SIDERAU: Sistema de Detecção de Fraudes em Cartões de Crédito baseado em Aprendizado Federado

Jackson S. R. da Silva[1], **Joahannes B. D. da Costa**[2], **Helder M. N. da S. Oliveira**[3]

[1] Universidade Federal do ABC (UFABC), Santo André, Brasil
[2]Universidade Federal de São Paulo (UNIFESP), São José dos Campos, Brasil
[3]Universidade de São Paulo (USP), São Paulo, Brasil

jackson.roque@ufabc.edu.br, joahannes.costa@unifesp.br
helderoliveira@ime.usp.br

Abstract. *Detecting credit card fraud has become a critical challenge for financial institutions due to the growth of digital transactions and the increasing sophistication of fraudster strategies. This work proposes SIDERAU, a system that integrates Federated Learning (FL) and IDS for collaborative fraud identification while preserving privacy. The results indicate that SIDERAU achieved an F1-Score of 78.9% and a recall of 86.4%. The architecture demonstrated a 28% reduction in execution time (124s vs. 172s) and a 5-fold decrease in local memory load per node (21.5 MB vs. 107.6 MB), enabling real-time detection.*

Resumo. *A detecção de fraudes em cartões de crédito tornou-se um desafio crítico para instituições financeiras devido ao crescimento das transações digitais e à crescente sofisticação das estratégias utilizadas por fraudadores. Este trabalho propõe o SIDERAU, sistema que integra Aprendizado Federado (FL) e IDS para identificação colaborativa de fraudes preservando a privacidade. Os resultados indicam que o SIDERAU alcançou F1-Score de 78,9% e Recall de 86,4%. A arquitetura demonstrou redução de 28% no tempo de execução (124s vs 172s) e diminuição de 5 vezes na carga de memória local por nó (21,5 MB vs 107,6 MB), viabilizando detecção em tempo real.*

1. Introdução

Fraudes em cartões de crédito representam um problema significativo para instituições financeiras e plataformas de comércio eletrônico [Akour et al. 2025]. O crescimento das transações digitais ampliou as oportunidades para atividades fraudulentas, exigindo mecanismos de detecção cada vez mais eficientes [Sharma and Gwale 2023]. Métodos tradicionais baseados em regras apresentam limitações diante da evolução constante das estratégias de fraude, motivando o uso de técnicas como o Aprendizado de Máquina, do inglês *Machine Learning (ML)*, para identificação automática de padrões suspeitos em transações financeiras [Rzayeva and Malekzadeh 2022].

Sistemas de Detecção de Intrusão, do inglês *Intrusion Detection Systems (IDSs)*, têm sido amplamente utilizados para monitorar atividades suspeitas e identificar comportamentos anômalos em diferentes domínios computacionais [Khraisat et al. 2019]. Tradicionalmente aplicados à segurança de redes, os IDSs têm sido empregados em sistemas financeiros para analisar eventos transacionais e detectar possíveis atividades fraudulentas [Btoush et al. 2023]. Nesse cenário, cada transação pode ser interpretada como um

evento a ser analisado por um modelo de detecção, permitindo a identificação de comportamentos suspeitos em tempo real.

Diversos algoritmos de aprendizado supervisionado têm sido aplicados nesse domínio, incluindo regressão logística, florestas aleatórias e redes neurais artificiais [Varmedja et al. 2019]. Entretanto, a detecção de fraude apresenta desafios importantes, especialmente devido ao forte desbalanceamento dos conjuntos de dados, nos quais transações fraudulentas representam apenas uma pequena fração das operações realizadas [Makki et al. 2019]. Esse cenário pode levar modelos de ML a apresentarem viés em favor da classe majoritária, comprometendo a capacidade de identificar fraudes. Além disso, a privacidade das informações financeiras limita o compartilhamento de dados entre instituições, dificultando o treinamento colaborativo de modelos de detecção.

O Aprendizado Federado, do inglês *Federated Learning (FL)*, tem sido investigado como uma alternativa promissora para o treinamento distribuído de modelos de ML preservando a confidencialidade dos dados [de Souza et al. 2024] [Fares et al. 2026]. Essa abordagem permite que diferentes participantes (chamados de clientes) treinem modelos localmente utilizando seus próprios dados e compartilhem apenas os parâmetros do modelo com um servidor de agregação. Dessa forma, torna-se possível explorar conhecimento distribuído entre múltiplas instituições sem a necessidade de centralizar bases de dados sensíveis.

Nesse contexto, considerando os desafios impostos pelo desbalanceamento dos dados e pelas restrições de privacidade no compartilhamento de informações financeiras, este trabalho apresenta o **SIDERAU**, um sistema de detecção de fraudes em cartões de crédito baseado em FL. O SIDERAU integra a privacidade proporcionada pelo FL aos mecanismos de detecção do IDS. A abordagem permite o treinamento colaborativo entre múltiplas entidades, mantendo os dados localmente e, assim, mitigando riscos associados à exposição de informações sensíveis. Adicionalmente, o sistema incorpora estratégias de balanceamento de dados, fundamentais em cenários de detecção de fraudes, nos quais a classe de interesse é significativamente minoritária. A ausência de tratamento adequado para esse desbalanceamento pode induzir os modelos a favorecerem a classe majoritária, resultando em baixa capacidade de detecção de transações fraudulentas. Dessa forma, a proposta busca não apenas preservar a privacidade dos dados, mas também melhorar a eficácia dos modelos na identificação de padrões raros e críticos.

