Mitigação de Ataques "*Label-Flipping*" no Aprendizado Federado: Experimentos e Estratégias de Seleção de Clientes

João Pedro C. Batista, Eduardo M. M. Sarmento, Johann J. S. Bastos
Vinícius F. S. Mota, Rodolfo S. Villaça

[1]Departamento de Informática - Universidade Federal do Espírito Santo (UFES)
Vitória/ES - Brasil

{joao.c.batista, eduardo.sarmento, johann.bastos}@edu.ufes.br
{vinicius.mota, rodolfo.villaca}@inf.ufes.br

Abstract. *This article explores challenges affecting model efficacy in Federated Learning, particularly due to malicious clients engaging in attacks like "label-flipping". Through experiments in the MininetFed environment, it assesses the influence of these clients and the effectiveness of different client selection strategies and clustering algorithms in mitigating such specific attacks. The findings provide crucial insights for enhancing training process security and effectively safeguarding models in Federated Learning against internal threats.*

Resumo. *Este artigo investiga os desafios que afetam a eficácia dos modelos no contexto do Aprendizado Federado, especialmente devido à presença de clientes maliciosos que realizam ataques como o label-flipping. Utilizando o ambiente MininetFed, são conduzidos experimentos detalhados para avaliar o impacto desses clientes e a eficácia de diversas estratégias de seleção e algoritmos de clusterização na mitigação desses ataques específicos. Os resultados obtidos fornecem insights fundamentais para fortalecer a segurança do processo de treinamento e proteger adequadamente os modelos no Aprendizado Federado contra ameaças internas.*

1. Introdução

O aprendizado de máquina é um conceito cada vez mais difundido no mundo contemporâneo, e à medida que os conjuntos de dados para treinamento de modelos aumentam, exige-se mais recursos computacionais para realizar esses processos de aprendizagem. No aprendizado distribuído, diferentes dispositivos conectados em uma rede realizam tarefas de treinamento, coordenadas por um servidor. Entretanto, a coleta de dados nos dispositivos na borda de rede, ou seja, próximos dos proprietários dos dados, gera um grave problema de privacidade, uma vez que está suscetível ao vazamento de informações sensíveis ou confidenciais [Alves et al. 2024].

Para implementar melhorias neste processo, surge o conceito de Aprendizado Federado, do inglês, *Federated Learning* (FL). O FL pode ser definido como uma técnica de aprendizagem colaborativa em que apenas os parâmetros dos modelos, treinados somente com os dados dos usuários locais, são compartilhados e agregados por um servidor centralizado. No FL não há compartilhamento de dados dos dispositivos de borda com o servidor de agregação [Mammen 2021]. Essa técnica de aprendizado de máquina possui

primariamente quatro passos [Mammen 2021]: seleção de clientes, treinamento de modelos locais, agregação e compartilhamento do modelo global. Esses passos propiciam um ambiente em que não há compartilhamento de dados, de modo a garantir um certo grau de privacidade.

Entretanto, apesar dos dados de treinamento continuarem nos dispositivos, sem compartilhamento com dispositivos externos, o processo de aprendizado continua vulnerável a ataques por clientes maliciosos. Esses ataques podem ser divididos em quatro grandes grupos [Mammen 2021]: ataques de inferência, envenenamento de modelos, envenenamento de dados e ataques *backdoor*, em que dispositivos ou grupos de dispositivos burlam processos e medidas de segurança para introduzir funcionalidades maliciosas ao modelo global.

O presente artigo aborda o segundo grupo, em especial um representante denominado *label-flipping*, e tem como objetivo implementar e avaliar técnicas de detecção de agentes maliciosos, bem como analisar métodos de seleção de clientes, de modo a mitigar a interferência externa no processo de treinamento e os danos ao procedimento de convergência dos modelos gerados.

O restante do artigo está organizado em 6 seções, apresentadas a seguir. Na Seção 2 é realizada uma breve revisão da literatura, focando nas técnicas de detecção de treinadores maliciosos no Aprendizado Federado. Em seguida, a Seção 3 define alguns algoritmos importantes usados na seleção de clientes maliciosos, suportados pelo Mininet-Fed [Bastos et al. 2024], ferramenta usada no apoio à execução deste trabalho; a Seção 4 define formalmente o ataque de label-flipping, e apresenta os resultados dos experimentos realizados para avaliar o impacto deste tipo de ataque no aprendizado. A Seção 5 apresenta uma técnica simples e inicial para detectar estes clientes maliciosos. Finalmente, a Seção 6 traz a conclusão deste artigo e propostas de trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção apresentamos alguns trabalhos relacionados que lidam com os impactos dos ataques de *label-flipping* em aprendizado federado, bem como algoritmos e soluções para mitigar este cenário.

