Rápido, Privado e Protegido: Uma Abordagem para Aprendizado Federado Eficiente em Ambiente Hostil

Nicolas R. G. Assumpção[1], **Leandro A. Villas**[1]

[1]Universidade Estadual de Campinas, Brasil

n121245@dac.unicamp.br, lvillas@ic.unicamp.br

Resumo. *O Aprendizado Federado (Federated Learning - FL) é um método de treinamento distribuído em que dispositivos colaboram para criar um modelo global sem compartilhar dados, permitindo treinamento em cenários com informações privadas. Entretanto, garantir a privacidade dos dados ao mesmo tempo que se protege a convergência do modelo é um grande desafio, dado que as soluções normalmente conseguem abranger apenas uma dessas duas proteções. Neste trabalho, introduzimos o RPP (Rápido, Privado e Protegido), uma abordagem de rápida convergência e que protege o treinamento contra ataques de envenenamento de modelo ao mesmo tempo que possibilita o uso de técnicas de criptografia homomórfica para proteger a privacidade dos dados. Isso é feito ao usar as avaliações dos clientes para avaliar as rodadas anteriores e recuperar o treinamento após um ataque agressivo. O RPP utiliza valores de reputação para dificultar que atacantes sejam selecionados. Experimentos realizados compararam o RPP com outras abordagens da literatura (FedAvg, PoC, Agregação por Mediana e Agregação por Média Podada) e mostraram como o RPP obteve uma convergência rápida e consistente em cenários onde todas as outras falharam em convergir.*

Abstract. *Federated Learning (FL) is a distributed training method where devices collaborate to create a global model without sharing data, enabling training in scenarios with private information. However, ensuring data privacy while protecting model convergence is a major challenge, as solutions typically address only one of these protections. In this work, we introduce RPP (Rapid, Private, and Protected), an approach with fast convergence that guards against model poisoning attacks while enabling the use of homomorphic encryption techniques to protect data privacy. This is achieved by leveraging client evaluations to assess previous rounds and recover training after aggressive attacks. RPP utilizes reputation values to deter attackers from being selected. Experiments comparing RPP with other literature approaches (FedAvg, PoC, Median Aggregation, and Trimmed Mean Aggregation) demonstrated how RPP achieved fast and consistent convergence in scenarios where all others failed to converge.*

1. Introdução

O Aprendizado Federado (*Federated Learning* - FL) é uma técnica de treinamento de aprendizado de máquina realizado de modo descentralizado em dispositivos distribuídos que compartilham apenas os parâmetros atualizados durante o processo de treinamento. Com isso, essa técnica elimina a necessidade de centralizar os dados, mantendo os dados dos participantes nos dispositivos.

Para isso, o FL reúne um grupo de dispositivos (clientes) com seus respectivos dados em um processo de treinamento distribuído onde, em cada iteração (chamada rodada), os clientes recebem o modelo global, executam algumas etapas de treinamento local usando seus dados privados e depois retornam os modelos treinados para que sejam agregados em um novo modelo global unificado. Deste modo, o treinamento acontece sem que os dados precisem deixar o dispositivo.

Essa propriedade oferece uma alternativa para treinamento de modelos sob os novos regulamentos de proteção de dados, como a *General Data Protection Regulation* (GDPR) da União Européia, a *California Consumer Privacy Act* (CCPA) do estado americano da Califórnia e a Lei Geral de Proteção de Dados (LGPD) do Brasil.

Entretanto, o uso de dados privados e a natureza distribuída do FL apresentam grandes desafios, como a natureza diversa da distribuição de dados dos participantes, a garantia da privacidade dos dados e a proteção contra eventuais ataques feitos por clientes maliciosos.

Como os clientes treinam em paralelo utilizando apenas os dados referentes ao próprio contexto, isso faz com que os dados não sejam independentes e identicamente distribuídos (dados não-iid). Tal cenário pode fazer com que os clientes treinem modelos incompatíveis entre si, fazendo com que o resultado da agregação perca a consistência.

A própria garantia da privacidade também é um grande desafio. Trabalhos como [Zhu et al. 2019] já demonstraram que é possível reconstruir o conjunto de dados usando tão somente os gradientes compartilhados pelos clientes.

Outro desafio que advém da natureza distribuída do treinamento é a possibilidade de ataques que buscam prejudicar o resultado final. Ataques de envenenamento de modelo (onde os atacantes corrompem o protocolo de treinamento) buscam arruinar o resultado da agregação, inutilizando os esforços dos clientes honestos na criação de um modelo global acurado [Lyu et al. 2020] [Nair et al. 2023]. Técnicas para proteger o treinamento podem incluir auditorias nos gradientes recebidos dos clientes a fim de procurar por sinais de corrupção nos valores dos gradientes (ver seção 2).

Proteger a privacidade dos dados privados ao mesmo tempo que se protege o treinamento contra ataques não é simples, dado que os problemas parecem exigir soluções antagonistas. De um lado, toma-se medidas para que os gradientes não possam ser averiguados e de outro lado cria-se técnicas para esquadrinhar os gradientes a procura de sinais de ataque.