As contribuições deste trabalho concentram-se em três aspectos principais. i) propõe-se uma arquitetura de detecção de fraudes que integra FL a um IDS baseado em redes neurais, permitindo o treinamento colaborativo de modelos de ML entre múltiplas entidades financeiras sem a necessidade de compartilhamento de dados sensíveis. ii) investiga-se o impacto da aplicação de estratégia de tratamento de desbalanceamento de dados em ambientes federados, comparando o uso de geração de dados sintéticos por meio do algoritmo *Synthetic Minority Over-sampling Technique (SMOTE)*. iii) realiza-se uma avaliação experimental comparando cenários de treinamento federado e centralizado, analisando métricas de desempenho relevantes para sistemas de detecção de fraude, como precisão, *recall*, *F1-score* e *AUC-ROC*, além de discutir aspectos relacionados à heterogeneidade de dados e à eficiência computacional da abordagem proposta.

Diferente das abordagens tradicionais que exigem a centralização de dados

sensíveis, o sistema proposto foca na eficiência e privacidade. As contribuições deste estudo incluem a demonstração de um modelo distribuído capaz de ser 48 segundos mais rápido que o centralizado, mantendo uma taxa de detecção superior a 86%.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a arquitetura do sistema proposto, bem como detalhes de seu funcionamento. A Seção 4 apresenta a metodologia experimental. A Seção 5 discute os resultados obtidos. Por fim, a Seção 6 apresenta as conclusões do trabalho e direções para trabalhos futuros.

2. Trabalhos Relacionados

A detecção de fraudes em cartões de crédito tem sido amplamente investigada utilizando técnicas de ML aplicadas à análise de transações financeiras. Diversos algoritmos supervisionados têm sido explorados nesse contexto, incluindo regressão logística, árvores de decisão e redes neurais artificiais. Varmedja *et al.* [Varmedja et al. 2019] realizaram uma análise comparativa de diferentes algoritmos de ML para detecção de fraude, demonstrando que modelos supervisionados podem alcançar bom desempenho quando treinados em conjuntos de dados representativos. De forma semelhante, Rzayeva *et al.* [Rzayeva and Malekzadeh 2022] propuseram uma abordagem híbrida combinando redes neurais profundas e o algoritmo *K-Nearest Neighbors*, obtendo melhorias na identificação de transações fraudulentas.

Um dos principais desafios nesse domínio refere-se ao forte desbalanceamento dos conjuntos de dados, uma vez que transações fraudulentas representam apenas uma pequena fração do total de operações financeiras. Makki *et al*. [Makki et al. 2019] investigaram diferentes estratégias de classificação para lidar com dados desbalanceados em sistemas de detecção de fraude, demonstrando que técnicas de balanceamento podem melhorar o desempenho dos modelos. Zaffar *et al*. [Zaffar et al. 2023] exploraram o uso de aprendizado de subespaços combinado com classificação de uma classe (*one-class classification*), modelando o comportamento normal das transações e tratando desvios como possíveis fraudes.

Além dos desafios relacionados ao desbalanceamento de dados, a privacidade das informações financeiras representa um fator crítico no desenvolvimento de sistemas de detecção de fraude. Abordagens baseadas em criptografia têm sido propostas para permitir o processamento de dados sensíveis sem exposição direta das informações originais. Nugent [Nugent 2022] investigou o uso de criptografia homomórfica para detecção de fraude em cartões de crédito, permitindo a análise de transações financeiras em formato criptografado.

Mais recentemente, o FL tem sido explorado como uma alternativa para o treinamento colaborativo de modelos de detecção de fraude preservando a privacidade dos dados. Yang *et al.* [Yang et al. 2019] apresentaram os fundamentos do aprendizado federado e suas aplicações em diferentes domínios, incluindo sistemas financeiros. No contexto específico de detecção de fraude, Aurna *et al.* [Aurna et al. 2023] investigaram a aplicação de FL combinado com técnicas de aprendizado profundo, analisando o impacto de diferentes estratégias de amostragem no desempenho dos modelos.

A detecção de anomalias em ambientes distribuídos também tem sido investigada no contexto de segurança de redes. Romani *et al.* [Romani et al. 2026] realizaram um

estudo comparativo entre FL e modelos centralizados para detecção de ataques Negação de Serviço Distribuído (DDoS) em ambientes *Internet of Things (IoT)*, utilizando dados de tráfego de rede gerados por *botnets* como o Mirai. Os resultados indicaram que modelos centralizados podem apresentar maior desempenho em termos de acurácia, enquanto abordagens federadas oferecem vantagens relacionadas à privacidade e escalabilidade. No entanto, o estudo considera dados de tráfego de rede e cenários de detecção de intrusão, que apresentam características distintas das transações financeiras. Além disso, o impacto do desbalanceamento extremo de classes não é explorado de forma explícita.

Abordagens híbridas também têm sido exploradas em ambientes distribuídos. Fares *et al.* [Fares et al. 2026] propuseram uma arquitetura que integra HIDPS, NIDS e FL para detecção e mitigação de ataques DDoS em redes IoT, combinando análise baseada em assinaturas e anomalias com mecanismos de resposta ativa nos dispositivos. Os resultados indicam melhorias na detecção de tráfego malicioso e na disponibilidade dos serviços. No entanto, a abordagem é voltada para dados de tráfego de rede e mitigação em tempo real, não contemplando cenários de classificação supervisionada com dados tabulares altamente desbalanceados, como no contexto de fraude financeira.

Diferentemente dos trabalhos existentes, este estudo investiga de forma comparativa o impacto de diferentes estratégias de tratamento de desbalanceamento em cenários federados de detecção de fraude. Para isso, avaliamos quatro configurações experimentais distintas envolvendo treinamento federado e centralizado, com e sem geração de dados sintéticos, permitindo analisar de forma sistemática o comportamento dessas estratégias em ambientes distribuídos.

3. Arquitetura do Sistema Proposto

Esta seção apresenta o SIDERAU, sistema de detecção de fraudes em cartões de crédito proposto neste trabalho, baseado na integração entre FL e um IDS. O objetivo do SIDERAU é permitir o treinamento colaborativo de modelos de ML entre múltiplas entidades financeiras, preservando simultaneamente a confidencialidade dos dados transacionais.

