Aprendizado Semi-Supervisionado e Ativo para Detecção de Fraudes Contábeis

Sérgio Henrique de Carvalho Pedroso
Departamento de Automática
Universidade Federal de Lavras
Lavras, Brasil
sergio.h.cp2@gmail.com

Wilian Soares Lacerda
Departamento de Automática
Universidade Federal de Lavras
Lavras, Brasil
lacerda@ufla.br

***Resumo*—Este trabalho investiga o uso de aprendizado semi-supervisionado e aprendizado ativo na detecção de fraudes contábeis em dados extraídos do ERP de uma empresa do setor logístico. Partindo de um conjunto reduzido de registros rotulados por denúncias internas e de um grande volume de transações não rotuladas, foram comparadas as estratégias *Active Learning*, *Pseudo-Labeling* e *Label Propagation* em modelos como Regressão Logística, MLP, XGBoost e CatBoost. Os experimentos mostraram que o *Active Learning* com modelos de árvores obteve desempenho superior (F1 até 0,87 e PR-AUC 0,94), com evolução consistente ao longo das iterações, enquanto o *Label Propagation* teve resultados intermediários e o *Pseudo-Labeling* apresentou degradação devido à propagação de rótulos incorretos. Conclui-se que a combinação de *Active Learning* com modelos robustos representa uma solução eficaz para apoiar auditorias internas em cenários de escassez de rótulos, oferecendo maior precisão e escalabilidade nos processos de governança corporativa.**

***Palavras-chave*—Aprendizado Semi-Supervisionado, Auditoria, Active Learning, Detecção de Fraudes, Anomalias.**

I. Introdução

Com um enfoque na melhoria contínua, a indústria está sempre em busca de métodos mais eficientes e escaláveis para seus processos. A evolução das técnicas de inteligência computacional tem aberto novas oportunidades para a automatização de processos complexos, anteriormente executados apenas pela mente humana. Um exemplo disso é a detecção de anomalias e fraudes em pedidos de compras internas de uma empresa, que é o foco deste trabalho.

A detecção de anomalias é um campo abrangente que se dedica a identificar instâncias de dados ou eventos que não seguem o comportamento esperado [1]. O termo detecção de *outliers* é frequentemente utilizado como sinônimo de detecção de anomalias.

Devido ao grande volume de dados processados diariamente em sistemas de contabilidade, torna-se necessária a auditoria dessas operações de forma escalável e confiável. A literatura se refere aos CAATs (*Computer Assisted Audit Tools*) como sistemas que oferecem diversos benefícios na detecção de fraudes. Esses benefícios incluem a capacidade de analisar grandes volumes de dados de forma rápida e precisa, aumentando a confiança na identificação de anomalias e irregularidades. Além disso, os CAATs melhoram a eficiência do processo de auditoria, reduzindo o tempo necessário para realizar testes e verificações rotineiras, permitindo que os auditores se concentrem em análises mais complexas e estratégicas. A integração destes sistemas também contribui para a sustentabilidade corporativa, promovendo transparência e responsabilidade nas operações empresariais, além de ajudar a garantir a conformidade com regulamentos e normas legais [2].

Na era digital, as operações contábeis se tornaram cada vez mais complexas e volumosas, exigindo ferramentas avançadas para garantir a integridade e a confiabilidade das informações financeiras [3]. Fraudes e anomalias contábeis podem resultar em prejuízos significativos para as empresas e afetar a confiança de investidores e partes interessadas. Nesse contexto, métodos tradicionais de auditoria frequentemente se mostram insuficientes para lidar com a crescente quantidade de dados e a sofisticação das fraudes. A aplicação de técnicas de aprendizado de máquina surge, portanto, como uma solução promissora para identificar padrões anômalos e fraudes em grandes volumes de dados contábeis, oferecendo uma abordagem mais precisa e eficiente.

Este projeto de pesquisa é motivado pela necessidade de desenvolver e validar modelos capazes de auxiliar os auditores na detecção de irregularidades, contribuindo para a robustez dos processos contábeis e a segurança financeira das organizações.

