recebidos dos clientes. Ruídos aditivos $\mathbf{n}_D^{(t)}$ são adicionados aos parâmetros globais $\mathbf{w}^t$ antes de serem transmitidos aos clientes. Finalmente, cada cliente estima a precisão e inicia a próxima rodada de treinamento com esses novos parâmetros recebidos $\tilde{\mathbf{w}}_k^{(T)}$. O procedimento de FL é completado quando se atinge o número predefinido T de agregação.

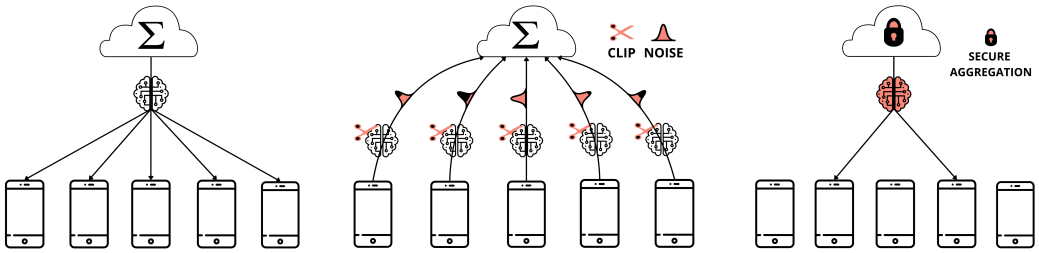
Assim como o FedAvg, o NbAFL divide o treinamento em rodadas de comunicação e seleciona um conjunto de clientes aleatoriamente para participar do treinamento. Porém, ao decorrer das rodadas de treinamento, o orçamento de privacidade (ϵ) para cada cliente $C_k \in C_1, C_2, ..., C_N$ se acumula até a rodada T. Assim, aqueles clientes que já possuem um orçamento de privacidade alto, continuam *perdendo* sua privacidade.

4.2. Seleção de Clientes baseada no Orçamento de Privacidade

O treinamento de modelos por clientes com orçamento de privacidade (ϵ) limitado permite que o acréscimo do ϵ seja restringido. Assim, clientes que já possuem um orçamento de privacidade (ϵ) alto esperem para que clientes com orçamento de privacidade baixo treinem seus modelos e, consequentemente, seus orçamentos de privacidade (ϵ) se aproximem dos demais. Em outros termos, tal abordagem permite que clientes com alto vazamento de privacidade não continuem perdendo sua privacidade ao decorrer das rodadas de comunicação. Nesse contexto, cada cliente deve compartilhar (com o servidor) seu modelo e uma métrica que seja possível mensurar seu orçamento de privacidade. No caso da privacidade diferencial, cada cliente compartilha seu ϵ atual. Assim, o servidor sabe quais clientes estão com seus dados *mais privados* ou *menos privados*, podendo assim aplicar suas estratégias de seleção com base no orçamento de privacidade de cada cliente.

A Figura 1 descreve o processo de seleção de clientes aplicado pelo PEGASUS. Como pode ser notado na Figura 1(a), de início o servidor envia o modelo para todos os clientes. Posteriormente, os clientes realizam o treinamento do modelo com seus dados locais, recortam os parâmetros locais (*clip*), adicionam ruído nesses parâmetros (*noise*) e compartilham o modelo com o servidor e também compartilham o orçamento de privacidade (ϵ) atual obtido após o treinamento, descrito na Figura 1(b). Com isso, na Figura 1, o servidor recebe os modelos dos clientes, realiza a agregação e faz uma média dos orçamentos de privacidade (ϵ). Por último, baseado no orçamento de privacidade, PEGASUS seleciona os clientes que estão abaixo do orçamento de privacidade médio dos clientes. A seleção de clientes adotada pelo PEGASUS reduz a perda de privacidade (ϵ) de todos clientes durante as rodadas de comunicação, pois menos clientes compartilham

seus parâmetros a cada rodada.



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

Figura 1. Visão geral do PEGASUS.

5. Avaliação de Desempenho

Esta seção descreve a metodologia e as métricas utilizadas para avaliar a eficiência do PEGASUS em comparação com outra abordagem do estado da arte. Além disso, apresenta e discute os resultados obtidos.