A Figura 1 apresenta uma visão geral da arquitetura do SIDERAU. O sistema segue o paradigma cliente-servidor típico de ambientes de FL, no qual diferentes participantes treinam modelos localmente utilizando seus próprios dados e compartilham apenas os parâmetros atualizados com um servidor agregador responsável pela construção do modelo global.

No SIDERAU, cada cliente federado representa uma entidade participante do sistema, como instituições financeiras, operadoras de cartão ou gateways de pagamento. Cada participante possui acesso a um conjunto de dados transacionais correspondente às operações realizadas em sua infraestrutura. Esses dados permanecem armazenados localmente e não são compartilhados diretamente com outros participantes, atendendo aos requisitos de privacidade e conformidade regulatória.

O treinamento no SIDERAU ocorre de forma distribuída por meio do paradigma de FL. Inicialmente, um modelo global é inicializado no servidor agregador e distribuído aos clientes participantes. Cada cliente realiza treinamento local utilizando seu conjunto de dados e atualiza os parâmetros do modelo com base nas amostras disponíveis local-

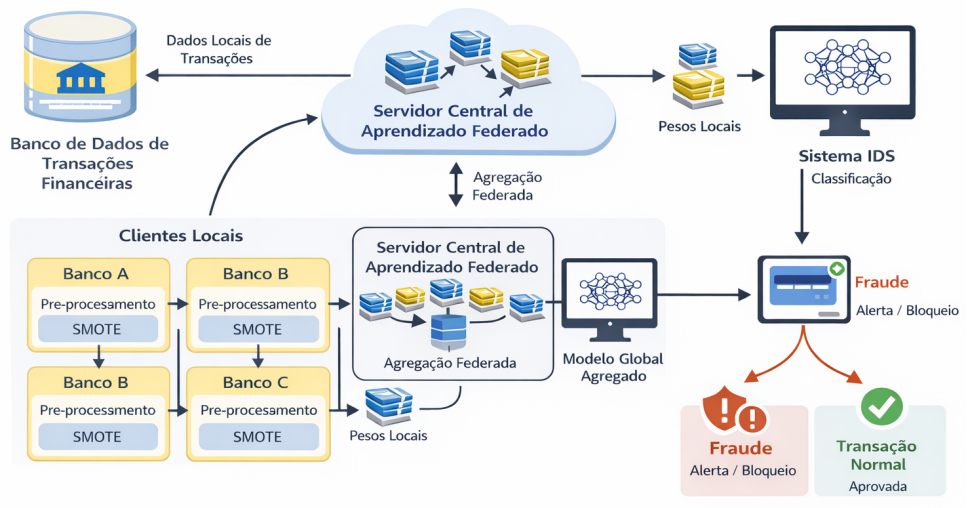


Figure 1. Arquitetura geral do SIDERAU para detecção de fraudes baseada em aprendizado federado.

mente. Após essa etapa, os parâmetros atualizados são enviados ao servidor, que realiza o processo de agregação para produzir uma nova versão do modelo global.

A agregação dos modelos locais no SIDERAU é realizada utilizando o algoritmo *Federated Averaging (FedAvg)*, amplamente adotado em sistemas de FL. Considere um conjunto de K clientes participantes do treinamento, onde cada cliente k possui n_k amostras de dados. Seja $n = \sum_{k=1}^{K} n_k$ o número total de amostras consideradas na rodada de treinamento. Após o treinamento local, cada cliente produz um vetor de parâmetros $\mathbf{w}_k$. O servidor agregador calcula o novo modelo global $\mathbf{w}$ por meio de uma média ponderada dos parâmetros locais, conforme definido na Equação 1.

$$\mathbf{w} = \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{w}_k \tag{1}$$

Esse processo permite ao SIDERAU combinar o conhecimento aprendido em diferentes conjuntos de dados distribuídos entre os participantes, capturando padrões mais representativos de comportamento fraudulento sem a necessidade de centralização das bases de dados originais.

Um desafio central no SIDERAU, assim como em outros sistemas baseados em FL, refere-se à heterogeneidade das distribuições de dados locais, caracterizando cenários *non-IID*. Em aplicações financeiras reais, diferentes instituições podem apresentar perfis distintos de clientes, padrões de transação heterogêneos e taxas de incidência de fraude significativamente diferentes, o que pode afetar a convergência do treinamento e a qualidade do modelo global.

Neste trabalho, optou-se por avaliar o SIDERAU inicialmente em um cenário *IID*, com o objetivo de isolar o impacto das estratégias de tratamento de desbalanceamento de dados no contexto federado. Para isso, o conjunto de dados foi particionado de forma uniforme entre os clientes, garantindo distribuições estatisticamente similares em termos de volume de dados e proporção de fraudes.

Além do processo de treinamento distribuído, o SIDERAU incorpora um IDS responsável pela análise das transações financeiras em tempo real. Nesse contexto, cada transação é tratada como um evento a ser classificado como legítimo ou potencialmente fraudulento.

O núcleo do IDS no SIDERAU é composto por um modelo de rede neural treinado por meio do processo federado. Esse modelo recebe como entrada os atributos das transações financeiras e produz uma probabilidade associada à classe fraudulenta. Quando essa probabilidade ultrapassa um limiar de decisão predefinido, a transação é classificada como suspeita e pode ser encaminhada para mecanismos adicionais de verificação, como autenticação reforçada ou bloqueio preventivo.

