Detecção e Mitigação de Ataques de Inversão de Rótulos em Modelos Compactados e Privados no Aprendizado Federado

João Pedro C. Batista[1], Johann J. Schmitz Bastos[1], Ramon dos Reis Fontes[2] Eduardo Cerqueira[3], Vinícius F. S. Mota[1], Rodolfo S. Villaça[1]

[1]Departamento de Informática (DI) – Universidade Federal do Espírito Santo (Ufes)

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

[2]Instituto Metrópole Digital (IMD) – Universidade Federal do Rio Grande do Norte

ramon.fontes@imd.ufrn.br

[3]Instituto de Tecnologia (Itec) – Universidade Federal do Pará (UFPA)

cerqueira@ufpa.br

Abstract. *This paper proposes a technique to detect malicious clients performing label-flipping attacks while training compressed and privacy-preserving Federated Learning (FL) models through the application of differential privacy. The goal is identifying malicious clients manipulating data labels, even when local models are compressed and privacy-protected. The proposed technique uses the weight vector of the last activation layer of the neural network to detect these clients, preserving data privacy. We evaluated the proposal on different datasets, such as MNIST and Fashion-MNIST, using the* `MininetFed` *emulator. The proposed technique was able to detect and neutralize even when the network was made up of 40% malicious clients.*

Resumo. *Este artigo propõe uma técnica para detectar clientes maliciosos que realizam ataques de inversão de rótulo (label flipping) durante o treinamento de modelos de Aprendizado Federado (FL). O objetivo é identificar clientes maliciosos que manipulam rótulos de dados, mesmo com modelos locais compactados e privados por meio da aplicação de privacidade diferencial. O vetor de pesos da última camada de ativação da rede neural é usado para detectar comportamentos anômalos desses clientes, preservando a privacidade dos dados. A solução foi avaliada em diferentes conjuntos de dados, como MNIST e Fashion-MNIST, e no emulador* `MininetFed`*. Os resultados mostram que a proposta foi eficaz na detecção e neutralização de ataques, mesmo em cenários onde até 40% dos clientes na rede eram maliciosos.*

1. Introdução

O Aprendizado de Máquina tem se tornado cada vez mais presente em cidades, indústrias, governos e sociedade, impulsionado pelo aumento exponencial dos dados disponíveis para treinamento de modelos de Inteligência Artificial (IA). Esse processo, contudo, exige cada vez mais recursos computacionais e privacidade dos dados, o que tem levado à adoção de técnicas de aprendizado distribuído, onde diferentes dispositivos conectados em rede

colaboram para realizar as tarefas de treinamento, coordenadas por um servidor central. No entanto, a coleta e o compartilhamento de dados, especialmente quando provenientes de dispositivos na borda da rede (próximos aos usuários), levantam sérias preocupações com a privacidade, pois expõem informações sensíveis e confidenciais a potenciais violações [Dehghani and Yazdanparast 2023]. Para contornar essa questão, o Aprendizado Federado (do inglês *Federated Learning* - FL) surge como uma técnica de aprendizado colaborativo que busca preservar a privacidade dos dados [McMahan et al. 2016].

No FL, os dados de treinamento permanecem nos dispositivos dos usuários, e apenas os parâmetros dos modelos locais, treinados com esses dados, são compartilhados com o servidor central para agregação. Esse paradigma, composto pelas etapas de seleção de clientes, treinamento local, agregação e compartilhamento do modelo global, garante que os dados brutos nunca deixem os dispositivos dos usuários, proporcionando um maior nível de privacidade, em comparação com as técnicas tradicionais de aprendizado distribuído.

Apesar dos avanços em privacidade proporcionados pelo FL, o processo de treinamento ainda se mostra vulnerável a ataques de clientes maliciosos, que podem comprometer a integridade e a confiabilidade dos modelos globais [Kolasa et al. 2024]. Entre essas diversas categorias de ameaças [Manzoor et al. 2024], o ataque de inversão de rótulos, ou *label-flipping* [Shen et al. 2024] destaca-se como uma ameaça particularmente insidiosa, pois os clientes maliciosos manipulam os rótulos dos seus dados de treinamento, introduzindo erros deliberados no processo de aprendizado. Essa manipulação envenena o modelo global, prejudicando sua acurácia e capacidade de generalização.

As técnicas tradicionais de detecção de clientes maliciosos no aprendizado federado geralmente analisam os pesos dos modelos locais para identificar comportamentos suspeitos [Kolasa et al. 2024, Aloran 2024]. Contudo, essas abordagens podem ser limitadas em cenários onde a privacidade e a compactação dos modelos são prioridades, uma vez que ataques de inferência ainda podem explorar informações sensíveis a partir dos parâmetros compartilhados [He et al. 2024]. Felizmente, existem soluções disponíveis na literatura, tais como o `FedSketch` [Sarmento et al. 2024], que busca superar essa limitação ao empregar *sketches*, estruturas de dados probabilísticas, combinadas com privacidade diferencial, para compactar os pesos dos modelos locais antes do envio ao servidor.

Neste contexto, este artigo investiga a detecção de clientes maliciosos em ataques de inversão de rótulos em cenários em que os modelos locais são privados e compactados. O objetivo principal é propor e avaliar uma nova técnica de detecção que consiga identificar clientes maliciosos mesmo quando os modelos locais não estão acessíveis ao servidor, devido ao uso de técnicas de compactação e privacidade, desejadas em ambientes de FL.