É neste contexto que introduzimos o RPP, uma nova abordagem de FL que atua na seleção de clientes para deixar o treinamento mais eficiente em cenários com dados não-iid ao mesmo tempo que propõe um método de avaliação do treinamento que pode ser utilizado junto com criptografia homomórfica. O RPP ainda possibilita a recuperação de uma versão anterior do modelo global em casos de ataques muito severos, preservando a evolução do treinamento anterior ao ataque.

A próxima Seção discute trabalhos relacionados, mostrando suas contribuições e limitações. A Seção 3 introduz o RPP e descreve como ele mitiga os problemas de dados não-iid e ataques de envenenamento de modelo ao mesmo tempo que protege a privacidade dos dados distribuídos. A Seção 4 descreve os experimentos e análises realizadas e

a Seção 5 conclui este trabalho apresentando também oportunidades de trabalho futuro.

2. Trabalhos Relacionados

Existem diferentes desafios no contexto de FL por conta de sua natureza distribuída e por envolver dados sensíveis, muitas vezes não-iid. Esta seção apresentará algumas das propostas encontradas na literatura para enfrentar esses desafios.

O FedAvg [Brendan McMahan et al. 2016] propõe a execução de várias etapas de treinamento nos clientes antes de realizar uma nova agregação. Essa proposta acelera o treinamento no FL, mas seu funcionamento não é adequado em casos onde os dados dos clientes não são independentes e identicamente distribuídos (não-iid).

O *Power-of-Choice* (PoC) [Cho et al. 2020] propõe uma estratégia de seleção de clientes mais sofisticada, permitindo treinar com menos clientes em cada rodada e alcançando a convergência em menos ciclos de treinamento. O PoC faz isso ao concentrar o treinamento nos clientes que apresentam maior valor de perda, despriorizando clientes que já apresentam boas métricas pois estes tem uma margem menor para colaborar com o treinamento. Essa abordagem é particularmente efetiva em cenários onde os dados são não-iid, pois nestes casos os clientes são muito diferentes entre si, apresentando diversos valores de métricas. Entretanto, o PoC não oferece qualquer proteção contra ataques de clientes maliciosos.

Visando proteger a privacidade dos dados dos usuários, a estratégia de agregação segura [Bonawitz et al. 2017] usa criptografia homomórfica para mascarar os gradientes de cada cliente de modo que a soma de todos os gradientes mascarados mantenha-se consistente. Isso é feito através de um protocolo onde valores são adicionados nos gradientes de modo que estes valores sejam cancelados durante a soma na etapa de agregação.

A agregação segura previne que os gradientes compartilhados por um cliente sejam descobertos, mas também impede auditorias a fim de garantir sua legitimidade. Deste modo, não é possível utilizar a agregação segura com métodos como o [Kang et al. 2019], que vasculha os gradientes recebidos em busca de distribuições incomuns para identificar ataques.

Alguns trabalhos [Wang et al. 2020][Zhang et al. 2021][Kang et al. 2019] usam conjuntos de dados públicos para avaliar a qualidade do modelo agregado, identificando ataques de envenenamento ao constatar uma piora nas métricas do modelo quando submetidos a este banco de dados de validação. Entretanto, considerando o contexto onde o FL é utilizado, é difícil garantir a representatividade de um banco de dados público sendo possível ainda que tal banco de dados não esteja disponível.

Outras estratégias que tentam oferecer proteção contra ataques de envenenamento de modelo atuam no momento da agregação, agregando os gradientes através de versões mais sofisticadas da média do FedAvg. [Yin et al. 2018] usa a mediana para agregar os modelos ou ainda propõe a remoção de valores extremos antes da média (média podada), [Pillutla et al. 2019] faz uso da mediana geométrica para obter o centro de massa dos gradientes e [Xie et al. 2018] faz a média apenas dos valores mais próximos da mediana. Essas estratégias buscam remover valores muito discrepantes do cálculo. Não obstante, tais estratégias também são infactíveis de serem combinadas com a agregação segura, que preserva a soma e a média dos gradientes, mas que não permite o cálculo da mediana ou

da média podada.

Estes trabalhos mostram que a segurança da privacidade de conjuntos de dados não-iid durante um treinamento distribuído com a potencial participação de atacantes é uma tarefa que tem sido tratada apenas em partes, com os trabalhos concentrando-se em apenas um problema. Em particular, tratar simultaneamente da garantia da privacidade e proteção contra ataques cria um impasse entre analisar os dados ou escondê-los.

Para mitigar essa limitação, apresentamos o RPP, uma estratégia que busca desfrutar dos mesmos benefícios da escolha de clientes baseada na pré-avaliação das métricas para obter ganho na velocidade de convergência em cenários não-iid, ao mesmo tempo que oferece proteção contra ataques de envenenamento, podendo ser combinado com a agregação segura a fim de proteger a privacidade dos dados dos clientes participantes.