II. Trabalhos Relacionados

O trabalho apresentado por [4] explora cinco métodos de aprendizado de máquina não supervisionados para detecção de anomalias em dados contábeis de grande escala, destacando o uso de *autoencoders* e OC-SVM, que alcançaram os melhores resultados. Os autores diferenciam ainda anomalias globais, ligadas a compras raras e discrepantes, e anomalias locais, associadas a pequenas variações em padrões recorrentes, mais difíceis de identificar.

Em [5], são investigados algoritmos *ensemble* na detecção de fraudes contábeis, avaliando também o impacto do desbalanceamento de dados. Entre as técnicas comparadas, o CusBoost obteve o melhor desempenho, reforçando o potencial de abordagens *ensemble* para capturar características intrínsecas em bases desbalanceadas.

Mais recentemente, abordagens baseadas em *Active Learning* (AL) têm ganhado destaque. Em [6], aplicou-se AL em fluxos de dados contábeis desbalanceados e não estacionários,

demonstrando que a estratégia de *High Risk Querying* melhora a adaptação a mudanças nos padrões de fraude. De forma semelhante, em contextos distintos como seguridade social [7] e demonstrações financeiras [8], verificou-se que o aprendizado ativo supera abordagens supervisionadas tradicionais, mesmo com poucos exemplos rotulados, reduzindo o custo de anotação e aumentando a precisão.

Além disso, pesquisas recentes ressaltam o potencial da combinação entre *Active Learning* e aprendizado semi-supervisionado (SSL). Essa integração permite explorar melhor os dados não rotulados e selecionar amostras mais informativas, ampliando a cobertura da detecção de fraudes sem comprometer a precisão. Aplicações em domínios como monitoramento aduaneiro e seguros médicos confirmam a eficácia dessa abordagem [9], [10].

III. Referencial Teórico

O processo de detecção de fraudes em dados contábeis consiste na identificação de anomalias de valores, frequências e tipos de compras ou serviços contratados. Dessa forma, a abordagem proposta visa adotar técnicas e estratégias de aprendizado de máquina para tratar desta questão.

Alguns conceitos importantes da literatura serão apresentados a fim de proporcionar melhor entendimento do trabalho realizado.

A. Anomalias

O conceito de anomalias pode ser definido conforme proposto por [11], em que um *outlier* é considerado uma observação que se desvia significativamente das demais, a ponto de levantar suspeitas de que tenha sido gerado por mecanismos distintos.

De forma ilustrativa, a Figura 1 apresenta um conjunto de dados representados em um espaço bidimensional, exemplificando diferentes tipos de anomalias. Os conjuntos agrupados em N1 e N2 são regiões de maior frequência de eventos, o que podemos considerar como não-anomalias. Observações individuais mais distantes como em O1 e O2, ou agrupamentos pequenos como O3, são classificados como anomalias por diferentes critérios.

Anomalias podem ser classificadas em três categorias: pontuais, contextuais e coletivas, conforme [12]. Tais técnicas são essenciais para a detecção de fraudes no contexto do trabalho proposto, além de auxiliar na construção de possíveis cenários de fraude.

Anomalias pontuais são instâncias individuais de dados que são significativamente diferentes do restante. Por exemplo, se um funcionário da empresa realiza uma compra interna de suprimentos com um valor muito maior que o usual, essa compra seria considerada uma anomalia pontual, como nos pontos O1 e O2.

Já as anomalias contextuais são instâncias de dados que são anômalas em um contexto específico. Por exemplo, se um funcionário normalmente compra peças de reposição apenas durante a semana, mas faz uma compra no fim de semana, isso pode ser considerado uma anomalia contextual, dado o contexto do período em que foi feita, independente do montante gasto.

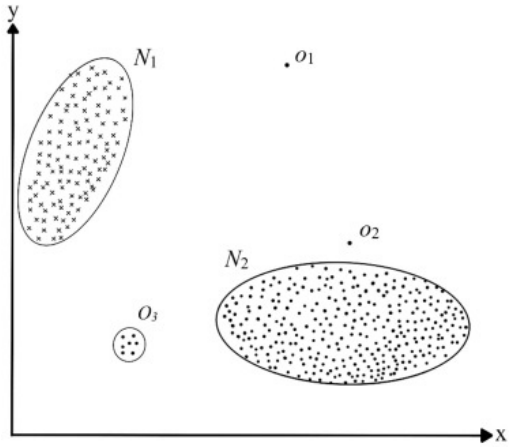


Figura 1: Representação de agrupamentos e anomalias em um espaço bidimensional [1].

