MO809A - Tópicos em Computação Distribuída

Seminário: “Quando o Aprendizado Federado encontra a Blockchain: Um novo paradigma do Aprendizado Distribuído”.

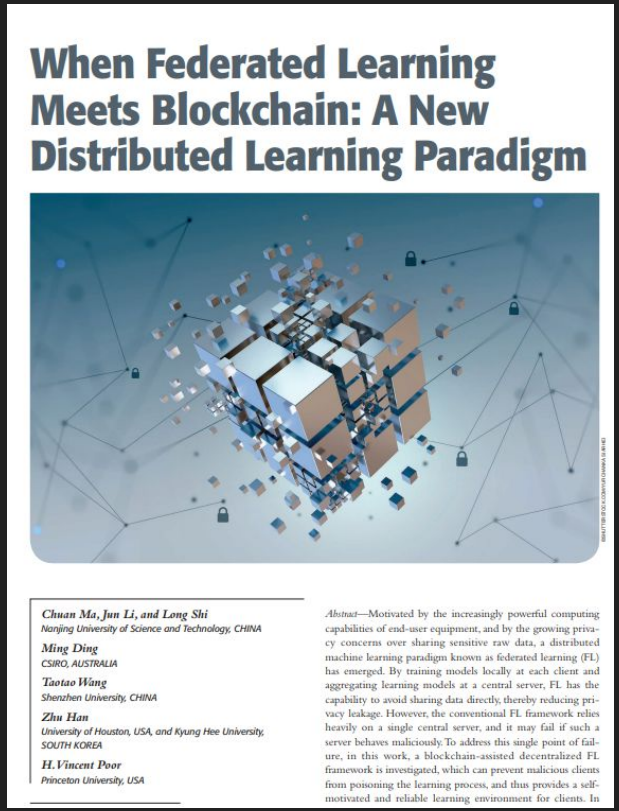
Wilson Bagni Junior - 010097

Artigo de base

When Federated Learning Meets Blockchain: A New Distributed Learning Paradigm

Chuan Ma; Jun Li; Long Shi; Ming Ding; Taotao Wang; Zhu Han; H. Vincent Poor

Publicação: IEEE Computational Intelligence Magazine (Volume: 17, Issue: 3, August 2022)



Trabalhos anteriores

[5] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020, doi: 10.1109/LCOMM.2019.2921755.

[6] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020, doi: 10.1109/TII.2019.2942190.

[7] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in *Proc. 2019 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, pp. 151–159, doi: 10.1109/BIGCOM.2019.00030.

[8] P. K. Sharma, J. H. Park, and K. Cho, "Blockchain and federated learning-based distributed computing defence framework for sustainable society," *Sustain. Cities Soc.*, vol. 59, p. 102,220, 2020.

[9] S. Wang, "Blockfedml: Blockchained federated machine learning systems," in *Proc. 2019 Int. Conf. Intell. Comput., Autom. Syst. (ICICAS)*, pp. 751–756, doi: 10.1109/ICICAS48597.2019.00162.

[10] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2964–2973, Apr. 2021, doi: 10.1109/TII.2020.3007817.

[11] S. Otoum, I. Al Ridhawi, and H. Mouftah, "Blockchain-supported federated learning for trustworthy vehicular networks," Dec. 2020, pp. 1–6.

Trabalhos anteriores

- Propuseram e estudaram integrações entre Aprendizado Federado e Blockchain
- Analisaram algumas questões de segurança e vazamento de informações propondo métodos de controle/mitigação dos problemas
- No geral, tiveram como suposição que a Federação de entidades que compunham a rede (ou parte dela) eram ‘confiáveis’.

Este artigo

O artigo em questão propõe uma estrutura baseada em blockchain (BLockchain-Assisted DEcentralized Federated Learning - BLADE-FL) que resolve a vulnerabilidade da centralização de dados em um servidor central e também procura mitigar a ação de usuários mal intencionados que estejam participando da rede.

Introdução e Motivação

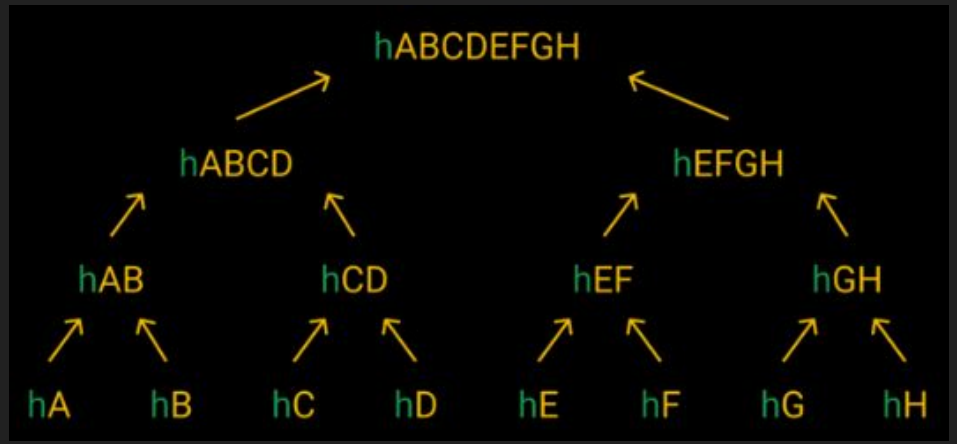
O Aprendizado Federado aproveita o poder computacional de equipamentos da “borda” da rede treinando modelos de maneira distribuída, agregando-os de forma centralizada, muitas vezes evitando vazamento de informações sensíveis.