5.1. Metodologia

O PEGASUS foi implementado no *framework* Flower [Beutel et al. 2022], na versão 1.7.0, e considerando o TensorFlow e TensorFlow Privacy, nas versões 2.14.1 e 0.8.10, respectivamente. No Flower, a comunicação entre os clientes e servidor é estabelecida por meio de *Google Remote Procedure Call (gRPC)*. Além disso, considerou-se o conjunto de dados composto por imagens de dígitos manuscritos (MNIST) [LeCun et al. 2010], por ser um *benchmark* bastante conhecido e utilizado pela comunidade de ML e FL.

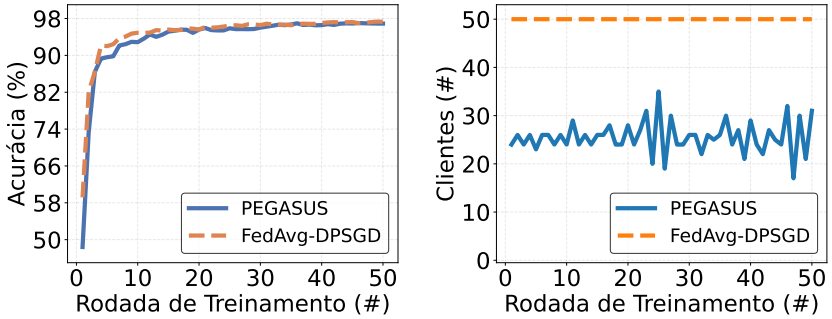
Para o ajuste do modelo foram utilizados os seguintes parâmetros: o *Differentially Private Stochastic Gradient Descent (DPSGD)* como mecanismo para atualizar os pesos do modelo e o *Sparse Categorical Crossentropy (SCC)* como função de perda. Agora, para calcular a garantia de privacidade diferencial $(\epsilon, \delta) - DP$, foi utilizada uma implementação do contador *Rényi Differential Privacy (RDP)* [McMahan et al. 2019].

Para analisar o desempenho do modelo e também o quão privado estão os dados de cada cliente, as seguintes métricas foram utilizadas: *(i) Acurácia do modelo distribuído:* mede a convergência da solução, sendo calculada após cada rodada de comunicação. Quanto menos rodadas forem necessárias para atingir uma certa acurácia desejada, mais rápida será a convergência da solução; e *(ii) Orçamento da privacidade* ϵ*:* mede o quanto cada cliente está "perdendo" sua privacidade. Quanto mais rodadas que o cliente participa do treinamento, maior será seu vazamento de privacidade e, consequentemente, seus dados estarão menos privados.

5.2. Resultados

Esta seção apresenta e discute os resultados obtidos, dando atenção especialmente à redução da perda de privacidade (ϵ) do PEGASUS. Por outro lado, foi definida a solução FedAvg como *baseline* para a comparação realizada, e assim, definimos que 100% dos clientes serão selecionados a cada rodada. Além disso, também da mesma forma que em PEGASUS, utilizamos o otimizador DPSGD para aplicar a privacidade diferencial e rastrear o orçamento de privacidade (ϵ) durante o processo de treinamento do modelo federado. A abordagem de comparação é chamada FedAvg-DPSGD neste trabalho.

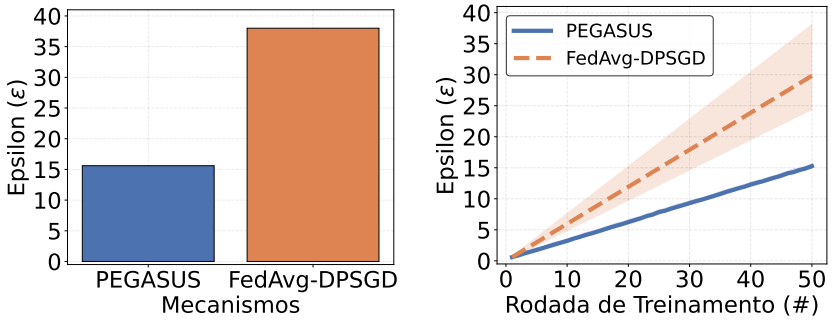
A Figura 2 exibe os resultados de acurácia e clientes selecionados a cada rodada de treinamento. Ambas as soluções apresentam níveis de acurácia semelhantes para um cenário com 50 clientes e 50 rodadas, conforme mostra a Figura 2(a). Além disso, enquanto a solução FedAvg-DPSGD seleciona 100% dos clientes a cada rodada, o PEGASUS seleciona os clientes adaptativamente, sem definir a priori a quantidade de clientes selecionados (Figura 2(b)), pois apenas clientes com um orçamento de privacidade baixo serão selecionados por rodada.