O funcionamento do SIDERAU pode ser dividido em duas fases principais. A primeira corresponde ao treinamento federado, no qual os clientes colaboram para atualizar o modelo global ao longo de múltiplas rodadas de comunicação. A segunda corresponde à fase de detecção, na qual o modelo treinado é utilizado pelo IDS para analisar novas transações em tempo real. Essa separação permite que o SIDERAU seja continuamente atualizado à medida que novos padrões de fraude são observados nos dados locais dos participantes, além de possibilitar a inclusão de novos clientes sem necessidade de modificações estruturais.

O processo de treinamento no SIDERAU segue um fluxo iterativo de comunicação entre o servidor agregador e os clientes participantes. Em cada rodada, o modelo global é distribuído aos clientes, que realizam treinamento local utilizando seus dados privados. Após essa etapa, apenas os parâmetros atualizados são enviados ao servidor, que realiza a agregação por meio do algoritmo FedAvg.

O Algoritmo 1 descreve formalmente o processo de treinamento federado adotado no SIDERAU. O treinamento ocorre de forma iterativa ao longo de múltiplas rodadas de comunicação entre clientes e servidor agregador. Em cada rodada, os clientes treinam o modelo localmente e enviam apenas os parâmetros atualizados ao servidor, que realiza a agregação para atualizar o modelo global.

Algorithm 1: Treinamento Federado para Detecção de Fraude

Entrada: clientes C, dados locais D_k, número de rodadas T
Saída: modelo global treinado w
Início:
inicializar modelo global w;
for $t \leftarrow 1$ **to** T **do** `// rodadas de treinamento`
  servidor envia w_t para todos os clientes $c_k \in C$;
  $W \leftarrow \emptyset$; `// lista de modelos locais`
  foreach $c_k \in C$ **do** `// treinamento local`
    treinar modelo local w_k usando dados locais D_k;
    atualizar pesos do modelo por gradiente descendente;
    W.append(w_k); `// armazenar modelo local`
  `// agregação federada (FedAvg)`
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k$;
return modelo global treinado w;

4. Metodologia

Esta seção descreve a metodologia adotada para avaliar o desempenho da arquitetura proposta. O processo de treinamento segue o procedimento apresentado no Algoritmo 1, sendo executado em um ambiente de FL ao longo de múltiplas rodadas de comunicação entre o servidor agregador e os clientes participantes. Em cada rodada, o modelo global é distribuído aos clientes, que realizam treinamento local utilizando seus dados privados e retornam apenas os parâmetros atualizados para agregação.

4.1. Conjunto de Dados e Particionamento

Os experimentos foram conduzidos utilizando um conjunto de dados público amplamente empregado na literatura de detecção de fraudes em cartões de crédito. O dataset contém 284.807 transações realizadas por titulares de cartões europeus ao longo de dois dias, das quais apenas 492 correspondem a fraudes confirmadas, resultando em uma taxa de aproximadamente 0,172%. Esse cenário caracteriza um problema altamente desbalanceado.

As transações são descritas por atributos numéricos obtidos por meio de *Principal Component Analysis (PCA)*, representados pelas variáveis $V1$ a $V28$, além dos atributos *Time* e *Amount*. A variável alvo é representada pelo atributo *Class*, no qual o valor 0 indica transações legítimas e o valor 1 indica fraude. Para garantir reprodutibilidade, os dados foram divididos em 80% para treinamento e 20% para teste utilizando `random_state=42`.

No contexto do FL, o conjunto de treinamento foi particionado uniformemente entre cinco clientes simulados. Cada cliente recebeu aproximadamente um quinto das amostras, mantendo a distribuição original das classes. Essa partição foi realizada antes da aplicação de qualquer técnica de balanceamento, garantindo independência entre os dados locais e evitando vazamento de informação entre clientes.

4.2. Pré-processamento e Balanceamento

Antes do treinamento, os dados foram submetidos a uma etapa de pré-processamento. Os atributos numéricos foram normalizados utilizando *StandardScaler*, ajustado exclusivamente sobre os dados de treinamento e posteriormente aplicado ao conjunto de teste.

Para tratar o desbalanceamento entre classes, foram avaliadas duas estratégias. A primeira consiste na aplicação do algoritmo SMOTE de forma local em cada cliente, ou seja, após a partição federada. Dessa forma, a geração de amostras sintéticas ocorre de maneira independente em cada subconjunto de dados, preservando os requisitos de privacidade do FL. Foram utilizados os parâmetros padrão do SMOTE, com $k = 5$ vizinhos mais próximos.

A segunda estratégia baseia-se em aprendizado sensível ao custo, no qual diferentes pesos são atribuídos às classes durante o treinamento. Os pesos foram definidos de forma inversamente proporcional à frequência das classes em cada cliente e incorporados diretamente à função de perda *Binary Crossentropy*, aumentando a penalização associada à classe minoritária.

4.3. Configuração do FL e Modelo

O ambiente de FL foi implementado utilizando o *framework* Flower[1], com cinco clientes e um servidor central responsável pela agregação dos modelos. O treinamento foi conduzido ao longo de 20 rodadas de comunicação. Em cada rodada, os clientes receberam o modelo global atualizado e realizaram treinamento local por uma época (*local epoch*).

A adoção de uma única época local por rodada foi motivada pela necessidade de controlar o desvio entre os modelos locais e o modelo global, reduzindo efeitos de divergência durante a agregação. Embora múltiplas épocas locais possam acelerar a convergência inicial, elas tendem a aumentar a heterogeneidade entre clientes. Dessa forma, a escolha adotada favorece a estabilidade do treinamento e permite uma análise mais controlada do impacto das estratégias de balanceamento.