A concentração realizada na agregação desses modelos num servidor central pode representar um ponto de vulnerabilidade para este paradigma (servidor comprometido/vulnerável, falta de transparência, viés etc)

Para lidar com esse ponto de vulnerabilidade é possível utilizar uma estrutura descentralizada de blockchain. Nela, a agregação dos modelos ocorre de forma descentralizada, auditável e as tarefas de treino do modelo e mineração/verificação dos blocos da blockchain são divididas entre todos os participantes.

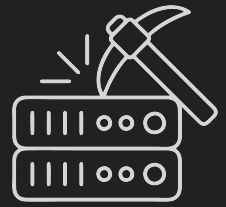
Mas antes de começarmos…o que é blockchain?

É uma estrutura de dados baseada em *Merkle Tree* (árvore de hash) que permite, com uso de recursos de criptografia, manter a integridade de todos os blocos de dados da estrutura. Ideal para uma rede P2P onde os participantes precisam compartilhar e validar informações de forma independente.



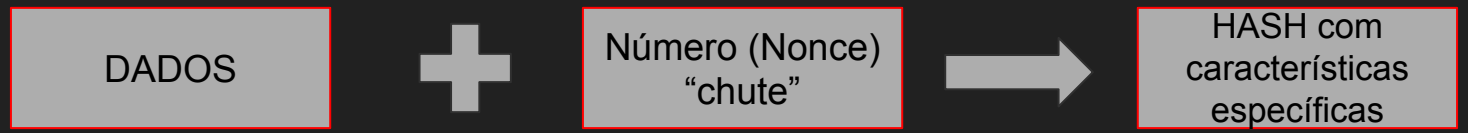
https://academy.binance.com/pt/articles/merkle-trees-and-merkle-roots-explained

Como um novo bloco é adicionado na rede?



Um novo bloco é adicionado a essa rede através de mecanismos de consenso.

O mecanismo *Proof of Work* (PoW) consiste em resolver um problema computacionalmente difícil. O nó que resolver esse problema ganha o direito de anexar um bloco na rede depois que isso for verificado por outros nós.



Uma vez verificado o novo bloco é incorporado e aceito na rede. Esse processo é chamado de mineração e geralmente as redes de blockchain recompensam o nó da rede que consegue agregar um bloco.

https://www.investopedia.com/tech/how-does-bitcoin-mining-work/

O framework BLADE-FL

Composto por 3 camadas: **1- Rede**, **2 - Blockchain** e **3 - Aplicação**

1 - Rede: Rede do tipo *peer-to-peer* (P2P): Composta por entidades que podem publicar tarefas e treinar modelos

2 - Blockchain: Diferente de uma comunidade confiável, as entidades, além de treinar modelos, mineram os blocos para publicar os resultados agregados (auditam os resultados)

3 - Aplicação: Através de um contrato inteligente (*Smart Contract* - SM) os eventos do aprendizado federado são executados e controlados.

Como funciona um contrato inteligente?

O contrato inteligente é uma coleção de códigos de dados que é implementada usando transações assinadas criptograficamente na rede blockchain.

Ele pode conter requisitos a serem atendidos e cláusulas autogerenciáveis de forma que executa transações de forma automática assim que determinadas condições são atendidas.