Por fim, as anomalias coletivas são uma coleção de instâncias relacionadas que são anômalas em conjunto, dado todo o espaço de observações realizadas. Por exemplo, várias pequenas compras de combustível feitas por um grupo de funcionários em um curto período, que individualmente podem parecer normais, mas coletivamente indicam um padrão anômalo, como no conjunto O3. Neste caso seriam uma anomalia coletiva.

B. Active Learning

Active learning é uma abordagem de aprendizado de máquina voltada para cenários em que obter rótulos para os dados é caro ou demorado, como ocorre em tarefas de classificação de fraudes. Ao invés de rotular grandes volumes de dados aleatoriamente, o modelo seleciona iterativamente os exemplos mais informativos para serem rotulados por um especialista, reduzindo custos de anotação e acelerando o aprendizado. Essa estratégia é especialmente útil em contextos de detecção de fraudes, onde casos fraudulentos são raros e desbalanceados em relação às transações legítimas [13].

C. Aprendizado semi-supervisionado

Técnicas de aprendizado semi-supervisionado (SSL) situam-se entre o aprendizado supervisionado e o não supervisionado, combinando um pequeno conjunto rotulado $X_L = \{(x_i, y_i)\}$ e um grande conjunto não rotulado $X_U = \{x_j\}$. Na prática, SSL busca melhorar a fronteira de decisão ao explorar a estrutura do dado não rotulado. Um recente levantamento sistemático organiza esse campo em cinco famílias principais: modelos gerativos, regularização por consistência, métodos baseados em grafos, pseudo-rotulagem (*self-training*) e abordagens híbridas [14].

Quanto ao escopo de generalização, o semi-supervisionado pode ser transdutivo ou indutivo. No cenário transdutivo, os não rotulados vistos no treino são exatamente aqueles que se deseja prever, caso típico dos métodos em grafo. No

cenário indutivo, aprende-se um classificador que generaliza para novos exemplos não vistos, como ocorre em regularização por consistência e pseudo-rotulagem [15].

A regularização por consistência força o modelo a produzir respostas semelhantes quando a mesma entrada sem rótulo sofre pequenas perturbações, como ruído, *data augmentation* ou *dropout*. Entre os exemplos clássicos estão o Π-model, que minimiza a diferença entre duas predições do mesmo exemplo perturbado de maneiras diferentes, o *Temporal Ensembling*/*Mean Teacher*, que usa uma média temporal das predições (ou dos pesos) como alvo mais estável e o *Virtual Adversarial Training* (VAT), que procura a perturbação "pior" nas vizinhanças do ponto para definir o termo de consistência [14], [16]. Esses métodos não criam rótulos, mas estabilizam o classificador e empurram a fronteira para regiões de baixa densidade.

Os métodos baseados em grafos representam a geometria de $X_L \cup X_U$ por meio de um grafo de vizinhança (k-NN) e propagam rótulos ao longo das arestas, suavizando pelas conexões entre pontos próximos. Em geral, são transdutivos: o grafo já contém as instâncias que se deseja prever e, ao final, obtêm-se distribuições de rótulo para os nós não rotulados [14]. Na prática, é comum selecionar as predições mais confiantes e incorporá-las ao treino de um classificador indutivo.

A pseudo-rotulagem (*self-training*) treina um classificador inicial em X_L, pontua os exemplos de X_U e, acima (ou abaixo) de um limiar de confiança, atribui pseudo-rótulos para reincorporá-los ao treino. A abordagem é simples e muitas vezes eficaz, mas sensível a ruído, pois erros podem se propagar. Trabalhos recentes combinam propagação em grafo e critérios de seleção para melhorar a qualidade dos pseudo-rótulos [17]–[19].