A metodologia empregada neste estudo é de natureza experimental, fundamentada no uso do `MininetFed` [Bastos et al. 2024], uma ferramenta de emulação projetada especificamente para o FL. Todos os artefatos produzidos durante o desenvolvimento deste trabalho estão publicamente disponíveis [1] para fins de reprodutibilidade. Para avaliar o desempenho da proposta em diferentes cenários foram utilizados dois conjuntos de dados populares em FL (MNIST[2] e Fashion-MNIST[3]). A acurácia do modelo global foi adotada

[1] `https://github.com/lprm-ufes/MininetFed-LabelFlipping`
[2] `https://www.kaggle.com/datasets/hojjatk/mnist-dataset`
[3] `https://github.com/zalandoresearch/fashion-mnist`

como a principal métrica de avaliação, sendo comparada com a acurácia alcançada em cenários sem a presença de clientes maliciosos, bem como com os resultados obtidos por meio de técnicas tradicionais de detecção.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica do artigo, contendo os conceitos básicos do ataque de inversão de rótulos e do `FedSketch`. a Seção 3 aborda pesquisas relevantes sobre o tema para reafirmar as contribuições do artigo para o estado da arte. A Seção 4 apresenta o estado da arte da detecção do ataque em cenários com e sem privacidade dos modelos. A Seção 5 descreve o algoritmo de detecção proposto, os experimentos realizados e os resultados encontrados. Por fim, a Seção 6 conclui o trabalho e discute os trabalhos futuros.

2. Fundamentação Teórica

Esta seção apresenta os conceitos fundamentais do ataque de inversão de rótulos e do `FedSketch`, técnica usada no artigo para compactar e aumentar a privacidade dos modelos compartilhados no FL, contextualizando a problemática abordada neste artigo e fornecendo o embasamento teórico necessário para a compreensão da técnica proposta.

2.1. O ataque de Inversão de Rótulos (*Label-Flipping*)

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. 2024b]. O conceito desse tipo de ataque é envenenar o modelo global (e consequentemente os modelos locais pelo envio de parâmetros envenenados 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 [Elmahfoud et al. 2024].

Um exemplo pode ser construído utilizando-se o *dataset* MNIST, que contém diversas imagens de dígitos manuscritos e rotulados conforme o valor que representam. 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 Figura 1 ilustra esse procedimento, explicitando o envio de informações envenenadas para o servidor. Em ambientes de produção, com *datasets* reais, esse ataque é viável e facilmente reprodutível pois não exige muito poder computacional e não pode ser detectado de forma trivial pelo servidor, que não possui conhecimento dos dados locais do cliente atacante [Jebreel et al. 2024b].

2.2. FedSketch

O `FedSketch` é uma técnica que visa proteger a privacidade dos dados no FL, combinando *sketches* (estruturas de dados probabilísticas), para compactar os modelos locais, com privacidade diferencial, para aumentar a privacidade, também dos modelos locais. Os *sketches* permitem compactar os vetores de pesos dos modelos locais antes do envio ao servidor, reduzindo a carga de comunicação e o risco de inferência de dados sensíveis a partir dos modelos compartilhados. No `FedSketch` os modelos são compactados utilizando *count sketches*, uma estrutura de dados probabilística que utiliza tabelas *hashing* independentes para comprimir vetores de alta dimensionalidade. Os *sketches* são então

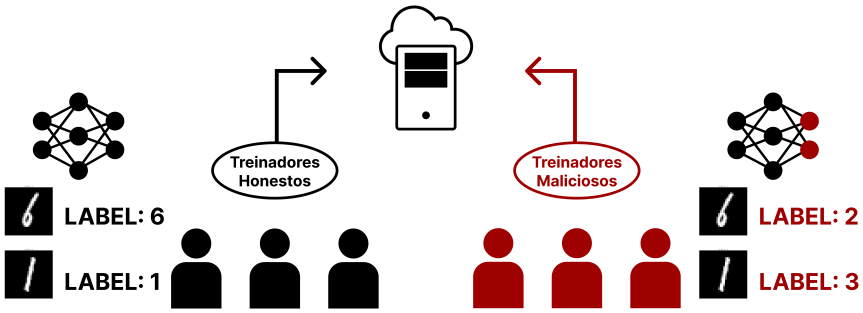


Figura 1. Exemplo de ataque de inversão de rótulos utilizando-se o *dataset* MNIST, em que os *labels* das Classes "6" e "1" foram alterados para as Classes "2" e "3", respectivamente.

transmitidos ao servidor que, por sua vez, agrega os modelos locais (sem descompactar) em um modelo global e os envia de volta aos clientes. Importante: reforça-se que, por privacidade, o servidor não consegue descompactar os modelos locais, fazendo a agregação somente a partir dos *sketches*, compactos e privados.

No `FedSketch`, os clientes descompactam, avaliam a utilidade do modelo global e retornam o resultado desta avaliação ao servidor de agregação por meio de métricas de utilidade e desempenho. Adicionalmente, a privacidade diferencial adiciona ruído aos dados compactados, garantindo que informações individuais não possam ser identificadas a partir dos modelos, mesmo em caso de ataques sofisticados. O `FedSketch` pode reduzir o tamanho do modelo em até 73 vezes, mantendo uma precisão similar ao FL convencional, enquanto pode alcançar um alto nível de privacidade diferencial, com $\epsilon \approx 10^{-6}$, o que significa que os dados individuais dos clientes são fortemente protegidos [Dwork 2006].

Essa combinação de técnicas torna o `FedSketch` uma solução importante e publicamente disponível[4] para o desenvolvimento de sistemas de FL mais privativos, devido ao emprego de privacidade diferencial nos modelos locais, e eficientes, em termos de comunicação de dados, devido à alta compactação dos modelos compartilhados. Desta forma, neste artigo, foi escolhida como ferramenta para implementar a compactação e o aumento da privacidade dos modelos locais no FL, com o objetivo de validar a proposta de detectar ataques de inversão de rótulos, mesmo em cenários onde os modelos locais estão compactados e protegidos por técnicas de privacidade diferencial.

3. Trabalhos Relacionados

A detecção de clientes maliciosos no FL é crucial para garantir a integridade e confiabilidade dos modelos globais. Diversos trabalhos se dedicam a estudar e propor soluções para detectar e mitigar os ataques de inversão de rótulos, que serão discutidos nesta seção.