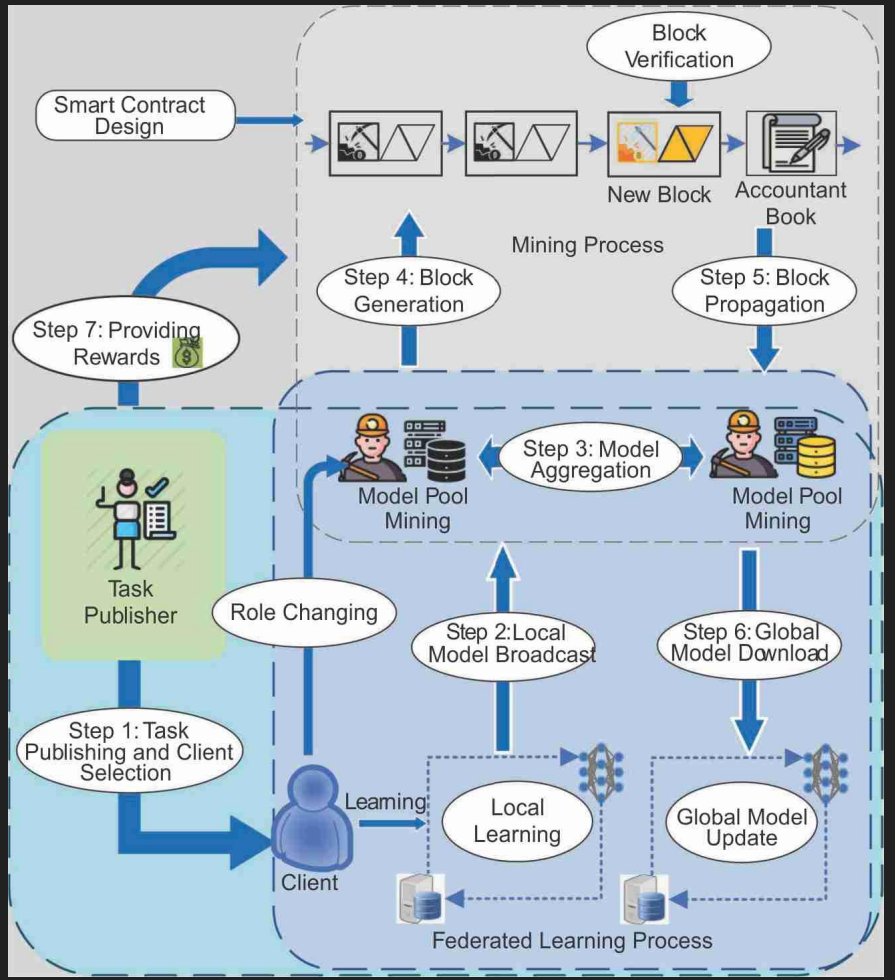
Workflow - Visão Geral

P2P Network Layer

Blockchain Layer

Application Layer

(Cada etapa será detalhada nos próximos slides)



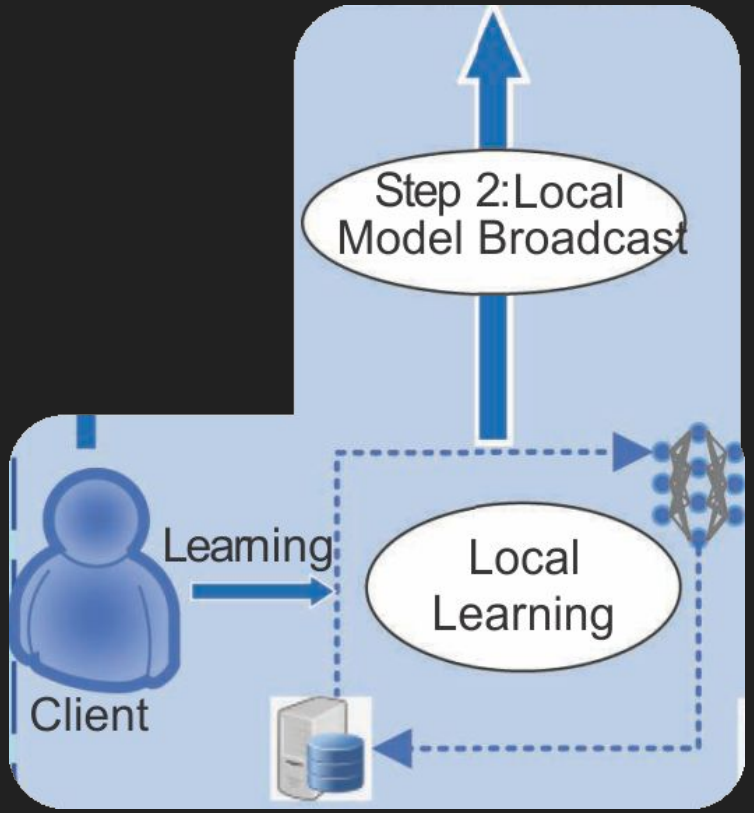
Workflow em detalhes

Passo 1 - Um usuário publica uma tarefa de Aprendizado Federado através de um contrato inteligente e deposita uma recompensa ($$$) de incentivo.

O contrato inteligente seleciona nós da rede disponíveis para participar da tarefa de aprendizado.