3. Introduzindo o RPP

O RPP foi desenvolvido para mitigar simultaneamente os problemas da convergência do treinamento frente a dados não independentes ou identicamente distribuídos, privacidade dos dados dos clientes e proteção contra ataques de envenenamento. Ele alcança isso por ter as seguintes características:

- Foca o treinamento em clientes com maior valor de perda, treinando onde há maior potencial de melhoria.
- Agrega os gradientes usando média comum, possibilitando o uso de Agregação Segura (criptografia homomórfica).
- Avalia o modelo global nos próprios clientes, usando dados reais.
- Oferece a possibilidade de recuperar uma versão anterior do modelo em caso de degeneração do treinamento.
- Mantém um valor de reputação nos clientes a fim de remover clientes problemáticos do treinamento.

3.1. Definição do Problema

O cenário do FL consiste em um conjunto C de N clientes onde cada cliente c $\in$ C tem um conjunto de dados D_c com uma quantidade de amostras $|D_c|$. Antes do treinamento, o servidor inicia um modelo global (tipicamente uma rede neural) com parâmetros ω_0 que podem ser inicializados de forma aleatória, usando um modelo pré-treinado em uma tarefa semelhante ou usando dados públicos quando disponíveis para pré-treinar o modelo.

O treinamento federado consiste na execução de τ rodadas de treinamento onde, em cada rodada $t \in \{1, 2, ..., \tau\}$, um subconjunto de clientes S_t de tamanho $0 < k \leq N$ é selecionado para treinar o modelo. Cada um dos clientes c $\in S_t$ recebe o modelo ω_t e executam, em paralelo, uma época de treinamento do modelo utilizando sua base de dados local.

O método de otimização é tipicamente o método do máximo declive, também conhecido como descida do gradiente ou SGD (ver equação 1, onde α é a taxa de aprendizagem e L é a função de perda). Na estratégia FedAvg [Brendan McMahan et al. 2016], cada cliente executa várias épocas de treinamento.

$$\omega_{t+1}^{n} \leftarrow \omega_t - \alpha \cdot \nabla L(\omega_t, D_c) \tag{1}$$

Após cada rodada de treinamento nos clientes, cada cliente c envia os parâmetros do modelo treinado ω_{t+1}^{c} para o servidor. Os parâmetros são agregados em um único modelo com parâmetros ω_{t+1}, utilizando tipicamente a média ponderada pelo número de amostras no conjunto de dados de cada cliente (equação 2).

$$\omega_{t+1} \leftarrow \sum_{c=1}^{k} \frac{|D_c|}{\sum_{m=1}^{k} |D_m|} \omega_{t+1}^{c} \tag{2}$$

A expectativa é que este novo modelo agregue todo o conhecimento obtido pelo treinamento de cada modelo individual de cada cliente. Esta expectativa, entretanto, pode ser difícil de ser obtida quando os clientes são muito diferentes uns dos outros.

3.2. Seleção de Clientes

A seleção de clientes é uma parte crucial do FL. Limitações na largura de banda para a transferência de informações entre o servidor e os clientes representam um gargalo que compromete a escalabilidade da participação de todos os clientes em todas as rodadas. [Cho et al. 2020] demonstrou como a seleção de clientes com maior valor de perda pode acelerar a convergência da aprendizagem ao direcionar o treinamento em clientes com maior oportunidade de contribuição.

Este método de seleção começa selecionando um número maior de clientes k' ($k < k' \leq N$). Cada um dos k' clientes recebe os pesos do modelo ω_t e avalia o modelo usando sua base de dados. Cada cliente retorna para o servidor o valor de perda encontrado para o modelo e o servidor escolhe os k clientes com os maiores valores de perda para o treinamento.

3.3. Valor de reputação no RPP

Para proteger o treinamento evitando que clientes maliciosos sejam selecionados, o RPP propõe a adição de valores de reputação dos clientes.

No RPP, a reputação de um cliente é representada por um valor numérico que é usado como peso na escolha inicial dos clientes para avaliação.

O RPP aproveita a avaliação do modelo feita pelos clientes na etapa de seleção para avaliar a última rodada do treinamento e detecta um ataque ao notar uma piora significativa nas métricas recebidas.

Cada cliente inicia o treinamento com o valor máximo igual a 1 (visão otimista, onde todos os clientes são considerados, a priori, confiáveis) e esse valor é atualizado a medida que o cliente participa de rodadas onde ocorre uma piora nas métricas do modelo.

A grande vantagem desta abordagem consiste em usar os valores de perda recebidos dos clientes para avaliar a rodada anterior, usando dados reais e eliminando a necessidade de bancos de dados públicos para avaliação, que podem estar indisponíveis ou serem muito diferentes dos dados reais da tarefa em questão.

Ao receber o valor de perda dos clientes selecionados, o RPP compara a média com o valor obtido na última rodada de treinamento. Sendo e_t a estimativa de desempenho do modelo após a rodada t, é considerado que houve um ataque caso $e_t > \beta \cdot e_{t-1}$. Onde β é um hiper-parâmetro do RPP, tipicamente um pouco maior do que 1 para tolerar pequenas

pioras esporádicas no desempenho do modelo ao longo do treinamento, algo normal no treinamento de modelos.