O modelo de classificação utilizado no IDS consiste em uma rede neural do tipo *Multilayer Perceptron (MLP)*, selecionada por sua capacidade de capturar relações não lineares. A arquitetura possui três camadas ocultas com 64, 32 e 16 neurônios, respectivamente, utilizando função de ativação ReLU. Para mitigar *overfitting*, foram aplicadas camadas de *Dropout* com taxa de 0,2. A camada de saída utiliza ativação sigmoide para produzir a probabilidade de fraude. O treinamento foi realizado utilizando o otimizador Adam, com taxa de aprendizado de 10^{-3}, função de perda *Binary Crossentropy* e mini-batches de tamanho 32.

Os experimentos foram conduzidos em um ambiente computacional equipado com processador Intel Core i5, 16 GB de memória RAM e sistema operacional Linux. A implementação foi realizada em Python, utilizando bibliotecas como Scikit-Learn, TensorFlow/Keras e Flower.

4.4. Métricas de Avaliação

A avaliação do modelo foi realizada com base em métricas de desempenho de classificação e métricas relacionadas ao custo de comunicação e computação no ambiente federado.

A precisão (*precision*) mede a proporção de transações classificadas como fraude que são de fato fraudulentas, sendo utilizada para avaliar a incidência de falsos positivos. O *recall* quantifica a capacidade do modelo de identificar corretamente as fraudes existentes, refletindo a taxa de detecção da classe minoritária.

F1-score corresponde à média harmônica entre precisão e *recall*, fornecendo uma medida equilibrada do desempenho do modelo em cenários desbalanceados. Já a métrica AUC-ROC avalia a capacidade do classificador de distinguir entre as classes ao longo de diferentes limiares de decisão.

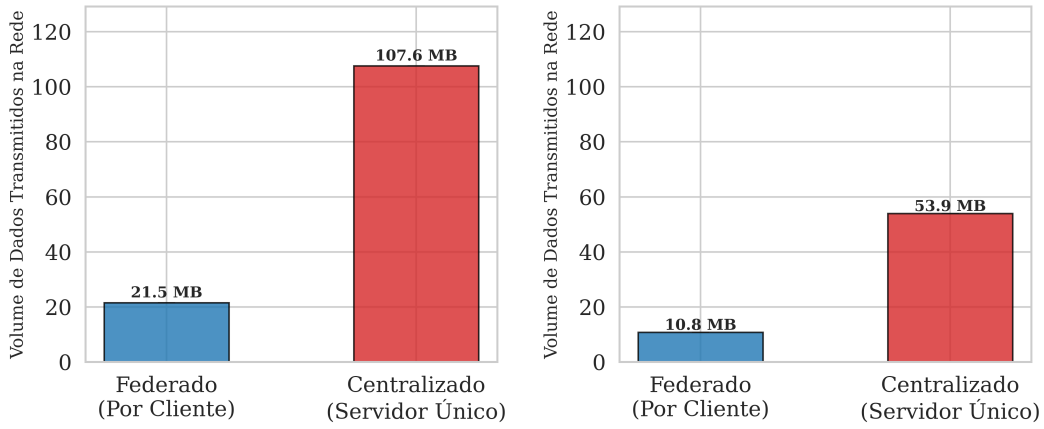
Além das métricas de classificação, o volume de bytes transmitidos foi utilizado para mensurar o custo de comunicação entre os clientes e o servidor durante o processo de aprendizado federado. Por fim, o tempo de CPU foi considerado para avaliar o custo computacional do treinamento, permitindo analisar a eficiência da abordagem proposta.

[1] `https://flower.ai/`

5. Resultados

Esta seção apresenta a avaliação experimental da arquitetura proposta considerando quatro cenários: i) federado com SMOTE; ii) federado sem balanceamento; iii) centralizado com SMOTE; e iv) centralizado sem balanceamento. A análise abrange desempenho de classificação, custo de comunicação e custo computacional.

A Figura 2 apresenta o volume de dados transmitidos durante o treinamento. Observa-se que a abordagem centralizada apresenta custo significativamente superior em ambos os cenários, uma vez que requer a transferência integral do conjunto de dados ao servidor. No cenário com SMOTE, esse custo é ampliado devido ao aumento do número de amostras, resultando em crescimento proporcional do volume transmitido. Em contraste, o aprendizado federado restringe a comunicação ao envio de parâmetros do modelo, reduzindo substancialmente o tráfego de rede. Essa diferença caracteriza uma vantagem estrutural do FL, na qual o custo de comunicação cresce com o número de parâmetros do modelo, enquanto no modelo centralizado cresce com o tamanho do dataset, tornando-se menos escalável.



(a) Cenário Com SMOTE (b) Cenário Sem SMOTE

Figure 2. Volume de dados transmitidos na rede.

A Tabela 1 apresenta os resultados quantitativos do desempenho de classificação. Inicialmente, observa-se que os cenários sem balanceamento apresentam melhor compromisso entre precisão e *recall*. O modelo federado sem balanceamento atinge precisão de 0.7437 e *recall* de 0.8678, enquanto o centralizado sem balanceamento apresenta precisão ligeiramente superior (0.7469), porém com *recall* inferior (0.8491). Essa diferença indica que o modelo federado é mais sensível à classe minoritária, detectando mais fraudes, ao custo de um leve aumento em falsos positivos.

Essa relação é refletida no *F1-score*, em que o modelo centralizado sem balanceamento apresenta valor ligeiramente superior (0.7945 contra 0.7890), indicando melhor equilíbrio global, embora com menor capacidade de detecção.