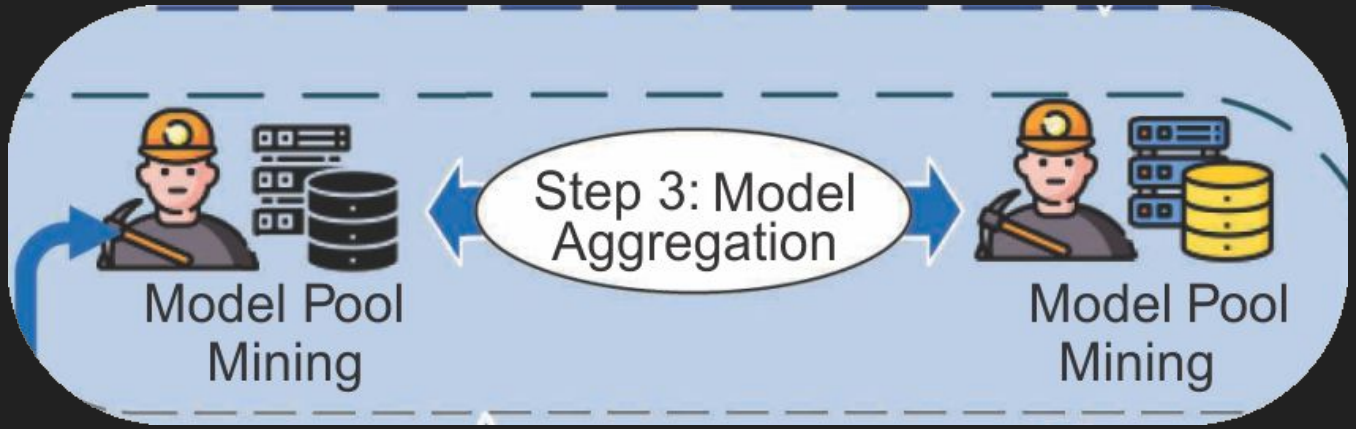
Workflow em detalhes

Passo 2 - Transmissão de modelos locais: Cada cliente treina localmente seu modelo e transmite atualizações locais do modelo pela rede P2P.



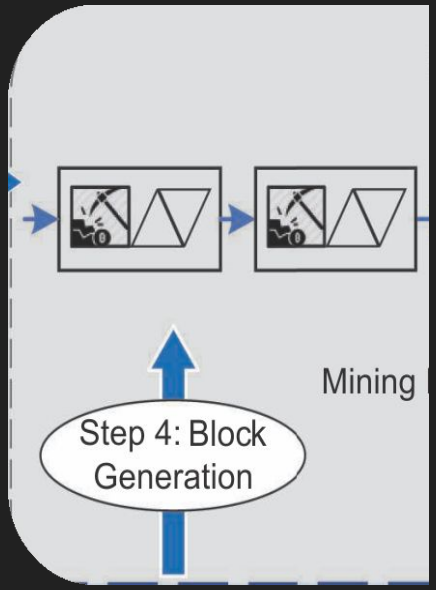
Workflow em detalhes

Passo 3 - Agregação de modelos: Ao receber atualizações locais de outros nós, cada cliente atualiza o modelo global com base nas regras do contrato inteligente.



Workflow em detalhes

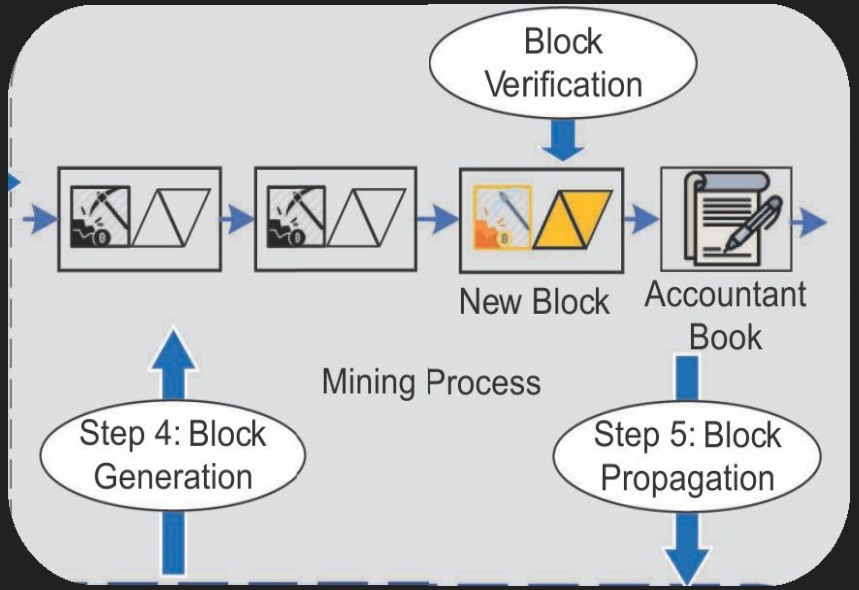
Passo 4 - Geração de bloco. Os clientes que estavam treinando modelos trocam de função e passam a atuar como mineradores: gerando um bloco ou recebendo um bloco gerado por outro cliente. Quando um minerador gera um bloco, os outros mineradores verificam o bloco gerado.



Workflow em detalhes

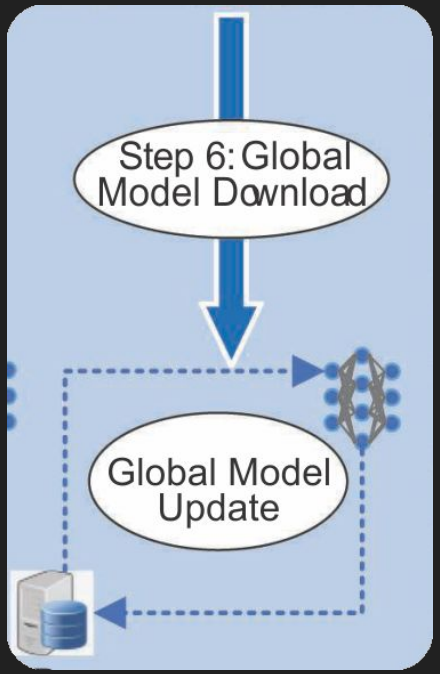
Passo 4 - Geração de bloco. Os clientes que estavam treinando modelos trocam de função e passam a atuar como mineradores: gerando um bloco ou recebendo um bloco gerado por outro cliente. Quando um minerador gera um bloco, os outros mineradores verificam o bloco gerado.