Ao detectar uma situação adversa ($e_t > \beta \cdot e_{t-1}$), cada cliente c participante do treinamento na rodada t (c $\in S_t$) tem seu valor de reputação penalizado segundo a equação 3. Isso reduz a probabilidade do cliente ser selecionado novamente nas rodadas posteriores.

$$r_c \leftarrow r_c \cdot \delta_p^{q_c} \tag{3}$$

Onde δ_p é a taxa de penalização dos valores de reputação e é um valor menor do que 1 (quanto menor, maior a penalização nas reputações) e q_c é a quantidade de vezes seguidas que o cliente c foi penalizado (esse valor começa em 0 e é incrementado depois de toda a penalização e zerado depois de uma rodada bem sucedida).

É preciso reparar que todos os clientes participantes tem seu valor de reputação reduzido após um incidente no treinamento, tanto os clientes honestos quando o atacante responsável pela piora. O RPP faz isso para possibilitar o uso de estratégias de agregação segura, onde é impossível acessar os gradientes individuais de cada cliente para procurar pelos clientes responsáveis pela piora.

Entretanto, a penalização no valor de reputação não deve ser muito rigorosa nas primeiras ocorrências, de modo que o cliente possa ser selecionado novamente, ainda que com uma probabilidade ligeiramente inferior. O fator exponencial (q_c) faz com que as penalizações sejam cada vez mais severas a medida que um cliente participa seguidamente de rodadas problemáticas.

Quando a rodada é aprovada na avaliação ($e_t \leq e_{t+1}$), os valores de reputação dos clientes c participantes na rodada t são aumentados seguindo: $r_i \leftarrow min(1, r_i \cdot \delta_r)$, onde δ_r é a taxa de recuperação nos valores de reputação e é um pouco maior do que 1. Isso recupera possíveis penalidades sofridas indevidamente em rodadas anteriores.

Deste modo, todos os clientes participantes da rodada problemática são colocados sob suspeita, tendo oportunidades futuras para recuperar seu valor de reputação ao participar de rodadas bem sucedidas.

Ao longo do treinamento, conforme vários grupos diferentes de clientes participam do treinamento, os clientes honestos que participaram junto com um atacante terão seu valor de reputação momentaneamente reduzido, mas que será recuperado a medida que o cliente participa de outras rodadas bem sucedidas. Já o atacante, como ele sempre está envolvido em rodadas problemáticas, seu valor de reputação será consistentemente reduzido, inviabilizando cada vez mais a seleção dele para rodadas de treinamento.

A figura 1 ilustra a dinâmica de uma rodada do RPP, onde o Servidor envia os parâmetros do modelo para alguns clientes selecionados (1) que executam a avaliação do modelo usando seus dados locais (2) retornando os valores de perda para o servidor (3). Com esses valores, o servidor faz a estimativa da métrica e atualiza os valores de reputação (4). Em seguida, o servidor seleciona os clientes com menor valor de perda (5) e solicita que estes façam uma rodada de treinamento (6). Após o treinamento com os dados internos (7), os clientes retornam apenas os gradientes do modelo (8). O Servidor agrega os gradientes recebidos e aplica no modelo global gerando um novo modelo (9).

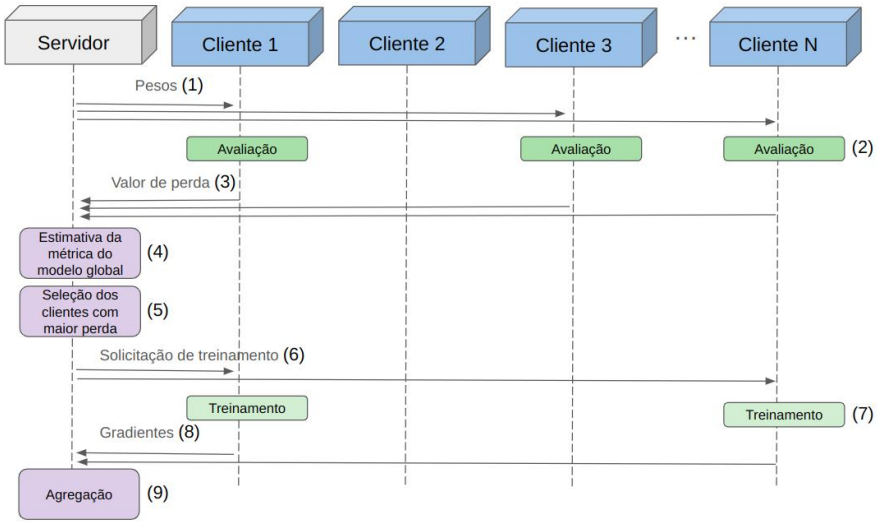


Figura 1. Passos realizados durante uma rodada de treinamento do RPP.

3.4. Recuperação Após Incidente

Os ataques de envenenamento de modelo podem ser particularmente destrutivos e prejudicar permanentemente a evolução do treinamento. Os experimentos mostraram que, mesmo após rodadas sucessivas de treinamento apenas com clientes honestos, é possível que um modelo global nunca se recupere de um ataque severo.

Para contornar isso, o RPP ainda propõe cancelar toda uma rodada em casos de piora excessiva no desempenho do modelo. Isso é feito ao salvar o último modelo que foi aprovado após a etapa de avaliação e recuperá-lo após uma rodada particularmente ruim.