Algumas hipóteses sobre a distribuição dos dados ajudam a explicar quando o semi-supervisionado traz ganhos [20]. A hipótese de *clusters* afirma que pontos no mesmo aglomerado tendem a compartilhar o mesmo rótulo. A hipótese de baixa densidade sugere que a fronteira de decisão deve passar por regiões pouco povoadas, evitando "cortar" *clusters*. A hipótese de *manifold* assume que os dados residem em uma variedade de baixa dimensão, permitindo fronteiras suaves nesse espaço. Por fim, a hipótese gerativa considera que, quando $p(x, y) = p(y)\, p(x \mid y)$ modela bem os dados, o conjunto não rotulado contribui para inferir a estrutura de classes.

D. Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) foram inicialmente desenvolvidas por [21], quando descreveram formalmente o conceito de neurônio artificial. Nesse modelo, o neurônio é representado como uma unidade computacional que recebe valores de entrada e produz uma saída. A transformação das entradas em saída ocorre por meio de uma função de ativação, conforme ilustrado na Figura 2.

O conceito de RNA é inspirado na rede neural biológica do cérebro humano, sendo constituída por um sistema de nós conectados organizados em camadas: uma camada de entrada,

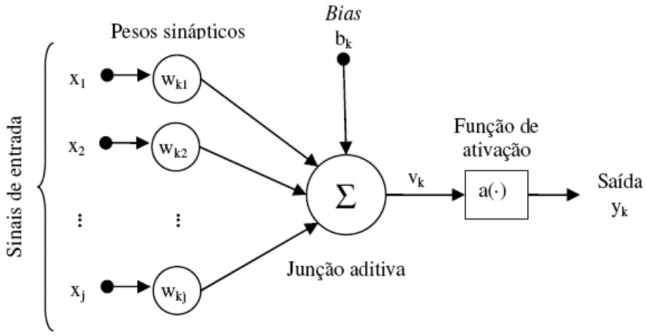


Figura 2: Representação de um neurônio artificial [22].

uma ou mais camadas intermediárias ocultas e uma camada de saída. Uma vez interligados, são formados links ponderados associados que enviam sinais entre si em forma numérica. A saída de cada neurônio é formada como uma função da soma ponderada de sua entrada. Os pesos nas conexões são modificados na fase de aprendizagem para representar a força das conexões em relação aos nós [23]. A RNA de perceptron multicamada (MLP) consiste em muitos neurônios organizados em camadas, com pelo menos uma camada oculta, conforme Figura 6b, cujos neurônios utilizam técnicas como *backpropagation* para aprendizagem supervisionada [24].

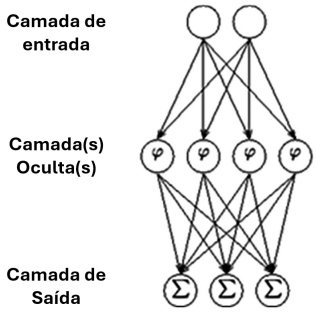


Figura 3: Representação de um Perceptron multicamadas [24].

E. XGBoost

O XGBoost (*eXtreme Gradient Boosting*) é uma implementação otimizada do algoritmo de Gradient Boosting, amplamente utilizada por sua eficiência computacional e alto desempenho preditivo em tarefas de aprendizado supervisionado. Ele incorpora regularização L1 e L2 para evitar *overfitting*, além de paralelização durante a construção das árvores, o que o torna mais rápido em comparação a implementações tradicionais de *boosting*. XGBoost também suporta processamento de dados ausentes e técnicas avançadas como *pruning*, removendo divisões que não proporcionam ganho significativo na função objetivo e *early stopping*, interrompendo o treinamento do modelo quando o desempenho em um conjunto de validação deixa de melhorar após um número definido de iterações [25].

F. Regressão Logística

A Regressão Logística é um modelo estatístico amplamente utilizado para problemas de classificação binária. Ela estima a probabilidade de um evento ocorrer com base em uma combinação linear das variáveis independentes, aplicando a função logística (sigmoide) para mapear o output entre o intervalo de 0 e 1. Sua interpretação probabilística e simplicidade matemática a tornam uma escolha comum em diversas áreas, como epidemiologia, ciências sociais e aprendizado de máquina [26].

Apesar de ser um modelo linear, sua performance pode ser bastante satisfatória em *datasets* com boa separabilidade e interpretabilidade é um de seus pontos fortes.