Tolpegin et al. [Tolpegin et al. 2020] demonstraram que ataques de inversão de rótulos podem comprometer significativamente a precisão e o recall dos modelos, levando à queda de mais de 6% nessas métricas, mesmo com uma pequena porcentagem de clientes mal-intencionados. Eles propuseram o uso de Análise de Componentes Principais (PCA) para distinguir clientes maliciosos de clientes honestos, explorando as características únicas

[4] `https://github.com/lprm-ufes/FedSketch`

das atualizações de parâmetros enviadas pelos atacantes. Posteriormente, [Li et al. 2021] aprimoraram esse método substituindo PCA por *Kernel* PCA (KPCA) e incorporando o algoritmo de clusterização *K-Means*, impedindo completamente a degradação do modelo em cenários com até 4% de clientes maliciosos.

Além de analisar as atualizações dos modelos, [Jebreel et al. 2024a] propuseram uma técnica que examina os gradientes das atualizações para detectar clientes maliciosos, partindo da hipótese de que estes exibam gradientes maiores na camada de saída da rede neural. Essa abordagem se mostrou eficaz, mantendo a robustez do sistema mesmo com até 50% de clientes maliciosos. [Upreti et al. 2024] propuseram uma nova técnica de defesa usando a aproximação e projeção de variedades uniformes (UMAP) para detectar e mitigar ataques de inversão de rótulos em sistemas FL. Seus resultados demonstram que o UMAP supera outros métodos de redução de dimensionalidade e agrupamento, como PCA, KPCA e agrupamento *K-means*, fornecendo detecção e mitigação superiores de ataques de envenenamento de dados. [Jebreel et al. 2024b] também propõem o LFighter, um mecanismo de defesa que explora os objetivos conflitantes de invasores e pares honestos refletidos nos gradientes de parâmetros, demonstrando desempenho superior em termos de precisão, estabilidade e resistência a ataques em comparação com as defesas existentes.

[Li et al. 2023] abordaram a vulnerabilidade do FL a ataques de inversão de rótulos, particularmente em configurações não-IID, propondo o esquema HSCS (*Honest Score Client Selection*), que avalia e seleciona clientes com base em um vetor de risco para melhorar a robustez do modelo. [Jiang et al. 2023] propuseram o algoritmo MCDFL (*Malicious Clients Detection in Federated Learning*), que utiliza um gerador de dados treinado com base no modelo global para estimar a qualidade dos dados de cada cliente. O MCDFL superou o algoritmo de agregação padrão no FL, o *FedAvg*, em experimentos com Fashion-MNIST e CIFAR-10, mantendo a acurácia mesmo com alta porcentagem de clientes maliciosos.

Em resumo, a pesquisa sobre a detecção de clientes maliciosos, especialmente em ataques de inversão de rótulos, tem se concentrado em analisar as atualizações dos modelos, identificar padrões suspeitos e utilizar técnicas de agrupamento para isolar os atacantes. Entretanto, os trabalhos não abordam situações onde os modelos locais são protegidos por privacidade diferencial e compactados, como no `FedSketch`. Desta forma, as principais contribuições deste artigo são:

- Proposta de uma nova técnica de detecção de ataques de inversão de rótulos que funciona em cenários onde os modelos locais são privados e compactados;
- Análise do impacto da proporção de clientes maliciosos na eficácia da técnica proposta em comparação às técnicas usadas no estado da arte;
- Extensão do `MininetFed` com o suporte a ataques de inversão de rótulos, habilitando a ferramenta para reprodução dos experimentos e trabalhos futuros.

4. Detecção de Clientes Maliciosos no FL

Nesta seção, apresenta-se inicialmente a influência do ataque de inversão de rótulos no FL, considerando um cenário em que não há uso de técnicas de compactação e privacidade diferencial nos modelos locais (Cenário I). Em seguida, analisa-se o mesmo ataque, mas com os modelos locais compactados e protegidos por meio do `FedSketch` (Cenário II). Os ex-

perimentos nesta seção utilizaram as funções de agregação *FedAvg* (Cenário I) e *FedSketch* (Cenário II), além do modelo de rede neural convolucional LeNet5 [LeCun et al. 1998].

4.1. Cenário I: Sem Compactação e Privacidade Diferencial (*baseline*)

Como consequência do ataque de inversão de rótulos, é esperado que a acurácia do modelo global seja comprometida, além de um aumento na dificuldade de convergência do treinamento. Para avaliar tal comprometimento e estabelecer uma linha de base para comparação, o ataque de inversão de rótulos foi reproduzido em dois *datasets*: MNIST e F-MNIST. Dez clientes participaram de todas as rodadas de treinamento. Inicialmente, todos os clientes eram honestos, e, em um segundo momento, 10% dos clientes passaram a realizar o ataque, invertendo os rótulos de todas as classes dos conjuntos de dados.

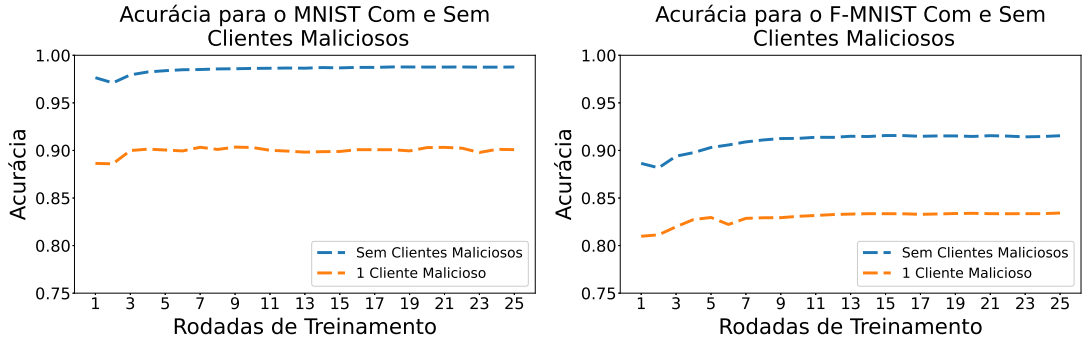


Figura 2. Acurácia do modelo global para os *datasets* (a) MNIST e (b) F-MNIST, com e sem a presença de agentes maliciosos.