Tolpegin *et al.* [Tolpegin et al. 2020] mostram que esta categoria de ataque pode causar a queda tanto de acurácia quanto de *recall* de modelos de classificação. Este efeito foi observado pelos pesquisadores mesmo quando existem poucos participantes maliciosos no processo, mas este ataque tem efeitos significativos, causando uma queda de mais de 6% na maioria dos casos, apenas nas classes que foram trocadas no conjunto de treinamento, as outras classes tendo no máximo 0,34% de queda nessas mesmas métricas. Os pesquisadores então propõem um algoritmo baseado em *Principal Component Analysis* (PCA) para separar os clientes maliciosos dos honestos, utilizando o fato de que as atualizações dos paramêtros enviadas por clientes maliciosos tem caracteristicas únicas quando comparados com as atualizações de clientes honestos. Fazendo uso deste algoritmo os pesquisadores foram capazes de separar os dois grupos de clientes mesmo quando 20% de todos os clientes eram maliciosos, mitigando a queda de qualidade do modelo global.

Li *et al.* [Li et al. 2021] propõem uma melhoria ao algoritmo de Tolpegin *et al.*, fazendo o uso da técnica de redução de dimensionalidade *Kernel Principal Component*

Analysis (KPCA) no lugar do PCA, somado do algoritmo de clusterização *K-Means*. Em experimentos com os conjuntos de dados CIFAR-10 e Fashion-MNIST os autores mostram que esta melhoria consegue resultados melhores do que o algoritmo proposto por Tolpegin *et al.* em todas as porcentagens de clientes maliciosos consideradas, sendo até mesmo capaz de previnir completamente a queda de qualidade do modelo global por ataques de *label-flipping* até quando 4% dos clientes eram maliciosos. Porém, estes experimentos também mostraram que o uso do algoritmo *K-Means* tem pouco impacto sobre o resultado final considerando o custo computacional necessário para sua execução, sendo assim os autores indicam que o uso de KPCA ja é suficiente.

Em [Jebreel et al. 2022a] é proposta uma abordagem contra os ataques *label-flipping*, que usa os gradientes dos *updates* dos modelos para diferenciar clientes honestos e maliciosos. Os nós com os maiores gradientes em módulo da camada de saída identificam os clientes maliciosos. Para dados *iid* é usado o *K-Means* para separar esses gradientes em dois grupos. Para dados *non-iid* os autores o HDBSCAN para gerar múltiplos clusters, onde os dois clusters mais próximos representam os clientes que possuem a classe atacada. Nos testes, foi usado os datasets MNIST e o CIFAR, e no caso *iid*, a aborgagen conseguiu lidar bem com esse tipo ataque quando existem até 50% de clientes maliciosos. Em cenários *non-iid*, todas as abordagens do estado da arte tiveram uma degradação da acurária muito grande com mais de 20% de clientes maliciosos. A aborgagem proposta manteve-se robusta com até 50% dos clientes maliciosos em ambos os cenários, *iid* e *non-iid*.

Em [Jiang et al. 2023] é proposto o algoritmo MCDFL *(Malicious Clients Detection in Federated Learning*). Nesse algoritmo, um gerador de dados é treinado a partir do modelo global. Esse gerador é enviado aos clientes, onde é usado para estimar a qualidade dos dados do cliente. O valor da qualidade dos dados de cada cliente é devolvido ao servidor e separado em dois grupos usando *K-Means*. O grupo com menor qualidade média é considerado malicioso e é excluído da seleção de clientes do próximo *round*. O algoritmo foi comparado com o FedAvg nos datasets Fashion-MNIST e CIFAR10, alterando a quantidade de clientes maliciosos entre 5% e 40%. A evolução da acurácia se manteve praticamente a mesma para o MCDFL em todos os testes, enquanto o FedAvg apresentou uma acurácia já abaixo com 5% e um declínio leve à medida que a porcentagem foi aumentando. A diferença chega a ser de 20 p.p. no CIFAR10 e 30 p.p. no Fashion-MNIST.