IV. Metodologia

Este capítulo descreve a metodologia adotada para desenvolver e avaliar um sistema de detecção de fraudes contábeis em cenário com rótulos escassos. Foram empregadas abordagens *semi-supervisionadas* como *Pseudo-Labeling* e *Label Propagation*, avaliadas com um *baseline* de *Active Learning*. Todas as estratégias operaram sob o mesmo orçamento de rotulagem por iteração e foram avaliadas em um conjunto de teste fixo.

O processo segue um ciclo iterativo. Em cada iteração: treina-se o modelo em um conjunto rotulado corrente, estima-se um limiar de decisão a partir de validação cruzada, avalia-se o desempenho em um conjunto de teste fixo e independente, selecionam-se amostras do *pool* não rotulado de acordo com a estratégia (PL, AL ou LP) e atualiza-se o conjunto rotulado para a próxima iteração. Cada modelo mantém estado próprio (seu conjunto rotulado e seu *pool*), evitando interferência entre modelos.

A. Extração e pré-processamento de dados

A organização utiliza o ERP Protheus (TOTVS)[1] com armazenamento em servidor local. Para evitar sobrecarga no ambiente transacional, foi criada uma consulta SQL que seleciona apenas os campos relevantes e materializa o resultado em um *dataflow* no Power BI[2].

A base histórica abrange lançamentos contábeis desde 2017, incluindo compras, serviços, movimentações de estoque, ativos, aluguéis e eventos, totalizando mais de 3,5 milhões de linhas e 53 colunas. Neste estudo, foi selecionado o recorte de serviços adquiridos entre janeiro/2023 e maio/2025, com cerca de 150 mil registros. Esse subconjunto foi empregado nas etapas de preparação, extração de *features* e modelagem. A classe positiva é composta por compras marcadas como fraude no canal de denúncia interno, totalizando 4,060 casos.

O pré-processamento inclui padronização de *strings* (remoção de acentos/caixa), conversão de tipos, extração temporal (ano, mês), seleção de grupos de operação e controle de qualidade (deduplicação e validação de domínios). Para modelagem, variáveis como D1_FILIAL (filial da empresa), D1_XOPER (código da operação), D1_QUANT (quantidade de produtos), VALORCTB (valor contábil do pedido), PRODUTO (nome do produto), PLACA_VEICULO (placa do veículo vinculado), MODELO_VEICULO (modelo do veículo vinculado), ANO (ano do pedido) e MES (mês do pedido) são transformadas por codificação categórica e normalização de numéricos. As Figuras 4 e 5 (escala log) ilustram a distribuição de valores monetários e quantidades.

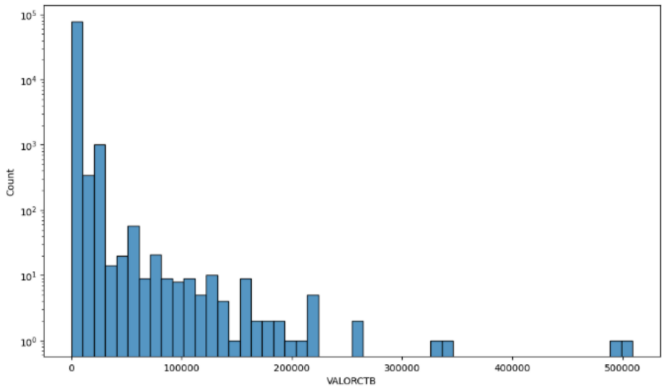


Figura 4: Distribuição logarítmica de valor contábil

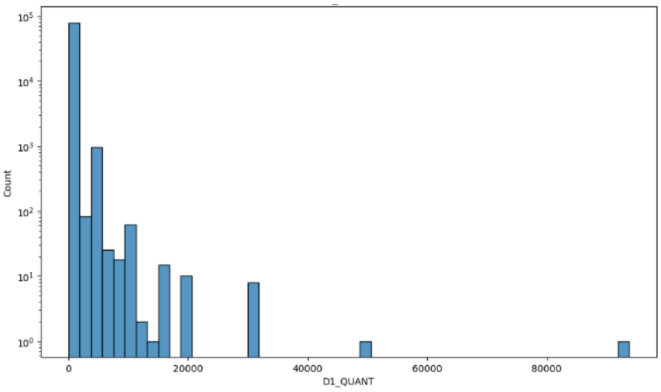


Figura 5: Distribuição logarítmica de quantidade

As distribuições de valor contábil, Figura 4, e quantidade, Figura 5, apresentam forte assimetria à direita, com muitas ocorrências em valores baixos e poucas observações muito altas. Esse perfil motivou a adoção de imputação por mediana e padronização no pré-processamento. Modelos como XGBoost e CatBoost são menos sensíveis a transformações monotônicas, lidando bem com essa heterogeneidade de escala, já para modelos lineares, a padronização é essencial para estabilizar o treinamento, bem como o uso de métricas robustas a desbalanceamento.