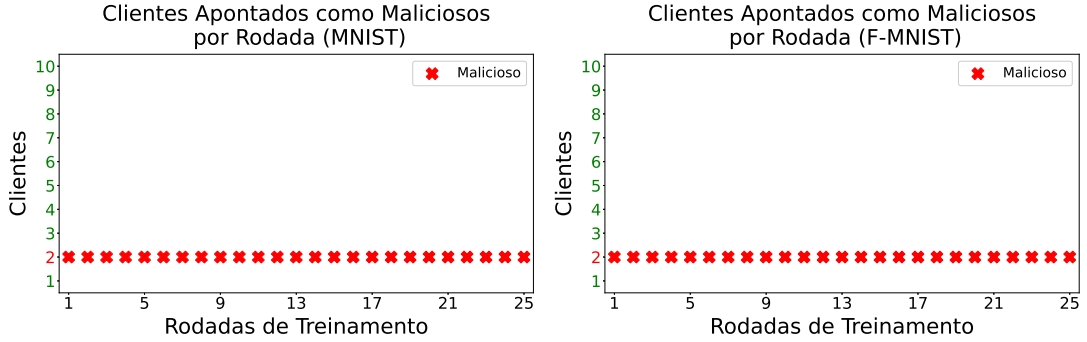


Figura 3. Seleção de clientes maliciosos por *K-Means* com os *datasets* (a) MNIST e (b) F-MNIST

A Figura 2(a) ilustra que, no caso do MNIST, a acurácia do modelo global, inicialmente estabilizada acima de 98%, foi reduzida para menos de 90% com a presença de um cliente malicioso. Resultado semelhante foi observado no F-MNIST (Figura 2(b)), cuja acurácia inicial, ligeiramente abaixo de 90%, caiu para aproximadamente 83%, demonstrando o impacto significativo do ataque na acurácia do modelo global.

O estado da arte para a detecção de clientes maliciosos no processo de aprendizado federado é a aplicação do algoritmo de clusterização *K-Means* [Jebreel et al. 2024b]. Nesse contexto, a clusterização divide os clientes participantes em dois grupos: maliciosos e honestos. Essa classificação é realizada com base na clusterização dos vetores de peso dos modelos enviados ao servidor em cada rodada.

A Figura 3 apresenta os clientes classificados como maliciosos pelo algoritmo *K-Means*, em cada rodada de treinamento, para ambos os *datasets*. Na simulação, cada cliente recebeu um identificador, sendo que o cliente *2* foi configurado como malicioso. Observa-se que o algoritmo identificou corretamente o cliente malicioso em todas as rodadas, demonstrando sua eficácia na detecção de comportamentos anômalos.

4.2. Cenário II: Com Compactação e Privacidade Diferencial (`FedSketch`)

Com a adoção do `FedSketch` para compactar as atualizações locais do modelo e adicionar privacidade diferencial, avalia-se a viabilidade de detecção de clientes maliciosos com base nos vetores gerados pelos *count sketches* de cada cliente.

Para essa análise, seguiu-se o mesmo procedimento do cenário anterior. Todos os clientes utilizaram o `FedSketch` para compactar os pesos de seus modelos locais. Em 10% dos clientes, os rótulos de todas as classes do *dataset* MNIST foram invertidos. Os vetores de peso compactados gerados pelos `FedSketch` foram então submetidos a uma técnica de redução de dimensionalidade (PCA), a fim de verificar sua distribuição no espaço tridimensional.

A Figura 4 demonstra que, devido à natureza probabilística dos *count sketches*, os vetores gerados pelo `FedSketch` apresentam distribuição aleatória no espaço, o que inviabiliza a aplicação de métodos descritos na literatura, como o *K-Means*, para identificar clientes maliciosos. Assim, destaca-se a necessidade de desenvolver um novo método capaz de realizar a análise proposta, dado que a similaridade entre os modelos locais compactados não é suficiente para agrupar clientes normais e maliciosos.

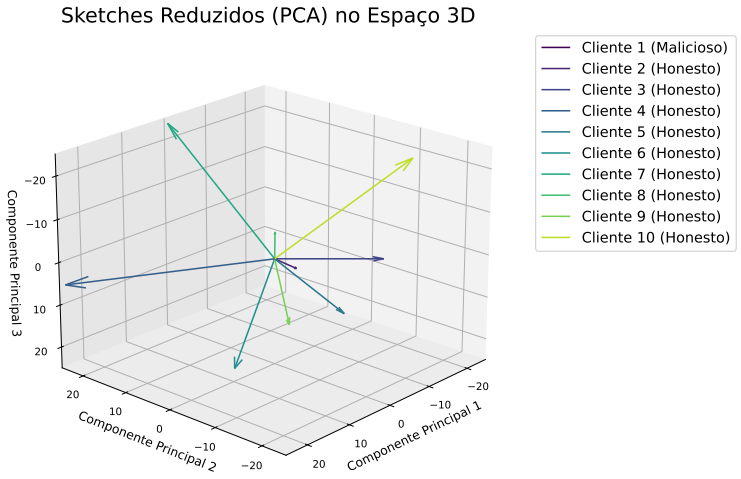


Figura 4. Distribuição no espaço tridimensional dos vetores reduzidos (PCA) gerados pelos *count sketches* dos clientes em uma rodada do FL.

5. Detecção de Clientes Maliciosos com Modelos Privados

Esta seção tem como objetivo apresentar uma proposta de algoritmo que seja eficaz na detecção e mitigação de ataques de inversão de rótulos em contextos em que os dados estão compactados e protegidos por privacidade diferencial, bem como demonstrar, a partir de experimentos e resultados, a eficácia da proposta.

5.1. Algoritmo

Como discutido na Seção 4.2, a aplicação de técnicas de compactação e privacidade diferencial, presentes na abordagem `FedSketch`, não permitem a aplicação dos métodos de detecção e mitigação de ataques comumente usados e descritos no estado da arte. Esse impedimento deve-se à característica probabilística e aleatória da estrutura de dados utilizada (*count sketches*). Nesse sentido, propõe-se uma nova abordagem para realizar essa detecção, descrita no Algoritmo 1.