Nos cenários com SMOTE, observa-se uma degradação consistente da precisão em ambos os paradigmas (queda de aproximadamente 17%), sem ganhos relevantes em *recall*. O *recall* permanece praticamente constante (variação inferior a 2%), indicando que o aumento de exemplos sintéticos não melhora a capacidade de identificação de

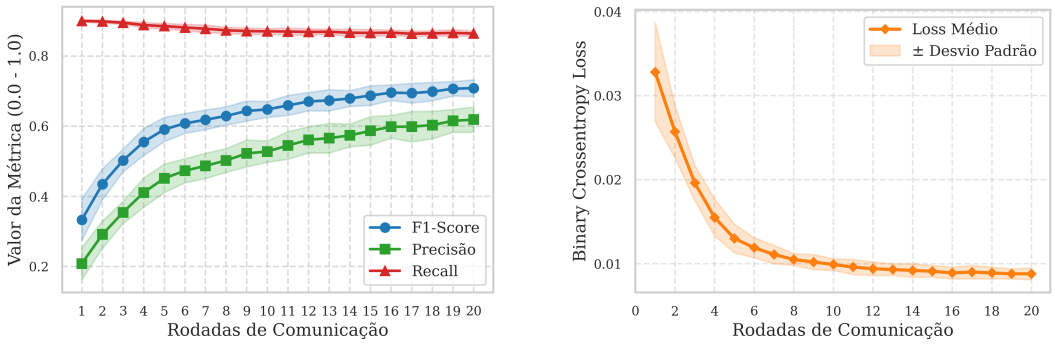
Table 1. Desempenho dos cenários experimentais.

Cenário	*Precision*	*Recall*	*F1-score*	AUC-ROC
Federado	0.7437	0.8678	0.7890	0.9779
Federado + SMOTE	0.6188	0.8642	0.7084	0.9755
Centralizado	0.7469	0.8491	0.7945	0.9790
Centralizado + SMOTE	0.6131	0.8506	0.7120	0.9683

fraudes. Como consequência, o *F1-score* também é reduzido (de aproximadamente 0.79 para 0.71), evidenciando perda de desempenho global. Esse comportamento sugere que o SMOTE introduz ruído adicional, deslocando o limiar de decisão e aumentando a taxa de falsos positivos sem benefício proporcional na detecção.

A métrica AUC-ROC apresenta valores elevados em todos os cenários (maior que 0.96), indicando boa separabilidade entre classes. Apesar do AUC elevado, a queda de precisão indica deslocamento do limiar de decisão. No entanto, a redução observada nos cenários com SMOTE confirma que o balanceamento sintético impacta negativamente a capacidade discriminativa do modelo.

A Figura 3 apresenta a evolução das métricas ao longo das rodadas. Observa-se que o modelo inicia com alto *recall* e baixa precisão, indicando um viés inicial para classificação positiva. Ao longo do treinamento, há aumento progressivo da precisão, com manutenção do *recall* em níveis elevados. Esse comportamento resulta em crescimento do *F1-score*, evidenciando ajuste das fronteiras de decisão. A convergência da função de perda ocorre de forma estável, sem oscilações significativas, indicando que o processo de agregação federada é consistente e não sofre instabilidades mesmo com múltiplos clientes.



(a) Métricas (b) Loss

Figure 3. Dinâmica de treinamento do modelo federado.

A Figura 4 evidencia variações entre os clientes em termos de precisão e recall, indicando heterogeneidade nos dados locais. Essa variação sugere diferenças nos padrões de transação e distribuição de classes entre os participantes. Apesar dessa dispersão, o desempenho agregado do modelo global permanece consistente (Tabela 1), indicando que o processo de agregação é capaz de mitigar os efeitos da heterogeneidade.

A Figura 5 ilustra a relação entre precisão e *recall* ao longo do treinamento.

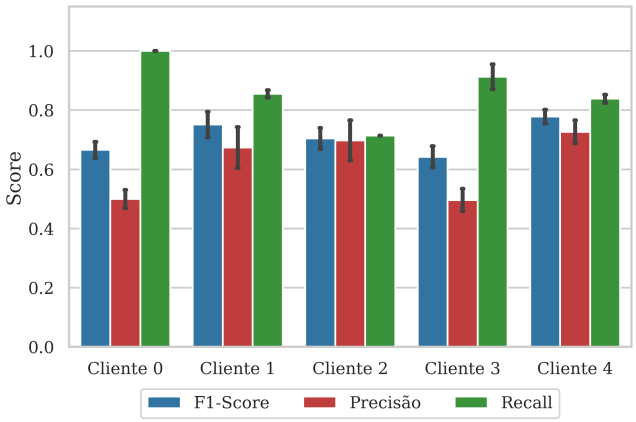
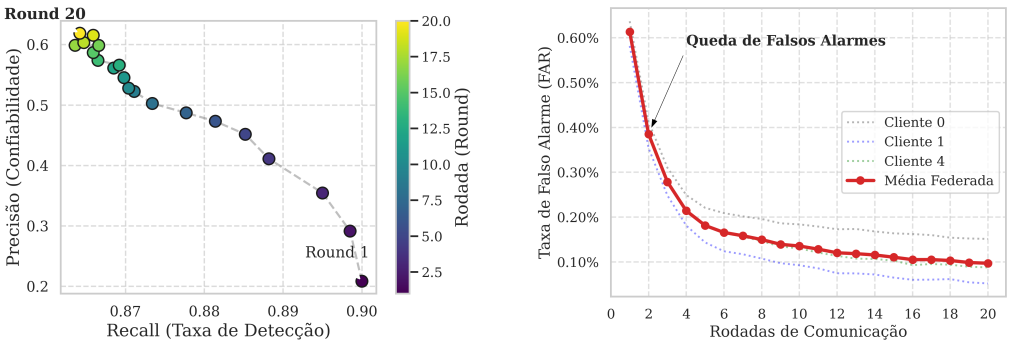


Figure 4. Desempenho por cliente federado.

Observa-se que o modelo evolui de uma região com alto *recall* e baixa precisão para uma região mais equilibrada. A redução da taxa de falsos alarmes ao longo das rodadas confirma esse comportamento, indicando que o modelo se torna progressivamente mais seletivo sem comprometer significativamente a capacidade de detecção. Esse ajuste é coerente com a evolução observada no *F1-score*, refletindo melhoria na qualidade das decisões.