Para o objetivo do presente artigo, as técnicas que foram empregadas para a sua obtenção já existem e foram implementadas a partir dos trabalhos relacionados que aqui foram citados. Nesse sentido, o diferencial proposto por esse documento é avaliação do impacto dos ataques *label-flipping* a partir do uso de um ambiente de emulação de Aprendizado Federado, no caso, o MininetFed [Bastos et al. 2024].

3. Seleção de Clientes

Uma das etapas do Aprendizado Federado é a seleção dos clientes que irão compor as rodadas de treinamento. A escolha do algoritmo de seleção de clientes é uma etapa fundamental do FL, pois, dentre outras funções, permite a detecção e exclusão de clientes maliciosos do processo de treinamento. Alguns possíveis algoritmos de seleção de clientes são:

- ***All***: Todos os clientes são selecionados para o processo de treinamento, sem distinção entre clientes honestos e maliciosos;
- ***Random***: Há uma escolha aleatória de clientes para o processo de treinamento;
- ***Deev***: Proposto em [de Souza et al. 2023], é um algoritmo que seleciona os k clientes com acurácia do modelo local menor do que a acurácia média de todos os clientes, em que k decresce de acordo com uma função de decaimento;
- ***FedSecPer***: Possui a mesma ideia e objetivos da função de seleção *Deev*, mas sem o decrescimento de k;
- ***B-Trimmed Mean***: Tem como critério de seleção a escolha de clientes que tenham uma acurácia do modelo local maior do que a média truncada da acurácia global [Wang et al. 2022].

4. O Ataque *Label-Flipping* e o seu impacto no Aprendizado Federado

O *label-flipping* é um exemplo de ataque de envenenamento de modelos em que os clientes maliciosos invertem os rótulos de uma determinada classe para outra classe [Jebreel et al. 2022b]. O conceito desse tipo de ataque é envenenar o modelo global (e consequentemente os modelos locais pelo envio de parâmetros evenenados pelo servidor) a partir da introdução de erros no modelo local. Assim, quando os clientes recebem os parâmetros envenenados do servidor central, os parâmetros dos modelos locais são fortemente afetados, o que diminui posteriormente a acurácia do modelo global e prejudica o processo de convergência [Mammen 2021].

Um exemplo pode ser construído utilizando-se o *dataset* MNIST, que contém diversas imagens de dígitos manuscritos e rotulados conforme o valor que representam [Bastos et al. 2024]. Assim, fazendo uso desse *dataset*, os clientes maliciosos invertem os rótulos que deveriam ser ”0”para o valor ”1”, por exemplo, envenenando os modelos globais conforme processo supracitado. A Fig. 1 ilustra esse procedimento do *label-flipping*, explicitando o envio de informações envenenadas para o servidor.

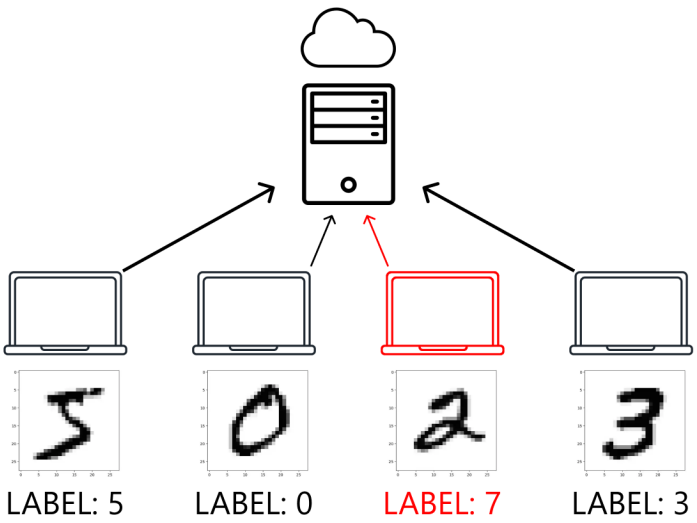


Figura 1. Exemplo de ataque *label-flipping* utilizando-se o *dataset* MNIST

Visando reproduzir o ataque *label-flipping* e analisar a influência dos cli-

entes maliciosos no processo de treinamento, utilizou-se da ferramenta MininetFed [Bastos et al. 2024]. O MininetFed é uma solução que permite emular um ambiente de execução de experimentos de FL, permitindo que o usuário possa definir parâmetros de execução dos dispositivos simulados, bem como a sua customização e aplicação a diferentes conjuntos de dados, funções de seleção de clientes e funções de agregação. Dois experimentos foram realizados para avaliar a atuação destes agentes mal-intencionados, descritos abaixo. Para ambos os experimentos, a quantidade de clientes foi escolhida levando em consideração a limitação do hardware de experimentação e para destacar a proporção de clientes maliciosos nos experimentos. Em especial, no segundo experimento foi utilizada uma quantidade menor de clientes para avaliar a capacidade das funções de seleção em cenários com alta influência de agentes maliciosos (25% dos clientes).