O RPP define um hiper-parâmetro γ ($\beta \leq \gamma$), a taxa de piora a partir da qual o treinamento é considerado irremediavelmente prejudicado, levando ao descarte do modelo e a recuperação do último modelo aprovado. O RPP propõe uma taxa de penalização δ_P mais severa ($\delta_P < \delta_p$).

3.5. Algoritmo

O Algoritmo 1 descreve o fluxo principal do RPP. O Algoritmo requer os seguintes hiperparâmetros:

- β: Limiar de detecção de ataque ($\beta > 1$).
- γ: Limiar de detecção de ataque severo ($\gamma > \beta$).
- δ_p: Penalização de ataque não severo ($\delta_p < 1$).
- δ_P: Penalização de ataque severo ($\delta_P < \delta_p$).
- δ_r: Taxa de recuperação de reputação ($\delta_P > 1$).
- k: Quantidade de clientes que fazem o treinamento a cada rodada.
- k': Quantidade de clientes que são selecionados para avaliar o modelo.

No algoritmo, a cada rodada t é feita a seleção de clientes, cada cliente faz o treinamento local em paralelo e os resultados são agregados em um único modelo global.

O algoritmo 2 detalha como é feita a seleção dos clientes, onde é feita uma seleção inicial de k' clientes que recebem o modelo global e retornam a métrica. A métrica é avaliada a fim de proteger o modelo em caso de ataques. Caso o modelo seja aprovado,

Algorithm 1 Algoritmo Principal do RPP

TreinamentoFederado($\beta, \gamma, \delta_P, \delta_p, \delta_r, k, k'$)**:**

$\omega_0 \leftarrow InicializacaoAleatoria()$
$C = \{c_1, c_2, ..., c_N\}$ ▷ Conjunto dos clientes
$R = \{r_i = 1 | 1 \leq i \leq N\}$ ▷ Valores de reputação
$Q = \{q_i = 1 | 1 \leq i \leq N\}$ ▷ Contador de incidentes seguidos
$e* \leftarrow None$ ▷ Métrica do último modelo aprovado
$w* \leftarrow None$ ▷ Último modelo aprovado
while t in 1, 2, 3,..., τ **do**
  $S^t \leftarrow SelecionaClientes$ ▷ Ver algoritmo 2
  while c in S_t **do** ▷ Em cada cliente em paralelo
    $\omega_t^c \leftarrow TreinamentoLocal(cliente = c, modelo = \omega_{t-1})$
  end while
  $\omega_{t+1} \leftarrow \sum_{c \in S_t} \frac{|D_c|}{\sum_{c' \in S_t} |D_{c'}|} \omega_{t+1}^c$ ▷ Agregação
end while

os clientes com maior valor de perda são retornados para o treinamento. Caso a última rodada tenha sido descartada, o processo de seleção é recomeçado sem necessitar de reavaliar o modelo recuperado (dado que ele já foi aprovado anteriormente).

Algorithm 2 Algoritmo da Seleção de Clientes

SelecionaClientes

$S'_t \leftarrow$ SubConjunto(clientes=C, tamanho=k', pesos=R)
while c in S'_t **do** ▷ Em cada cliente em paralelo
  $loss_c \leftarrow$ Avalia(ω_{t-1})
end while
if Primeira vez avaliando nesta rodada **then**
  $e_t \leftarrow$ Media($\{loss_c | c \in S'_t\}$) ▷ Estimativa da métrica do modelo na rodada t
  Avalia_E_Recupera(e_t) ▷ Ver algoritmo 3
  if modelo anterior foi descartado **then**
    Recomeçar Seleção de Clientes.
  end if
end if
Retorna os k clientes com maior valor de loss

O algoritmo 3 contém o processo de avaliação e possível recuperação do modelo. Essa função recebe a estimativa da métrica do modelo atual e compara com os limiares, rebaixando ou elevando o valor das reputações dos clientes que participaram da rodada anterior (S_{t-1}) conforme o resultado da avaliação.

4. Análise

Esta seção detalha os resultados obtidos do RPP quando comparados ao FedAvg [Brendan McMahan et al. 2016] (abordagem base da literatura), PoC [Cho et al. 2020] (abordagem com rápida convergência) e duas propostas de técnicas de agregação citadas por [Yin et al. 2018] para proteger contra ataques: Média Podada e Mediana. As cinco

Algorithm 3 Algoritmo da Avaliação do Treinamento e Recuperação de Modelo

AvaliaERecupera(e_t)