Passo 5 - Propagação do bloco. Se um bloco é verificado por uma maioria de clientes, ele é anexado à cadeia da blockchain e aceito por toda rede.



Workflow em detalhes

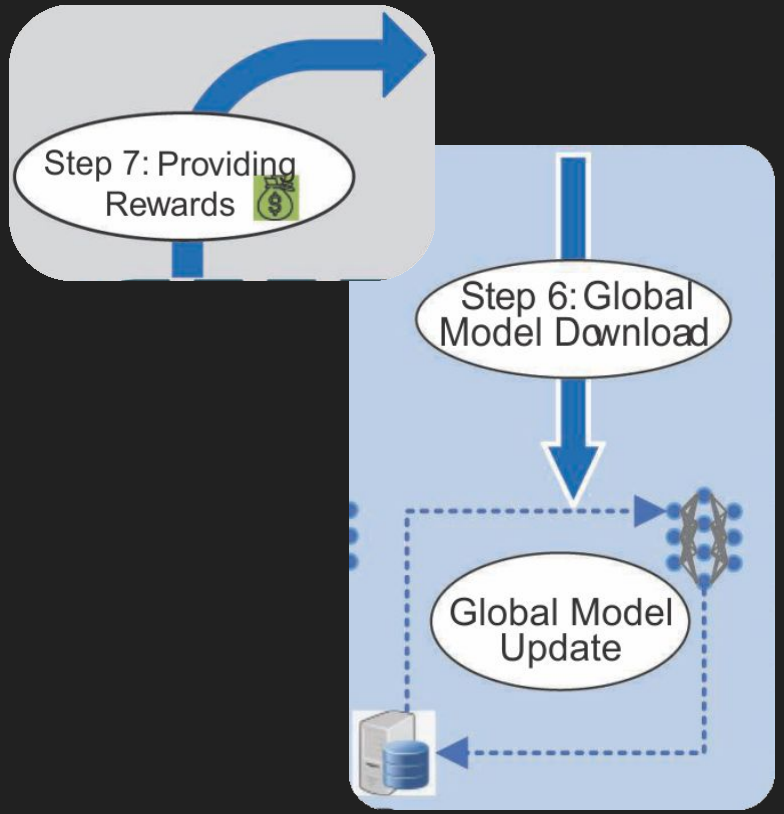
Passo 6 - O modelo global é atualizado e repassado a todos os clientes antes da próxima rodada de aprendizado.



Workflow em detalhes

Passo 6 - O modelo global é atualizado e repassado a todos os clientes antes da próxima rodada de aprendizado.

Passo 7 - A recompensa depositada no passo 1 é distribuída entre os participantes de acordo com a relevância da atuação



Funções do Contrato Inteligente

1 - Publicar a tarefa de aprendizado para todos na rede. O contrato contém os requisitos (tamanho dos dados, acurácia etc), regras para agregar os dados e a recompensa a ser paga.

2 - “Leiloar” a atividade. Os nós da rede interessados na recompensa fazem lances indicando seus custos e capacidades e o contrato inteligente selecionará aqueles mais vantajosos e confiáveis (há índices de confiabilidade com base em atividades anteriores)

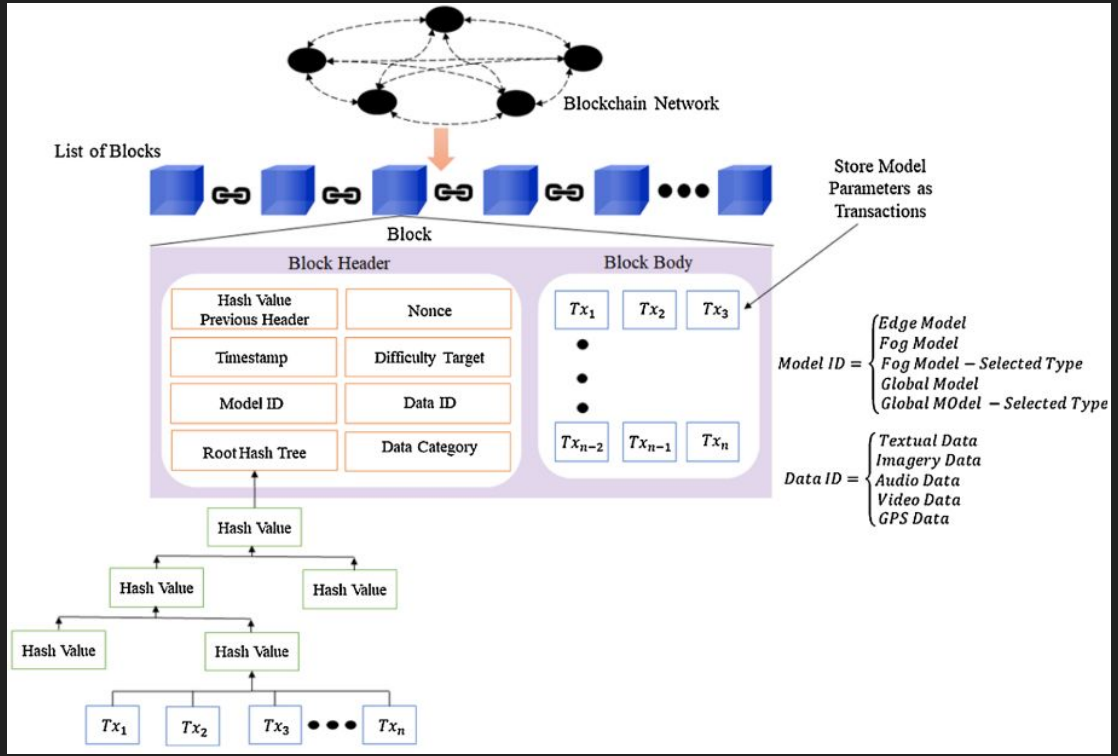
3 - Agregar os resultados à blockchain e recompensar os nós que participaram do processo de aprendizagem