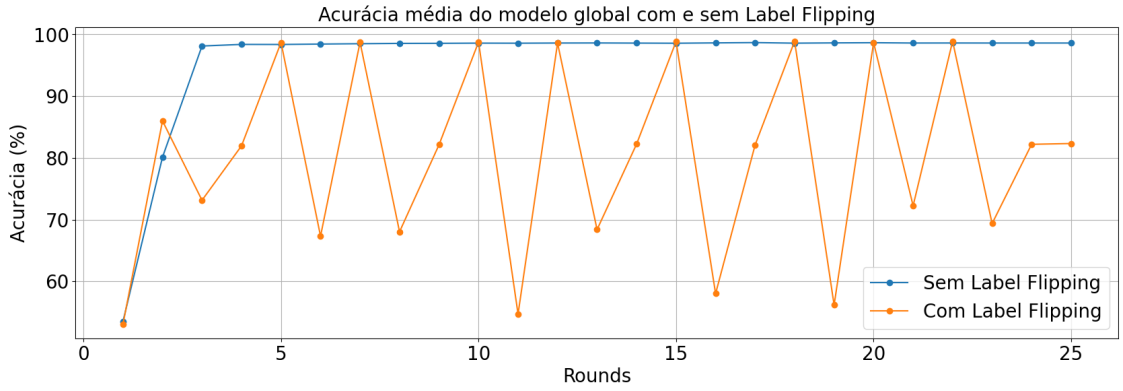
- **Experimento 1 - Avaliação da acurácia do modelo global**: Pretende avaliar o efeito de clientes que efetuem o ataque *label-flipping* sobre a acurácia do modelo global. Foram avaliados dois cenários de aprendizado com o *dataset* MNIST com seis clientes: i) seis clientes honestos e nenhum malicioso, todos com 100% da capacidade de processamento; e ii) cinco clientes honestos e um malicioso, todos com 100% da capacidade de processamento. A função de agregação utilizada foi a *FedAvg* e a função de seleção de clientes utilizada foi a *Random*.
- **Experimento 2 - Avaliação das funções de seleção de clientes**: Tem como objetivo avaliar a eficácia das funções de seleção de clientes na mitigação do efeito de ataques *label-flipping* sobre a acurácia do modelo global. Foram avaliados cinco cenários de aprendizado com o *dataset* MNIST com quatro clientes, sendo três honestos e um malicioso, todos com 100% da capacidade de processamento. Cada cenário utilizou uma função de seleção de clientes diferente. Foram elas: i) função *All*; ii) função *Random*; iii) função *Deev*; iv) função *FedSecPer*; e v) função *B-Trimmed Mean*. A função de agregação utilizada foi a *FedAvg*.

Nos dois experimentos, o agente malicioso realizou o *flip* dos rótulos de todas as classes. Todas as classes diferentes de zero foram rotuladas como zero e a classe zero foi rotulada como um (*dataset* MNIST). Assim, a totalidade dos dados de treinamento utilizados pelo cliente atacante foi envenenada.