A técnica proposta utiliza o vetor de pesos da camada de saída da rede neural convolucional como meio de detecção. Para isto, os clientes enviam este vetor de pesos ao servidor, por meio de uma mensagem definida no `MininetFed`, juntamente com os *count sketches*, que representam a compactação e aplicação de privacidade diferencial nos modelos completos locais. A premissa da proposta é que esse vetor da última camada de ativação, quando provenientes de treinadores maliciosos, contenha valores significativamente diferentes em relação aos clientes honestos [Jebreel et al. 2024b] e, a partir deles, seja possível identificá-los por meio de técnicas de agrupamento (como apresentados na Seção 4.2) sem comprometer a privacidade dos modelos locais. É importante ressaltar que a revelação da última camada não compromete a privacidade dos modelos locais, pois a última camada, embora importante para a detecção de *outliers*, não contém informações suficientes para reconstruir os dados de treinamento ou inferir informações sensíveis sobre os clientes [He et al. 2024, Kolasa et al. 2024].

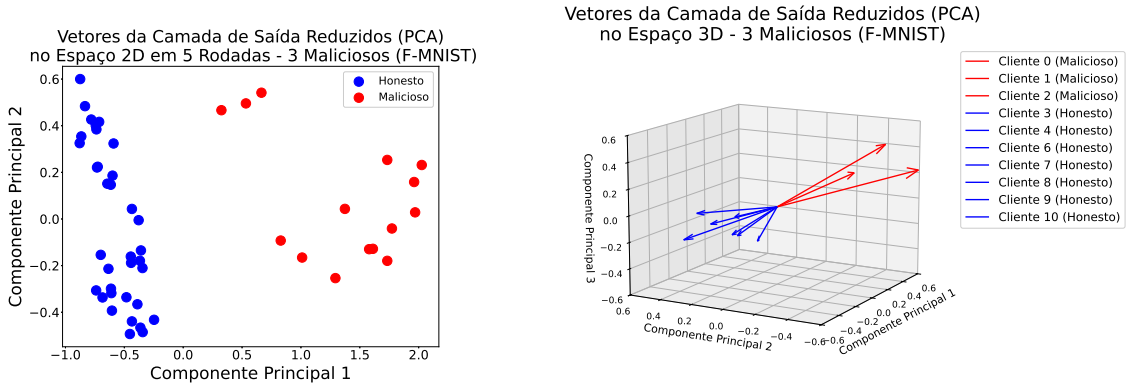


Figura 5. Distribuição no espaço (a) bidimensional e (b) tridimensional dos vetores da última camada do modelo, enviados pelos clientes. Em (a) estão as contribuições de todos os clientes ao longo de cinco rodadas e em (b) estão as contribuições de todos os clientes em uma única rodada

A Figura 5 demonstra um cenário em que foi aplicada uma redução de dimensionalidade (PCA) aos vetores da última camada da rede neural convolucional, enviados pelos clientes durante o treinamento com o *dataset* F-MNIST e na presença de três clientes maliciosos. A partir dessa aplicação, foi possível avaliar a distribuição (a) bidimensional e (b) tridimensional dos dados. Observa-se que os vetores enviados pelos agentes maliciosos, em vermelho, apresentam distâncias consideráveis em relação aos enviados pelos demais clientes, o que aponta para a viabilidade da técnica proposta neste artigo.

Desse modo, é possível analisar e classificar os clientes a partir da distância calculada entre os vetores de peso da última camada, classificando como clientes maliciosos

Algoritmo 1 Algoritmo Proposto para Detecção do Ataque de Inversão de Rótulos por meio da Camada de Ativação dos Modelos Locais

Require: $trainer\_list$: Lista de treinadores, $activation\_vectors$: Lista de vetores de pesos da última camada da rede neural convolucional dos clientes
Ensure: Lista de treinadores possivelmente maliciosos
1: $total\_distances \leftarrow []$
2: **for all** $trainer\_layer \in activation\_vectors$ **do**
3: $\quad euclidean\_distance \leftarrow 0$
4: $\quad$ **for all** $trainer\_layer2 \in activation\_vectors$ **do**
5: $\quad\quad euclidean\_distance \leftarrow euclidean\_distance + ||trainer\_layer - trainer\_layer2||$
6: $\quad$ **end for**
7: $\quad total\_distances.append(euclidean\_distance)$
8: **end for**
9: $modified\_zscore \leftarrow []$
10: **for all** $distance \in total\_distances$ **do**
11: $\quad zscore \leftarrow CalculateModifiedZScore(distance)$
12: $\quad modified\_zscore.append(zscore)$
13: **end for**
14: $malicious\_trainers \leftarrow []$
15: **for all** $i, zscore \in modified\_zscore$ **do**
16: $\quad$ **if** $zscore \geq 5 \vee zscore \leq -5$ **then**
17: $\quad\quad malicious\_trainers.append(trainer\_list[i])$
18: $\quad$ **end if**
19: **end for**
20: **return** $malicious\_trainers$

aqueles que se apresentarem como *outliers* no conjunto de treinadores presentes no FL. A partir disso, sugere-se a implementação do Algoritmo 1, descrito a seguir, e executado no servidor de agregação.

No Algoritmo 1, as linhas 1 a 3 realizam a concatenação de todas as camadas de ativação (recebidas por cada um dos clientes) em um vetor único. Em seguida, nas linhas 5 a 12, é calculada a distância euclidiana de cada vetor para os demais (recebidos), sendo cumulativa a soma das distâncias (i.e., a soma das distâncias do vetor de um cliente para os demais). Posteriormente, nas linhas 13 a 17, é calculado o valor Z_i de cada uma das distâncias, sendo Z_i o *Z-Score*-Modificado. Para o cálculo de Z_i, utiliza-se o MAD, que é o desvio absoluto da mediana. Z_i e MAD estão definidos nas equações que seguem, em que $\tilde{X}$ é a mediana das distâncias:

$$Z_\text{i} = \frac{x_\text{i} - \tilde{X}}{MAD} \tag{1}$$

$$MAD = median(|x_\text{i} - \tilde{X}|) \tag{2}$$

Às somas das distâncias aplicou-se o teste de Shapiro-Wilk para verificar se estes dados seguiam uma distribuição normal. Em todos os casos, o valor encontrado para a probabilidade condicional (*p-value*) foi inferior a 5%, o que significa que há evidências suficientes para afirmar que os dados não seguem tal distribuição, permitindo o uso do *Z-Score* Modificado. Essa métrica estatística é utilizada para determinar *outliers* em conjuntos de dados que não possuem distribuição normal, uma vez que considera a mediana e o

desvio absoluto da mediana em detrimento à média e ao desvio padrão dos dados, que, por sua vez, são fortemente afetados por valores discrepantes [Iglewicz and Hoaglin 1993]. Nesse caso, considera-se que uma amostra é um *outlier* quando o seu valor *Z-Score* é superior ao módulo de um valor de referência (*threshold*). [Iglewicz and Hoaglin 1993] sugerem que se considere como *threshold* valores superiores, em módulo, a 3.5, sendo 5 o valor escolhido para o algoritmo utilizado neste artigo.

Por fim, nas linhas 18 a 24, selecionam-se os clientes considerados maliciosos conforme avaliações realizadas nas linhas anteriores. De posse da lista de agentes suspeitos, o servidor exclui as suas contribuições do processo de agregação e de cálculo da acurácia global para aquela rodada. Entretanto, essa ação é efetuada apenas pelo servidor, sem que os clientes maliciosos saibam que estão sendo excluídos do processo. Portanto, esses treinadores participam normalmente do treinamento, mas seus dados não são considerados na agregação do modelo global, que, lembrando, é feita de forma privativa com uso do `FedSketch`.

5.2. Avaliação do Algoritmo Proposto

Para avaliar o desempenho do algoritmo proposto sobre a acurácia do modelo global e na tarefa de identificação dos clientes maliciosos, realizaram-se dois experimentos. Estes foram construídos a partir de dois *datasets*: MNIST e F-MNIST, respectivamente.

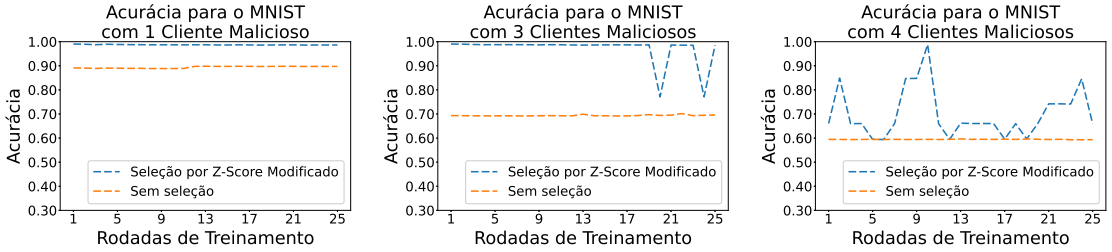
Em ambos os experimentos, 10 (dez) clientes participaram do treinamento, em três diferentes situações: i) 1 (um) cliente malicioso e 9 (nove) honestos; ii) 3 (três) clientes maliciosos e 7 (sete) honestos e; iii) 4 (quatro) clientes maliciosos e 6 (seis) honestos. Todos os clientes utilizaram a rede neural convolucional LeNet5, treinada por 10 épocas e o método `FedSketch`, disponível publicamente no `MininetFed`, e aqueles que eram maliciosos efetuaram a inversão dos rótulos de todas as classes do *dataset* utilizado. O servidor utilizou a função de agregação *fed_sketch*.

A quantidade de clientes e representantes maliciosos foi cuidadosamente definida para evidenciar o impacto da proporção de agentes atacantes durante o treinamento. Foram avaliados cenários com proporções de 10%, 30% e 40% de clientes maliciosos. Cada sessão de treinamento foi limitada a 25 rodadas, o que se mostrou suficiente para alcançar um platô na acurácia para os conjuntos de dados (*datasets*) utilizados.

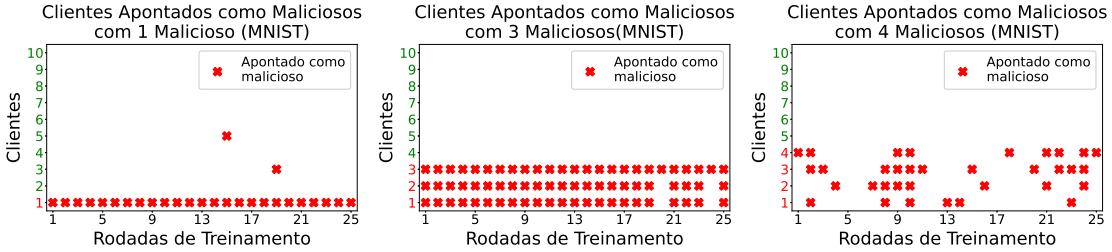
É importante destacar que o foco principal dos experimentos foi avaliar a eficácia do algoritmo proposto na detecção de clientes maliciosos. Assim, a obtenção da maior acurácia possível com os *datasets* e configurações adotadas no treinamento não foi o objetivo central do experimento. O delineamento experimental priorizou a análise do desempenho do algoritmo sob diferentes níveis de adversidade, reforçando sua capacidade de identificar comportamentos maliciosos em cenários federados.

Os resultados dos experimentos realizados para avaliar a proposta deste artigo demonstraram diversas tendências relevantes. Inicialmente, na Figura 6(a) com o *dataset* MNIST, observou-se que quando 10% dos clientes maliciosos executaram o ataque de inversão de rótulos, a acurácia final do modelo global caiu em aproximadamente 10% na ausência do algoritmo proposto. Com o uso do algoritmo, a acurácia foi mantida, evidenciando a eficácia da solução proposta neste cenário. Quando a proporção de clientes maliciosos foi aumentada para 30% e 40%, os efeitos adversos sobre a acurácia foram ainda mais pronunciados. Sem o algoritmo de detecção, a acurácia final caiu drasticamente

para aproximadamente 70% com 30% de clientes maliciosos e para aproximadamente 60% com 40% desses agentes. No entanto, utilizando o algoritmo de detecção, a acurácia foi mantida em torno de 98% durante 23 das 25 rodadas, com 30% de clientes maliciosos, além de registrar alguns picos e platôs consideráveis na acurácia para o cenário, com 40% de agentes atacantes. Isso demonstra que a solução consegue mitigar parcialmente os impactos de um percentual elevado de adversários.