Pontos-chave do framework

1 - Atualização e upload do modelo local: Os nós da rede que vencem o ‘leilão’ atualizam, de forma paralela, um modelo local de aprendizado de máquina usando o modelo global e suas amostras de dados locais e transmitem seu modelo local atualizado na rede. Essa comunicação pode ser feita na rede P2P com uso de protocolos do tipo “gossip”, por exemplo.

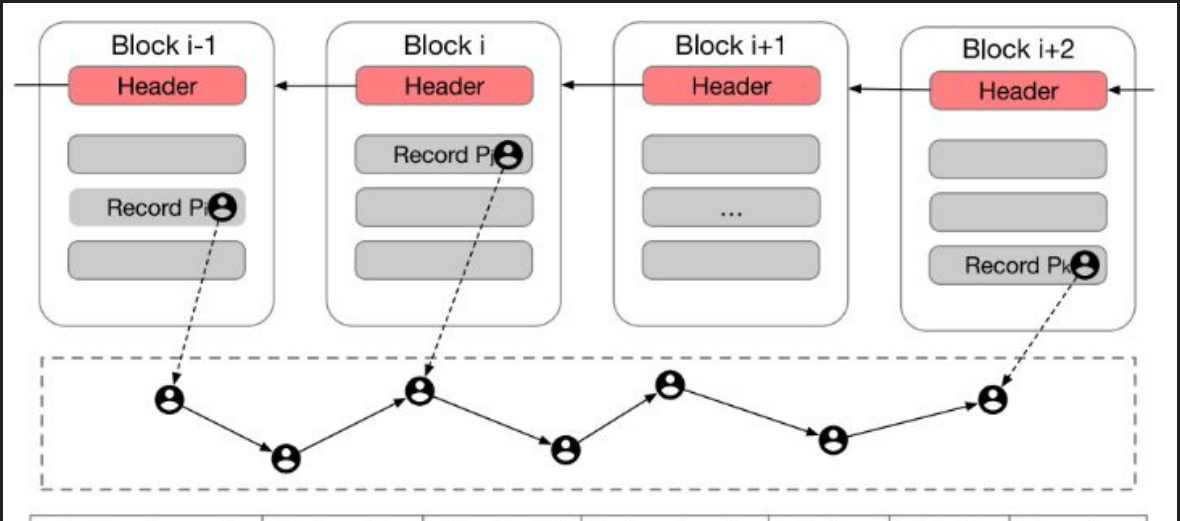
2 - Agregação dos modelos: depois de coletar os modelos enviados, cada cliente calcula as atualizações globais do modelo de acordo com a regra de agregação prevista pelo contrato inteligente. O armazenamento pode ser feito na blockchain como um ‘livro razão’

Pontos-chave do framework (continuação)



ARTIGO: BLOCKCHAIN AND FEDERATED LEARNING-BASED DISTRIBUTED COMPUTING DEFENCE FRAMEWORK FOR SUSTAINABLE SOCIETY

Pontos-chave do framework (continuação)



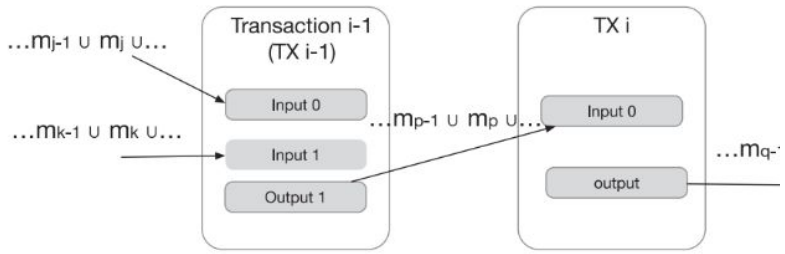
Retrieval record	Hash pointer	Provider ID	Data categories	Data type	Data Size	Timestamp
Data sharing record	Request type	Requester ID	Data categories	Data type	Data Size	Timestamp