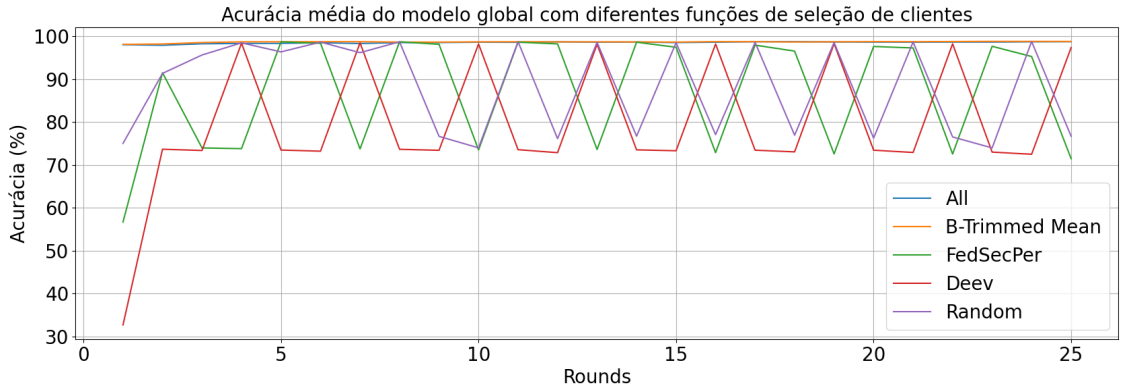
A Fig. 2(a) mostra a acurácia média do modelo global durante as rodadas de treinamento do primeiro experimento, evidenciando o efeito negativo da presença de um treinador malicioso, que comprometeu a convergência dos modelos. No gráfico, percebe-se que a acurácia média estabilizou-se acima de 98% após a terceira rodada de treinamento, no contexto sem clientes maliciosos. Com a presença desses agentes, a acurácia variou durante todo o treinamento. Na Fig. 2(b) nota-se que as funções de seleção *FedSecPer*, *Deev* e *Random* não obtiveram bons desempenhos para a mitigação do ataque. Ademais, a função *B-Trimmed Mean* obteve desempenho similar à função *All*, o que possivelmente pode indicar que, embora destacada pela quantidade de clientes presentes, a proporção de clientes maliciosos não foi suficiente para afetar o desempenho do treinamento. Assim, sugere-se que essa possível causa seja investigada em trabalhos futuros.

5. Detecção por Clusterização

Conforme citado na Seção 2, uma possível técnica para a detecção de clientes maliciosos no processo de FL é a aplicação do algoritmo de clusterização *K-Means* [Li et al. 2021].



(a) Experimento 1



(b) Experimento 2

Figura 2. Acurácia média do modelo global nos Experimentos 1 (a) e 2 (b)

Nesse contexto, a clusterização dividiria os clientes participantes do treinamento em dois grupos: maliciosos e honestos. Essa divisão seria feita por meio dos dados enviados ao servidor a cada rodada.

O algoritmo *K-Means* define, de forma aleatória, características médias para cada grupo, sendo essas características ajustadas conforme a chegada de novos modelos e o avanço do processo de treinamento. A essas características médias dá-se o nome de centroides. Assim, baseado na semelhança dos modelos atualizados dos clientes com os centroides de cada *cluster* (grupo), é possível classificá-los visando detectar clientes anômalos.

Para averiguar o desempenho desse algoritmo de clusterização, propôs-se o Experimento 3, também realizado no MininetFed, cujas condições estão descritas a seguir:

- **Experimento 3 - Avaliação de desempenho da clusterização de clientes por *K-Means*:** Tem como objetivo dividir os clientes participantes em dois grupos, maliciosos por ataque *label-flipping*, e honestos, a cada rodada de treinamento. A avaliação dos grupos baseou-se nos pesos da camada de saída da rede, que contém 10 neurônios [Bastos et al. 2024]. Foi avaliado o seguinte cenário de treinamento com o *dataset* MNIST: i) um cliente malicioso e três clientes honestos, todos com 100% da capacidade de processamento. A função de seleção de clientes utilizada foi a *All*. Nota-se que neste experimento não há uma preocupação em excluir o

agente mal-intencionado do processo, apenas identificá-lo, devendo este participar de todas as rodadas. A função de agregação utilizada foi a *FedAvg*.

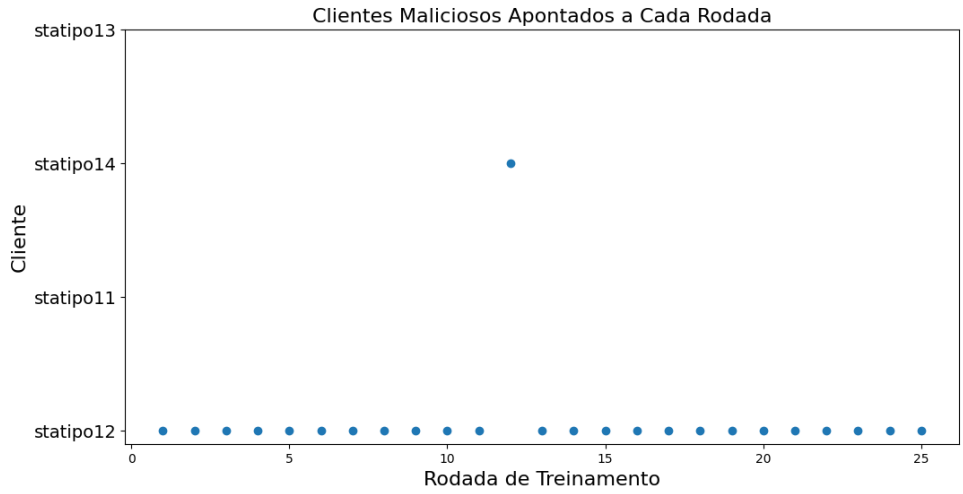


Figura 3. Resultado da clusterização de clientes em maliciosos e honestos a cada rodada de treinamento a partir do uso do algoritmo *K-Means*