O conjunto total de dados foi separado em três partes: um treino inicial com rótulos confiáveis, um conjunto não rotulado e um conjunto de teste com fraudes rotuladas, mantido fixo e usado apenas para avaliação. A seleção respeitou amostragem estratificada por classe (fraude/negativo). Em cada iteração, adotou-se um orçamento fixo de exemplos para serem movidos do *pool* não rotulado para a base rotulada. Cada modelo man-

[1]TOTVS é uma empresa brasileira especializada em softwares de gestão empresarial. Mais informações em: https://www.totvs.com/

[2]Power BI é uma ferramenta de análise e visualização de dados da Microsoft. Mais informações em: https://powerbi.microsoft.com

teve estado independente, tendo sua base rotulada e seu *pool* atualizados a cada iteração sem interferência entre modelos.

B. Modelos

Foram avaliados quatro modelos: Regressão Logística, XGBoost, MLP e CatBoost. Para os modelos de Regressão Logística e MLP do Scikit-learn foi implementada a padronização de atributos numéricos e binarização para atributos categóricos por *one-hot encoding*. Para o CatBoost e XGBoost, as variáveis categóricas foram tratadas nativamente, com imputação simples e indicação explícita das colunas categóricas, sem a codificação binária. Para evitar vazamento, o pré-processador foi ajustado somente no conjunto rotulado da iteração vigente e, depois, aplicado ao *pool* não rotulado e ao conjunto de teste.

C. Estratégias utilizadas

Em relação ao *Pseudo-Labeling* (PL), a cada iteração o modelo calcula a probabilidade de fraude para todas as amostras do *pool* e selecionam-se apenas as de alta confiança, muito próximas de 1 (provável fraude) ou de 0 (provável não fraude). Dentro do orçamento de cada iteração, priorizam-se esses extremos e busca-se manter equilíbrio entre as classes. As amostras escolhidas recebem pseudo-rótulos por regra: positivo se a probabilidade $\geq 0{,}9$ e falso se $\leq 0{,}1$. Finalmente, são movidas do *pool* para o conjunto rotulado da próxima iteração, passando a integrar o treino subsequente.

No *Active Learning* (AL), adotou-se *uncertainty sampling* em lote: a cada iteração, selecionam-se no *pool* as amostras em que o modelo está mais incerto, ou seja, probabilidade prevista próxima de 50%. Em seguida, consulta-se um oráculo simulado de forma automática, que revela o rótulo oficial do conjunto de dados, como se um especialista tivesse conferido o caso. Dessa forma, os rótulos reais obtidos são incorporados ao conjunto rotulado da iteração seguinte.

No *Label Propagation* (LP), os dados foram primeiro transformados em vetores numéricos usando o mesmo pré-processamento aplicado aos demais modelos. Em seguida, aplicou-se o algoritmo Label Spreading do Scikit-learn, baseado em k-vizinhos (k = 10), kernel "knn" e com parâmetro de suavização α = 0,1, sobre o conjunto que reúne exemplos rotulados e não rotulados. O método estima, para cada exemplo do *pool*, a probabilidade de ser fraude. Com essas probabilidades, são selecionados até o limite do orçamento, os casos de maior confiança, ou seja, com probabilidades mais distantes de 50%. Esses casos recebem pseudo-rótulos conforme a regra, probabilidade $\geq$ 50% = positivo e $<$ 50% = negativo, e são inclusos no conjunto de dados de treinamento para a iteração seguinte.