if $t > 0$ e $e_t > e^* \cdot \beta$ **then** ▷ Métrica piorou acima do limiar tolerado
  if $e_t > e^* \cdot \gamma$ **then** ▷ Modelo é considerado como irrecuperável
    $\omega_t \leftarrow \omega*$ ▷ Recupera o último modelo aprovado
    while c in S_{t-1} **do**: ▷ Para cada cliente participante da última rodada
      $r_c \leftarrow r_c \cdot \delta_P^{q_i}$ ▷ Aplica a penalização severa
      $q_c \leftarrow q_c + 1$ ▷ Incrementa o contador de penalização
    end while
  else if $e_t \leq e_{t-1} * \gamma$ **then** ▷ Modelo piorou, mas sem necessidade de descarte
    while c in S_{t-1} **do**: ▷ Para cada cliente participante da última rodada
      $r_c \leftarrow r_c \cdot \delta_p^{q_i}$ ▷ Aplica a penalização branda
      $q_c \leftarrow q_c + 1$ ▷ Incrementa o contador de penalização
    end while
  end if
else if $t = 0$ ou $e_t \leq e_{t-1} \cdot \beta$ **then** ▷ Modelo aprovado
  $w* \leftarrow w_{t-1}$ ▷ Salva o modelo aprovado
  $e* \leftarrow e_{t-1}$ ▷ Salva a métrica do modelo aprovado
  while c in S_{t-1} **do**: ▷ Para cada cliente participante da última rodada
    $r_c \leftarrow r_c \cdot \delta_r$ ▷ Recupera o valor de reputação
    $q_c \leftarrow 0$ ▷ Zera a quantidade de rebaixamentos seguidos
  end while
end if

abordagens foram submetidas a um processo de FL em cenários com e sem a presença de atacantes.

Para os experimentos com a presença de atacantes, 3 clientes (0,9% do total de 345 clientes) tiveram seu comportamento modificado a fim de praticar ataques de envenenamento de modelo ou envenenamento de dados.

4.1. Base de dados e Modelo

A base de dados foi construída fazendo uso da ferramenta LEAF [Caldas et al. 2018], um *framework* para FL capaz de gerar um conjunto de dados sintético com as propriedades adequadas para validar o RPP e compará-lo com outras estratégias.

O conjunto de dados sintético de treinamento foi gerado contendo 76843 amostras distribuídas em 345 clientes de modo não-iid. Os dados contém 10 *features* numéricas e uma saída categórica com 6 classes. Além disso, um outro conjunto de dados foi gerado seguindo a mesma distribuição geral com 19369 amostras para ser usado como métrica geral do modelo.

Para realizar a classificação, foi usado um modelo Perceptron Multicamada (rede neural) com 3 camadas intermediárias com 256 neurônios lineares retificados (Relu) e uma saída softmax com 6 neurônios. O otimizador escolhido foi a descida estocástica do gradiente (SGD) com taxa de aprendizado de 10^{-3}.

Todos os experimentos partiram do mesmo modelo inicial, eliminando uma das

fontes de variação da análise deste trabalho.

4.2. Experimentos

O primeiro teste do RPP validou o funcionamento frente a somente clientes honestos. Neste cenário, com a ausência de ataques, é esperado que todas as rodadas sejam aprovadas e que nenhum cliente tenha seu valor de reputação alterado (dado que todos são honestos).

O primeiro tipo de ataque simulado consistiu em um ataque de envenenamento de modelo onde um pequeno grupo de participantes tenta atacar o modelo enviando valores aleatórios ao invés de gradientes legítimos. Neste caso, os valores enviados seguem uma distribuição normal com média zero e 3 versões de desvio padrão: severo, médio e baixo.

Outra versão de ataque testada foi o envio de gradientes legítimos, mas escalados em duas diferentes abordagens: o atacante pode multiplicar os gradientes por um valor maior do que 1 de modo a tentar impor o resultado do seu treinamento no modelo global; o atacante pode tentar multiplicar o gradiente por um número negativo de modo a forçar o modelo global para a direção oposta para a arruinar o treinamento.

Nos resultados, foi observado que, para o cenário estudado, o treinamento estabilizava após 300 rodadas, então esse foi escolhido como o limite de rodadas dos experimentos.

Em cada cenário, foram conduzidos 3 experimentos. Em cada rodada, 7 clientes participavam do treinamento e, no caso das abordagens com seleção inicial (RPP e do PoC), 32 clientes eram selecionados para avaliar o modelo. Para o RPP, foram usados os valores $\beta = 1,15, \gamma = 1,25, \delta_p = 0,98, \delta_P = 0,85 e \delta_r = 1,12$, que foram definidos empiricamente. Para a Média Podada, a agregação realizou a média dos gradientes após eliminar os dois maiores e os dois menores valores.

4.3. Resultados

Quando todos os clientes são honestos, as abordagens que concentram o treinamento nos clientes com maior valor de perda (RPP e PoC) tiveram um desempenho melhor, tanto em termo da velocidade de convergência quanto em termo da acurácia após 300 rodadas de treinamento federado, alcançando 80% de acurácia contra 70% do FedAvg e da Média Podada que começaram com uma ascensão rápida, mas reduzindo seu crescimento depois de 45 rodadas (ver figura 2). A abordagem de agregação usando a mediana foi a pior, sendo muito lenta na convergência e atingindo um platô de 56% de acurácia ao fim de 200 rodadas.

A ausência de ataques e de falso positivos na detecção também fez com que os valores de reputação de todos os clientes ficassem inalterados, mantendo o valor máximo de 1.

O teste seguinte adicionou apenas 3 clientes maliciosos (representando menos de 0.9% dos clientes). Estes clientes não executam o treinamento, mas enviam apenas ruído aleatório como se fossem gradientes obtidos de treinamento (ataque de envenenamento de modelo). Foi usado ruído normal do média zero e três níveis de desvio padrão.