A Fig. 3 demonstra os clientes apontados como maliciosos pelo algoritmo a cada rodada de treinamento. Nessa simulação, cada cliente recebeu um identificador, sendo os clientes identificados por *statipo11*, *statipo13* e *statipo14* honestos e o cliente identificado por *statipo12* malicioso.

A partir da Fig. 3, percebe-se que, nos experimentos realizados, o algoritmo foi capaz de identificar o cliente malicioso em 24 das 25 rodadas de treinamento, tendo classificado de forma errônea em apenas uma rodada, a décima segunda. Isso demonstra a capacidade do algoritmo K-Means possui, em um cenário inicial de exploração, de realizar a clusterização proposta e identificar os clientes maliciosos, o que reforça sua boa eficácia nas diferentes estratégias de seleção de clientes.

6. Conclusão

Este artigo realizou a reprodução simulada do ataque *label-flipping* no ambiente MiniNet-Fed. Os experimentos abrangeram treinamentos com e sem a presença de clientes maliciosos para investigar seu impacto na acurácia global dos modelos e na convergência dos mesmos. Diversas estratégias de seleção de clientes foram aplicadas, revelando que a maioria delas não conseguiu mitigar efetivamente a influência dos atores mal-intencionados. Esses resultados ressaltam a necessidade urgente de desenvolver técnicas mais robustas para proteger modelos no contexto do Aprendizado Federado contra ameaças externas.

Além disso, foi possível aplicar o algoritmo de clusterização *K-Means* no processo de treinamento. Isto possibilitou a detecção de clientes malicioso em 24 das 25 rodadas, o que demonstra a eficácia desse algoritmo para esse fim nos experimentos realizados. A partir disso, recomenda-se que seja melhor avaliado o uso dessa técnica em trabalhos futuros com mais experimentação.

Para trabalhos futuros, recomenda-se explorar novas métricas de avaliação de segurança e privacidade, além de investigar técnicas avançadas de detecção e mitigação

de ataques específicos como o *label-flipping*. Além disso, é fundamental aprimorar as estratégias de seleção de clientes, garantindo assim a integridade e a confiabilidade dos processos de treinamento no Aprendizado Federado.

Agradecimentos

Este trabalho possui financiamento parcial da Fapes (#2023/ RWXSZ, #2022/ ZQX6, #2022/ NGKM5, #2021/ GL60J) e Fapesp/ MCTI/ CGI.br (#2020/ 05182-3).

Referências

Alves, V. R. M. et al. (2024). Seleção de clientes adaptativa baseada em privacidade diferencial para aprendizado federado. In *SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC)*.

Bastos, J. J. S. et al. (2024). Mininetfed: Uma ferramenta para emulação e análise de aprendizado federado com dispositivos heterogêneos. In *SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC)*.

de Souza, A. M. et al. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. In *SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC)*.

Jebreel, N. M., Domingo-Ferrer, J., Sánchez, D., and Blanco-Justicia, A. (2022a). Defending against the label-flipping attack in federated learning.

Jebreel, N. M. et al. (2022b). Lfighter: Defending against the label-flipping attack in federated learning. In *Neural Networks*. Elsevier.

Jiang, Y., Zhang, W., and Chen, Y. (2023). Data quality detection mechanism against label flipping attacks in federated learning. *IEEE Transactions on Information Forensics and Security*, 18:1625–1637.

Li, D., Wong, W. E., Wang, W., Yao, Y., and Chau, M. (2021). Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pages 551–559.

Mammen, P. M. (2021). Federated learning: Opportunities and challenges. In *Proceedings of ACM Conference (Conference'17)*. ACM.

Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. (2020). Data poisoning attacks against federated learning systems.

Wang, T. et al. (2022). Federated learning framework based on trimmed mean aggregation rules. SSRN.