ARTIGO: BLOCKCHAIN AND FEDERATED LEARNING FOR PRIVACY-PRESERVED DATA SHARING IN INDUSTRIAL IoT

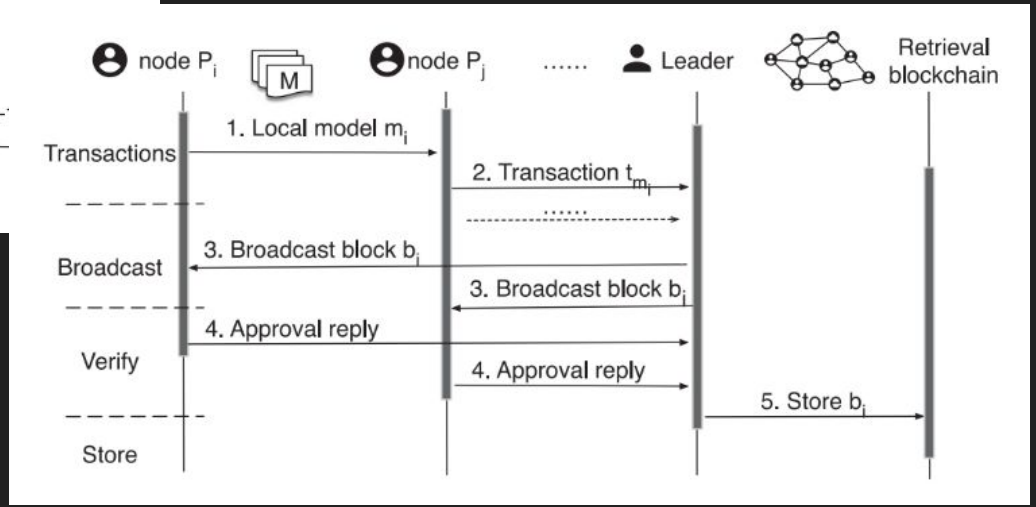
Pontos-chave do framework (continuação)

TABLE I
RECORD TUPLE OF A MODEL TRANSACTION

Hash pointer	Owner id	Receiver id	Model data	MAE	Timestamp



ARTIGO: BLOCKCHAIN AND FEDERATED LEARNING FOR PRIVACY-PRESERVED DATA SHARING IN INDUSTRIAL IoT



Pontos-chave do framework (continuação)

3 - Publicação dos modelos: blocos gerados são compartilhados na rede e verificados. Nesta etapa são aplicados protocolos de consenso (por exemplo o PoW - *Proof of Work*) para a distribuição de modelos entre os clientes da rede e para a decisão se o bloco será anexado ou rejeitado.

Caso o bloco seja aceito, a atualização da blockchain é repassada a todos os usuários da rede, as recompensas são pagas e um novo ciclo de treinamento pode começar.

Investigações adicionais (1- privacidade)

Na etapa inicial de treino dos modelos há maior chance de vazamento de informações sensíveis. Trabalhos anteriores lidaram com essa questão separando usuários mineradores daqueles que treinavam os modelos e também admitindo que toda ou parte da rede era confiável.

Este artigo investigou a diminuição do vazamento de informações pela adição de ruído (por exemplo, ruído Gaussiano ou Laplaciano) nos dados a ser efetuada por cada ente de rede.

A quantidade de ruído a ser colocada pode ser um parâmetro de escolha a ser analisado pelo contrato inteligente.

Investigações adicionais (1- privacidade)

Com a adição de ruído haverá prejuízo no aprendizado, e será necessário encontrar a melhor relação de custo benefício em cada caso.

A performance pode ser melhorada também se o ruído a ser adicionado diminuir a cada rodada de treino.

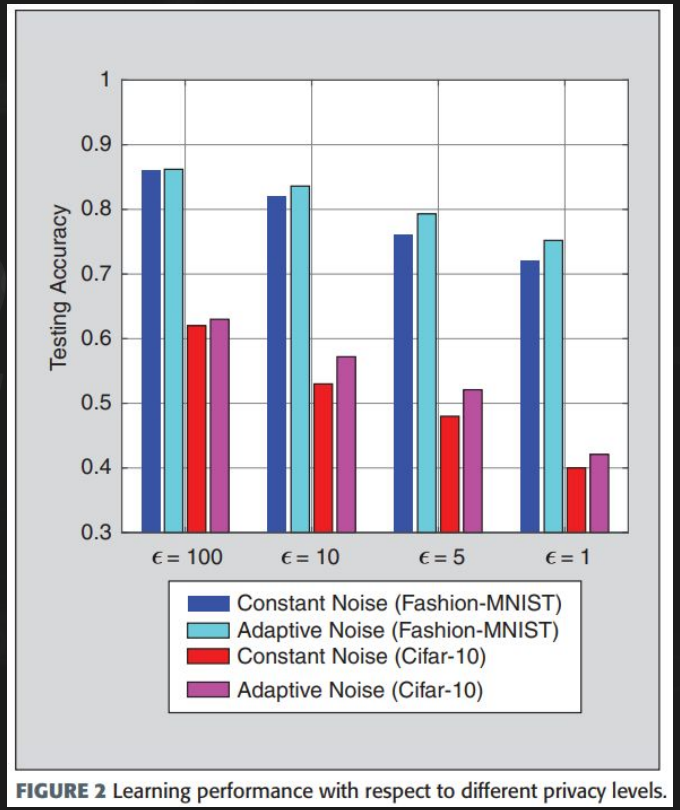


FIGURE 2 Learning performance with respect to different privacy levels.