Para valores baixos de desvio padrão (ruído leve com desvio padrão 0,1), o próprio processo de treinamento absorve esses ruídos, não comprometendo de modo significativo

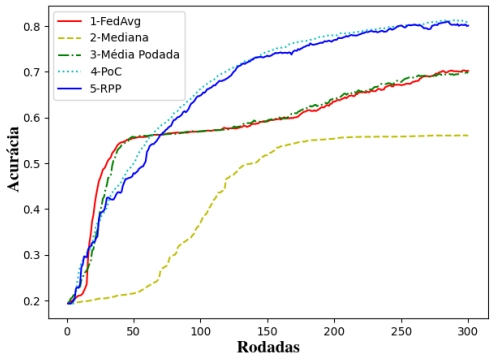


Figura 2. Resultado da acurácia do treinamento somente com clientes honestos.

o resultado final do modelo (ver figura 3.a). Os valores de reputação também não foram alterados neste experimento.

Para ruídos de intensidade moderada (desvio padrão 1), a figura 3.b mostra que as abordagens FedAvg, Média Podada e PoC até conseguem recuperar-se de um ataque, mas a custos de um pequeno atraso no processo de treinamento. O RPP mostra um crescimento mais consistente ao recuperar o último modelo aprovado.

Neste experimento com ruído médio, os valores de reputação mínimo dentre os clientes honestos foi de 0,74, enquanto o valor mínimo dentre os atacantes foi de 0,01, demonstrando que o algoritmo foi bem sucedido ao discriminar clientes honestos de atacantes ao longo de 300 rodadas.

Para ruídos de intensidade alta (desvio padrão maior do que 100), mesmo para poucos clientes, o modelo é irremediavelmente corrompido no FedAvg, Média Podada e no PoC. O RPP, por outro lado, consegue identificar o ataque e recuperar o último modelo aprovado, fazendo com que o RPP seja a única abordagem capaz de convergir frente a presença de menos de 1% de clientes maliciosos (figura 3.c) e atingir uma acurácia final maior do que 80%.

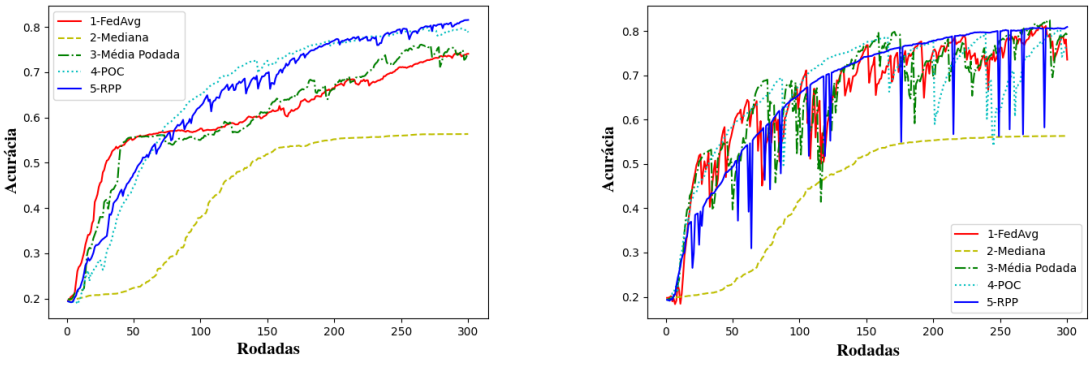
Nos experimentos com ruído de alta intensidade, o valor mínimo de reputação dos clientes honestos foi de 0,85 enquanto o valor mínimo dentre os atacantes foi de 0,003.

Em todos os níveis de ruído, a abordagem que agrega os gradientes usando a Mediana mostrou um comportamento igual, sendo insensível às tentativas de ataque. Entretanto, ao ignorar gradientes muito discrepantes, ela parece ignorar também informações relevantes, comprometendo sua convergência para valores de acurácia mais elevados.

Resultados similares foram obtidos ao multiplicar os gradientes por um valor de modo a tentar aumentar a importância do treinamento na etapa de agregação ou para apontar os gradientes na direção inversa para maximizar a perda do modelo.

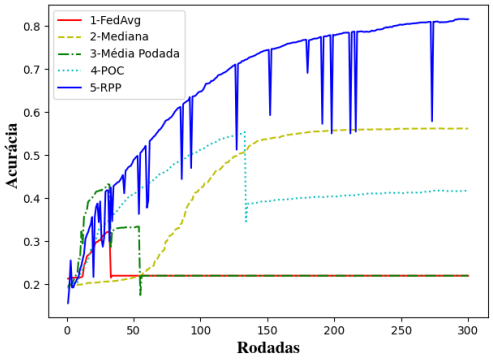
A figura 4.a, mostra o resultado quando 3 atacantes tentaram privilegiar seus gradientes ao multiplicá-los por 100, todas as abordagens convergem, mas o RPP foi o menos prejudicado, chegando a 80% de acurácia ao final das 300 rodadas em comparação ao PoC, Média Podada e o FedAvg que ficaram um pouco acima de 70%.