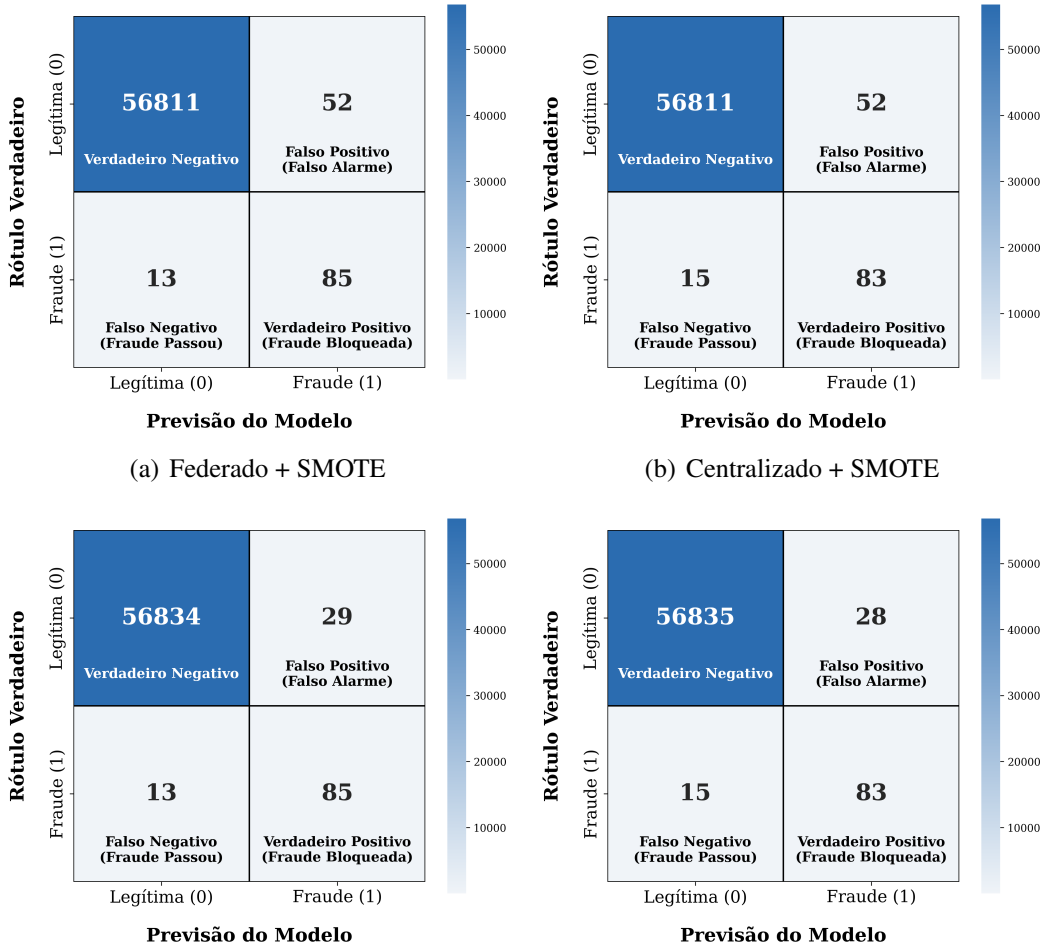


(a) Precisão vs *Recall* (b) Falsos alarmes

Figure 5. *Trade-off* entre precisão e *recall* e evolução dos falsos positivos.

A Figura 6 apresenta as matrizes de confusão. Nos cenários sem SMOTE, observa-se menor incidência de falsos positivos, consistente com os maiores valores de precisão. Nos cenários com SMOTE, há aumento significativo de falsos positivos, sem crescimento proporcional de verdadeiros positivos. Esse padrão confirma a análise quantitativa, indicando que o *oversampling* não melhora a capacidade de detecção, mas altera o limiar de decisão do modelo.

A Figura 7 apresenta o tempo total de execução. O FL apresenta menor tempo em ambos os cenários, com diferença mais acentuada no cenário com SMOTE. Esse resultado indica que o custo computacional do balanceamento sintético é amortizado pelo paralelismo entre clientes. No modelo centralizado, esse custo é concentrado, resultando



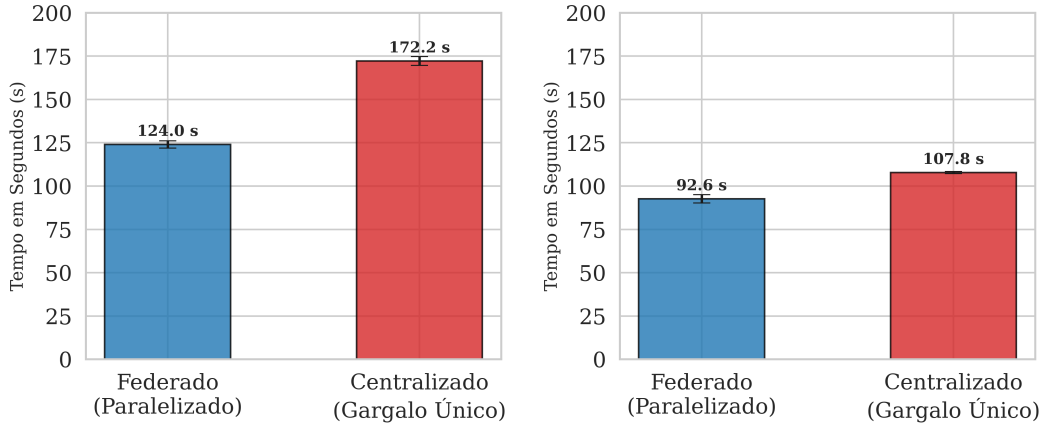
(a) Federado + SMOTE (b) Centralizado + SMOTE

(c) Federado sem SMOTE (d) Centralizado sem SMOTE

Figure 6. Matrizes de confusão para os quatro cenários.

em maior tempo total. Mesmo considerando o *overhead* de comunicação, o FL demonstra vantagem computacional, evidenciando que a distribuição do processamento compensa os custos adicionais de sincronização.