(a) Acurácia média do modelo global para diferentes proporções de clientes maliciosos.



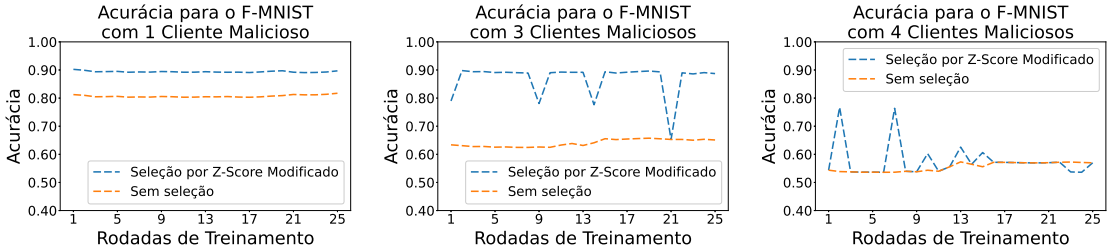
(b) Clientes identificados como maliciosos pelo servidor a cada rodada do treinamento

Figura 6. Resultados do experimento com o *dataset* MNIST. Em (b), os rótulos de clientes (no eixo Y) que estão em vermelho indicam os clientes que atuaram como maliciosos e, em verde, os que atuaram como honestos.

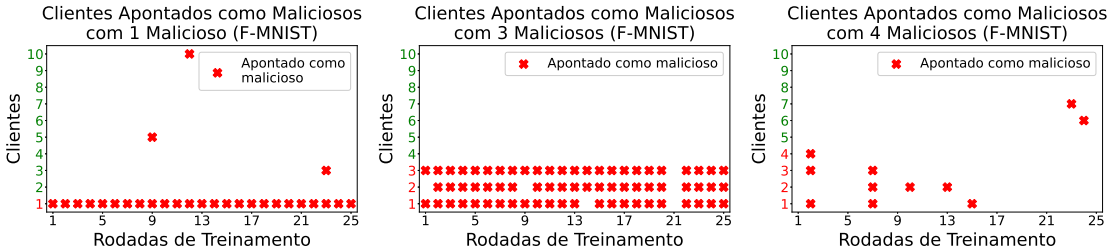
Além disso, a convergência do modelo global tornou-se progressivamente mais desafiadora à medida que a proporção de clientes maliciosos aumentou. Com 40% de clientes dessa classe no *dataset* MNIST, o modelo global não conseguiu atingir estabilidade no treinamento. Isso indica que o algoritmo possivelmente sofre uma degradação em sua eficácia a partir de um determinado percentual de agentes mal-intencionados, o que se justifica pelo fato de este trabalhar sobre *outliers*. Assim, quando os clientes maliciosos aproximam-se da igualdade ou superioridade numérica em relação aos clientes honestos, eles não mais atuam como *outliers* naquele universo, o que dificulta a ação do algoritmo.

Outro aspecto avaliado foi a eficiência do algoritmo na identificação dos clientes maliciosos. Pode-se observar na Figura 6(b) que, quando a proporção de clientes maliciosos era de 10%, o algoritmo conseguiu identificar corretamente todos os clientes maliciosos, com a ocorrência de dois falsos positivos nas rodadas 15 e 19. Com 30% de clientes maliciosos, não foram observados falsos positivos. Entretanto, ocorreram dois falsos negativos nas rodadas 20 e 24, que impactaram na estabilidade da acurácia do modelo global. Quando a proporção foi aumentada para 40%, o algoritmo apresentou dificuldade significativa em detectar os clientes maliciosos, resultando em vários falsos negativos, que impactaram de forma relevante a acurácia do modelo global e comprometeram sua convergência.

Os resultados obtidos com o *dataset* F-MNIST, apresentados na Figura 7, registra-



(a) Acurácia média do modelo global para diferentes proporções de clientes maliciosos.



(b) Clientes identificados como maliciosos pelo servidor a cada rodada do treinamento.

Figura 7. Resultados do experimento com o *dataset* F-MNIST. Em (b), os rótulos de clientes (no eixo Y) que estão em vermelho indicam os clientes que atuaram como maliciosos e, em verde, os que atuaram como honestos.

ram um comportamento similar. Contudo, com o uso do algoritmo, houve o registro de uma acurácia do modelo global pior do que a métrica registrada sem o algoritmo para o cenário com 40% de adversários.

De forma geral, o algoritmo de detecção proposto apresenta maior eficiência quanto menor for a proporção de clientes maliciosos, permitindo mitigar ataques, manter altos níveis de acurácia e assegurar a convergência dos modelos. Contudo, essas características se deterioram gradualmente à medida que a proporção de clientes maliciosos aumenta.

5.3. Considerações Finais

Na proposta, a agregação dos modelos é realizada utilizando *count sketches* compactados e privados por meio da técnica de privacidade diferencial, empregando o algoritmo `FedSketch`. Essa abordagem assegura a privacidade e segurança dos dados durante o processo de agregação, ao mesmo tempo que reduz a sobrecarga de comunicação. É crucial destacar que não seria possível utilizar apenas a última camada para a agregação dos modelos locais em um modelo global, pois isso tornaria a agregação inviável [He et al. 2024].