Investigações adicionais (2- usuários preguiçosos)

Um determinado usuário da rede pode tentar ganhar as recompensas por apenas replicar modelos treinados por outros usuários.

Ao invés de treinar o modelo, quando ele recebe um modelo para verificar, adiciona ruído ou altera levemente os parâmetros e repassa para a rede o modelo como se fosse dele.

Para mitigar esse tipo de problema, o artigo estuda o uso de pseudo-ruído.

Investigações adicionais (2- usuários preguiçosos)

Ao verificar o modelo, o usuário inicialmente verifica se ele pode ter sido plagiado. (verificação de correlação cruzada).

Se essa verificação atingir determinado limiar, é detectado o plágio e o usuário preguiçoso é punido na rede.

TABLE I Detection rate with different PN sequence powers for Fashion-MNIST and Cifar-10 datasets.

SNR	9 dB	6 dB	3 dB
Fashion-MNIST	0.931	0.989	0.999
Cifar-10	0.925	0.975	0.996

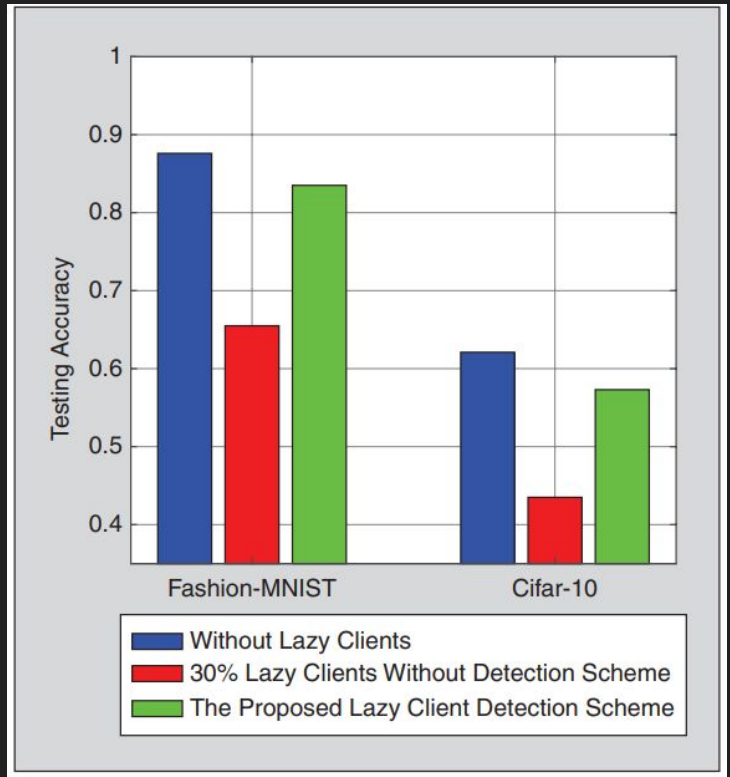


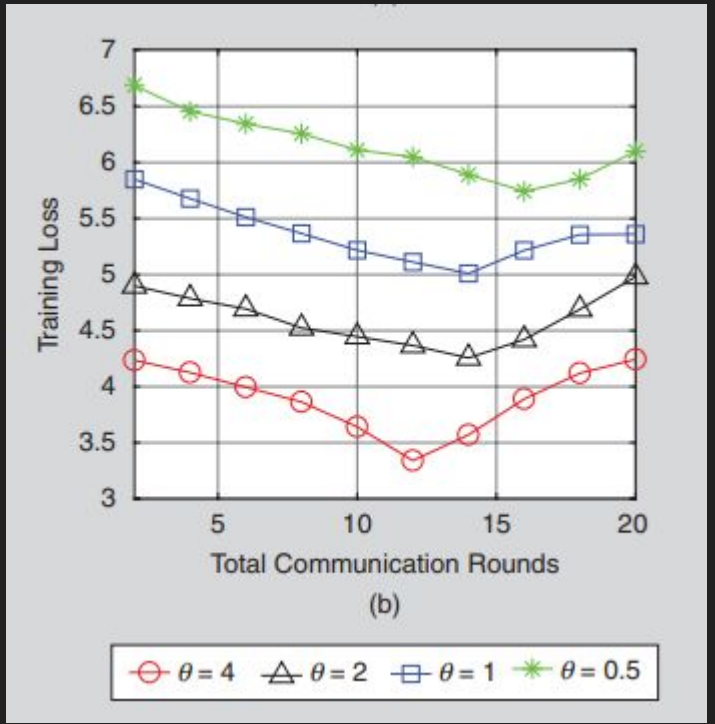
FIGURE 4 Learning performance with/without lazy client detection.

Investigações adicionais (3- alocação de recursos)

O artigo estuda também o impacto da razão entre os recursos que cada cliente usa com Treino e Mineração.

No gráfico, quanto maior theta, mais recursos são gastos no treino do modelo. Os dados foram treinados usando a base Cifar-10.

Esta razão impacta diretamente na quantidade de rounds ótima para treino dos modelos.



Dúvidas?