De forma consolidada, os resultados demonstram que o modelo federado alcança desempenho comparável ao treinamento centralizado, apresentando diferenças marginais em termos de *F1-score* e AUC-ROC, ao mesmo tempo em que reduz significativamente o custo de comunicação e o tempo total de execução. Observa-se ainda que o modelo federado tende a manter maior *recall*, indicando maior sensibilidade na detecção de fraudes, com um leve aumento na taxa de falsos positivos. Em relação às estratégias de balanceamento, a aplicação de SMOTE resultou em redução consistente da precisão (entre 17% e 18%) sem ganhos relevantes em *recall*, o que implica aumento expressivo de falsos positivos, conforme evidenciado pelas matrizes de confusão. Esse comportamento indica que o uso de *oversampling* sintético desloca o limiar de decisão do modelo sem melhorar sua capacidade discriminativa, apesar dos valores elevados de AUC-ROC. Em conjunto, esses resultados sugerem que abordagens baseadas em aprendizado sensível ao custo são mais adequadas do que técnicas de geração de dados sintéticos no contexto de aprendizado



(a) Com SMOTE (b) Sem SMOTE

Figure 7. Tempo total de execução.

federado para detecção de fraudes.

6. Conclusão

Este trabalho apresentou o sistema SIDERAU, uma arquitetura baseada em FL e IDS para a detecção de fraudes em transações de cartões de crédito. A proposta visou superar os gargalos de privacidade e centralização de dados sensíveis, comuns em modelos tradicionais de aprendizado de máquina. Os resultados obtidos validaram a eficácia do sistema frente ao desbalanceamento severo de classes. Diferente de abordagens genéricas, os testes quantitativos provaram que o SIDERAU alcançou um *F1-score* médio de 78,9% e uma taxa de detecção (*recall*) de 86,4%. Esses números demonstram que o modelo federado mantém um rigor preditivo comparável a modelos centralizados, mesmo sem a exposição dos dados dos clientes. No que tange ao desempenho computacional, a análise provou que a descentralização atua como um acelerador de processos. A paralelização do pré-processamento SMOTE entre os cinco nós clientes resultou em uma latência total de 124,02 segundos, o que representa uma redução de 28% no tempo total de execução comparado aos 172,20 segundos do modelo centralizado. Essa economia de tempo é acompanhada por uma otimização crítica de hardware, pois a carga de memória local por nó foi reduzida em 5 vezes, exigindo apenas 21,5 MB por máquina, contra o gargalo de 107,6 MB demandado pelo servidor central.

Em suma, conclui-se que o SIDERAU é uma solução viável e escalável para o setor financeiro. O sistema não apenas garante a conformidade com normas de privacidade, mas também oferece uma infraestrutura de detecção em tempo real mais célere e menos onerosa aos recursos locais das instituições. Como trabalhos futuros, pretende-se avaliar o desempenho do SIDERAU em cenários não-IID mais realistas, com maior número de clientes e maior heterogeneidade de dados, além de investigar mecanismos adicionais de preservação de privacidade, como agregação segura e privacidade diferencial.

References

Akour, I., Mohamed, N., and Salloum, S. (2025). Hybrid cnn-lstm with attention mechanism for robust credit card fraud detection. *IEEE Access*.

Aurna, N. F., Hossain, M. D., Taenaka, Y., and Kadobayashi, Y. (2023). Federated learning-based credit card fraud detection: Performance analysis with sampling methods and deep learning algorithms. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 180–186.

Btoush, E. A. L. M., Zhou, X., Gururajan, R., Chan, K. C., Genrich, R., and Sankaran, P. (2023). A systematic review of literature on credit card cyber fraud detection using machine and deep learning. *PeerJ Computer Science*, 9:e1278.

de Souza, A. M., Maciel, F., da Costa, J. B., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2024). Adaptive client selection with personalization for communication efficient federated learning. *Ad Hoc Networks*, 157:103462.

Fares, A. A. Y. R., Wunder, C. S., de Caldas Filho, F. L., Rocha Filho, G. P., Serrano, A. L. M., and Gonçalves, V. P. (2026). Identificação e mitigação de ataques ddos em redes iot utilizando hidps, nids e aprendizado de máquina. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 1(E78):1–12.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22.

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.-S., and Zeineddine, H. (2019). An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, 7:93010–93022.

Nugent, D. (2022). Privacy-preserving credit card fraud detection using homomorphic encryption. *arXiv preprint arXiv:2211.06675*.

Romani, M. F., dos Santos, J. K. M., de O Cardoso, G. F., Viana, G. B., de Caldas Filho, F. L., and Demendonça, F. L. (2026). Mitigação de ataques ddos em iot: Comparação entre aprendizado federado e modelos centralizados. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 1(E78):170–183.

Rzayeva, D. and Malekzadeh, S. (2022). A combination of deep neural networks and k-nearest neighbors for credit card fraud detection. *arXiv preprint arXiv:2205.15300*.

Sharma, S. and Gwale, D. (2023). A review of credit card fraud detection using machine learning. *International Journal of Scientific Development and Research (IJSDR)*, 8(7):584.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. (2019). Credit card fraud detection - machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12:1–12:19.

Zaffar, Z., Sohrab, F., Kanniainen, J., and Gabbouj, M. (2023). Credit card fraud detection with subspace learning-based one-class classification. *arXiv preprint arXiv:2309.14880*.