Essa convergência é esperada uma vez que os atacantes estão ajustando o modelo



(a) Ruído baixo

(b) Ruído médio



(c) Ruído severo

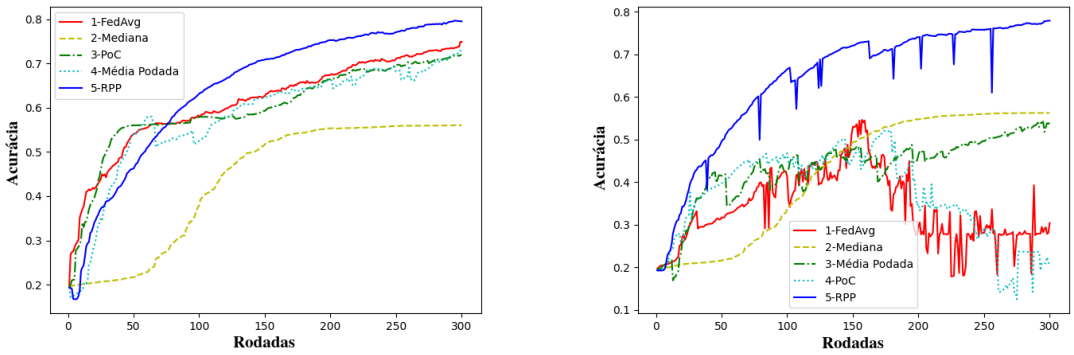
Figura 3. Resultado da acurácia do treinamento com 3 atacantes enviando ruído ao invés de gradientes legítimos.

na direção correta, mas apenas tentando obter uma vantagem frente aos clientes honestos.

A figura 4.b mostra a evolução da acurácia quando 3 atacantes multiplicam os gradientes por um valor negativo para maximizar a perda ao invés de reduzi-la. Neste cenário, o FedAvg e o PoC falharam em convergir, a Média Podada ofereceu uma proteção bem limitada chegando a somente um pouco acima de 50% de acurácia e somente o RPP foi capaz de convergir a valores próximos de 80% por conta de sua capacidade de identificar os ataques mais prejudiciais e recuperar o último modelo aprovado.

Neste experimento, o valor mínimo de reputação para clientes honestos foi de 0,61, enquanto o valor mínimo para os atacantes foi de 0,38. Embora os valores mínimos não tenham se diferenciado significativamente após 300 rodadas, destaca-se que a recuperação dos clientes honestos pode levar mais tempo, dependendo da frequência com que um cliente treinou com atacantes. Observou-se que a reputação dos atacantes diminui mais rapidamente, porém com menor frequência ao longo das rodadas, uma vez que valores mais baixos tornam a seleção do cliente para avaliação mais difícil.

A estratégia da Mediana teve o mesmo comportamento nos casos de gradientes escalados, sendo insensível aos ataques, mas sem uma evolução interessante em sua acurácia.



(a) Valor positivo (b) Valor negativo

Figura 4. Resultado da acurácia do treinamento com 3 atacantes escalando os gradientes.

5. Considerações Finais

Este trabalho introduziu o RPP, uma abordagem para o Aprendizado Federado que ataca três diferentes problemas: velocidade de convergência ao focar o treinamento em clientes com maior valor de perda, privacidade de dados ao permitir o uso de Criptografia Homomórfica e resistência a ataques de envenenamento de modelos ao propor avaliação com possibilidade de recuperação de um estado anterior.

Foram feitos experimentos comparando o RPP com diferentes técnicas da literatura onde o RPP conseguiu apresentar uma evolução consistente mesmo na presença de atacantes que corrompem seus gradientes, um cenários onde as outras técnicas falharam em convergir.

A estratégia de avaliar os clientes via reputação também foi capaz de reduzir a seleção de atacantes ao mesmo tempo que preserva a probabilidade de selecionar clientes honestos, mesmo sem acessar os valores de gradientes de cada cliente individualmente.

Como trabalho futuro, propõe-se avaliar como oferecer proteção frente a técnicas de ataque ainda mais sofisticadas como o ataque de portas dos fundos [Bagdasaryan et al. 2020] e estratégias que visam comprometer a privacidade mesmo na presença de agregação segura [Kariyappa et al. 2023].

Referências

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.

Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv e-prints*, pages arXiv–1602.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Cho, Y. J., Wang, J., and Joshi, G. (2020). Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*.

Kang, J., Xiong, Z., Niyato, D., Xie, S., and Zhang, J. (2019). Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714.

Kariyappa, S., Guo, C., Maeng, K., Xiong, W., Suh, G. E., Qureshi, M. K., and Lee, H.-H. S. (2023). Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In *International Conference on Machine Learning*, pages 15884–15899. PMLR.

Lyu, L., Yu, H., and Yang, Q. (2020). Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*.

Nair, A. K., Raj, E. D., and Sahoo, J. (2023). A robust analysis of adversarial attacks on federated learning environments. *Computer Standards & Interfaces*, page 103723.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. (2019). Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*.

Wang, H., Kaplan, Z., Niu, D., and Li, B. (2020). Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1698–1707. IEEE.

Xie, C., Koyejo, O., and Gupta, I. (2018). Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR.

Zhang, W., Wang, X., Zhou, P., Wu, W., and Zhang, X. (2021). Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access*, 9:24462–24474.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.