A detecção de clientes maliciosos é implementada por meio de uma análise de *outliers*, utilizando as distâncias dos vetores gerados pela última camada de ativação. Essa estratégia permite identificar comportamentos anômalos no treinamento federado, diferenciando contribuições legítimas de padrões suspeitos, mesmo em cenários com uma alta proporção de agentes atacantes. Entretanto, quando a proporção de clientes maliciosos aumenta, eles deixam de ser minoria no aprendizado, o que dificulta a detecção de anomalias, especialmente quando suas características se aproximam das observadas nos clientes honestos. Além disso, como a detecção é feita por uma métrica estatística, falsos positivos podem surgir, especialmente em cenários em que existem poucos clientes para comparação. Assim, apesar da proporção destacada de clientes maliciosos, pequenas

diferenças entre os modelos podem rotular um cliente como malicioso mesmo ele sendo honesto (Figuras 6(b) e 7(b)).

6. Conclusão

Este artigo avaliou a mitigação de ataques de inversão de rótulos no Aprendizado Federado utilizando uma abordagem que combina a detecção de clientes maliciosos com a preservação da privacidade dos modelos locais. Os experimentos, conduzidos em um ambiente emulado no `MininetFed`, demonstraram a eficácia da técnica proposta na detecção e neutralização de clientes maliciosos, quando estes são minoria no processo de aprendizado (até 40%). Os resultados obtidos são comparáveis ao estado da arte em termos de acurácia do modelo global, confirmando a viabilidade da abordagem em cenários realísticos. Este trabalho destaca-se como o primeiro a abordar a detecção de clientes maliciosos em FL utilizando modelos locais compactados com *count sketches* e privacidade diferencial. A combinação destas técnicas demonstra um avanço significativo na busca por soluções robustas e seguras para FL em ambientes suscetíveis a ataques.

Novas frentes de pesquisa surgem a partir desta proposta. Uma delas é a avaliação da técnica em *datasets* com distribuição *non-iid*, o que representaria um desafio adicional para a detecção de clientes maliciosos, reduziria a eficiência dos resultados encontrados e exigiria a adaptação dos algoritmos apresentados para clusterização e detecção de anomalias. Outra linha de pesquisa importante é a exploração de diferentes mecanismos de privacidade diferencial e técnicas de compactação de modelos, além do `FedSketch`. Adicionalmente, a investigação de métodos de detecção de ataques mais sofisticados, como *backdoors* e ataques mais complexos. Por fim, busca-se prover alguma forma de assinatura digital para os treinamentos locais, de modo a evitar que os clientes maliciosos possam burlar a técnica proposta enviando um vetor de última camada que seja diferente da última camada associada ao modelo compactado e privado.

Agradecimentos

Este trabalho possui financiamento parcial de: PIIC/Ufes; CNPq; CAPES (Finance Code #001 e Processo 88887.005666/2024-00); Fapes (#2023/ RWXSZ, #2022/ ZQX6, #2022/ NGKM5, #2021/ GL60J); e Fapesp/ MCTI/ CGI.br (#2020/ 05182-3 e #2023/00148-0).

Referências

Aloran, I. (2024). *Defending Federated Learning Against Model Poisoning Attacks*. Master's thesis, University of Windsor. Electronic Theses and Dissertations, 9458.

Bastos, J., Sarmento, E., Villaça, R., and Mota, V. (2024). Mininetfed: Uma ferramenta para emulação e análise de aprendizado federado com dispositivos heterogêneos. In *Anais Estendidos do XLII SBRC*, Porto Alegre, RS, Brasil. SBC.

Dehghani, M. and Yazdanparast, Z. (2023). From distributed machine to distributed deep learning: a comprehensive survey. *Journal of Big Data*, 10(1):158.

Dwork, C. (2006). Differential privacy. In Bugliesi, M., Preneel, B., Sassone, V., and Wegener, I., editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg. Springer Berlin Heidelberg.

Elmahfoud, E., Hajla, S. E., Maleh, Y., Mounir, S., and Ouazzane, K. (2024). Label flipping attacks in hierarchical federated learning for intrusion detection in iot. *Information Security Journal: A Global Perspective*.

He, X., Xu, Y., Zhang, S., Xu, W., and Yan, J. (2024). Enhance membership inference attacks in federated learning. *Computers & Security*, 136:103535.

Iglewicz, B. and Hoaglin, D. C. (1993). *How to Detect and Handle Outliers*, volume 16. American Society for Quality Control.

Jebreel, N. M., Domingo-Ferrer, J., Blanco-Justicia, A., and Sánchez, D. (2024a). Enhanced security and privacy via fragmented federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):6703–6717.

Jebreel, N. M., Domingo-Ferrer, J., Sánchez, D., and Blanco-Justicia, A. (2024b). Lfighter: Defending against the label-flipping attack in federated learning. *Neural Networks*, 170:111–126.

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.

Kolasa, D., Pilch, K., and Mazurczyk, W. (2024). Federated learning secure model: A framework for malicious clients detection. *SoftwareX*, 27:101765.

LeCun, Y. et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

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 Int. Conf. on Dependable Systems and Applications (DSA)*, pages 551–559. IEEE.

Li, Y., Chen, H., Bao, W., Xu, Z., and Yuan, D. (2023). Honest score client selection scheme: Preventing federated learning label flipping attacks in non-iid scenarios.

Manzoor, H. U., Shabbir, A., Chen, A., Flynn, D., and Zoha, A. (2024). A survey of security strategies in federated learning: Defending models, data, and privacy. *Future Internet*, 16(10).

McMahan, H. B. et al. (2016). Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*.

Sarmento, E., Mota, V., and Villaça, R. (2024). Privacidade e comunicação eficiente em aprendizado federado: Uma abordagem utilizando estruturas de dados probabilísticas e seleção de clientes. In *Anais do XLII SBRC*, pages 85–98, Porto Alegre, RS, Brasil. SBC.

Shen, X., Liu, Y., Li, F., and Li, C. (2024). Privacy-preserving federated learning against label-flipping attacks on non-iid data. *IEEE Internet of Things Journal*, 11(1).

Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. (2020). Data poisoning attacks against federated learning systems. In *ESORICs 2020: 25th European symposium on research in computer security*, pages 480–501. Springer.

Upreti, D., Kim, H., Yang, E., and Seo, C. (2024). Defending against label-flipping attacks in federated learning systems using uniform manifold approximation and projection. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 13(1):459–466.