(a) Acurácia do modelo. (b) Clientes selecionados.

Figura 2. Acurácia do modelo e clientes selecionados por rodada de treinamento.

A Figura 3(a) mostra o orçamento de privacidade (ϵ) final considerando todos os clientes envolvidos no treinamento, onde o PEGASUS diminuí em até 58% o orçamento de privacidade (ϵ) final, garantindo assim maior privacidade nos dados de todos clientes participantes do treinamento, para o mesmo nível de acurácia. Já a Figura 3(b) rastreia a perda de privacidade de cada cliente nas duas soluções comparativas, mostrando a média do orçamento de privacidade (ϵ) de todos os clientes envolvidos. Enquanto o FedAvg-DPSGD tem uma curva com um comportamento linear, pois todos clientes acumulam sua perda de privacidade (ϵ) nas rodadas de comunicação, PEGASUS apresenta uma curva em escada (para cada cliente), que representa o *atraso* no aumento da perda de privacidade (ϵ) devido a seleção de clientes do PEGASUS. Note que um cliente que não participa da rodada de comunicação, não expõe seus parâmetros locais, tendo $\epsilon = 0$ naquela rodada.



(a) Orçamento de Privacidade Final (b) Rastreamento do parâmetro ϵ

Figura 3. Comparação entre os orçamentos de privacidade das soluções.

6. Conclusão

Este trabalho apresentou o PEGASUS, uma solução de aprendizado federado que utiliza da privacidade diferencial para evitar ataques adversários, protegendo assim os dados vulneráveis dos usuários. PEGASUS também implementa uma seleção adaptativa de clientes

baseada no orçamento de privacidade para reduzir a quantidade de clientes selecionados a cada rodada de comunicação, reduzindo a perda de privacidade de todos clientes no treinamento. Os resultados mostraram que PEGASUS reduz em até 58% a perda de privacidade dos clientes com uma acurácia semelhante à solução de comparação. Para trabalhos futuros, serão incorporados estratégias que alterem de forma dinâmica os ruídos que serão adicionados aos parâmetros dos clientes. Além disso, o recorte dos parâmetros também será ajustado de forma dinâmica.

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].

Referências

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K. H., Parcollet, T., de Gusmão, P. P. B., and Lane, N. D. (2022). Flower: A friendly federated learning research framework.

de Souza, A. M., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14. SBC.

de Souza, A. M., Maciel, F., da Costa, J. B. D., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2024). Adaptive client selection with personalization for communication efficient federated learning. *Ad Hoc Networks*, page 103462.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. *Proc. 24th Annu. Int. Conf. The Theory Appl. Cryptography*, pages 486–503.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9:221–407.

Ficco, M., Guerriero, A., Milite, E., Palmieri, F., Pietrantuono, R., and Russo, S. (2024). Federated learning for iot devices: Enhancing tinyml with on-board training. *Information Fusion*, 104:102189.

Fu, J., Chen, Z., and Han, X. (2022). Adap dp-fl: Differentially private federated learning with adaptive noise.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Ling, X., Fu, J., Wang, K., Liu, H., and Chen, Z. (2024). Ali-dpfl: Differentially private federated learning with adaptive local iterations.

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.

McMahan, H. B., Andrew, G., Erlingsson, U., Chien, S., Mironov, I., Papernot, N., and Kairouz, P. (2019). A general approach to adding differential privacy to iterative training procedures.

Ouadrhiri, A. E. and Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380.

Ren, H., Deng, J., and Xie, X. (2022). Grnn: Generative regression neural network - a data leakage attack for federated learning. *ACM Transactions on Intelligent and Technology*, 1.

Talaei, M. and Izadi, I. (2024). Adaptive differential privacy in federated learning: A priority-based approach.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., and Vincent Poor, H. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469.