D. Validação cruzada e escolha do limiar

Em cada iteração foi feita validação cruzada estratificada com cinco partições sobre o conjunto rotulado vigente. Registraram-se médias e desvios de ROC AUC, PR-AUC, F1 e *recall*. Em cada partição, ajustou-se o modelo no trecho de treino e buscou-se, no trecho de validação, o ponto de corte que maximiza o F1 na curva Precisão–Revocação. Ao final, calculou-se a média desses cinco pontos de corte e esse valor único foi aplicado às probabilidades do conjunto de teste. Assim, a escolha do limiar ficou restrita ao treino/validação, enquanto a aferição final ocorreu apenas no teste, reduzindo viés.

O ciclo foi repetido por um número fixo de iterações e, em todas as estratégias (PL, AL e LP), manteve-se o mesmo orçamento de exemplos adicionados por iteração (n=3000 registros) e estados independentes por modelo. Em cada iteração treinou-se o modelo com o conjunto rotulado atual, avaliou-se no teste usando o limiar médio obtido na validação cruzada e selecionaram-se exemplos do *pool* não rotulado para compor a próxima iteração. No caso do Aprendizado Ativo, o "oráculo" foi simulado automaticamente ao revelar o rótulo oficial disponível no conjunto de dados, emulando a conferência de um especialista humano. Os exemplos consultados foram incorporados ao treino seguinte e removidos do *pool*.

V. Resultados

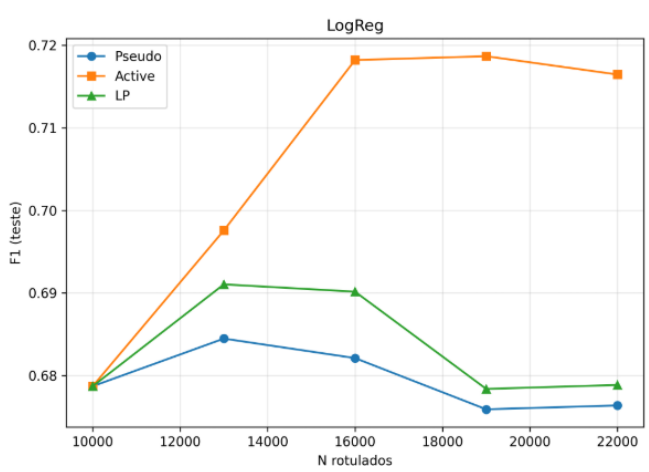
Esta seção apresenta as curvas de F1-score no conjunto de teste ao longo das iterações para cada modelo base, comparando as estratégias Pseudo-Labeling (PL), Active Learning (AL) e Label Propagation (LP) e uma tabela-resumo com as métricas finais no teste.

A Figura 6 mostra a evolução do F1-score no conjunto de teste à medida que novos exemplos são incorporados em cada estratégia. A Tabela I resume os resultados finais por modelo e estratégia, medidos no conjunto de teste na última iteração.

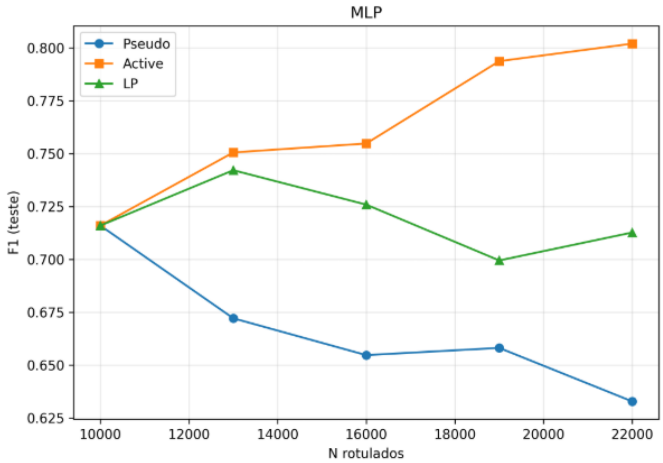
Observando a Figura 6, é perceptível uma tendência de crescimento do *Active Learning* de forma consistente em todas as arquiteturas, com ganhos mais nítidos nos modelos de árvores (CatBoost e XGBoost). Em *Pseudo-Labeling*, o F1 tende a estagnar ou cair ao longo das iterações, sinalizando propagação de ruído dos pseudo-rótulos. O *Label Propagation* se apresenta como uma estratégia intermediária entre as demais, trazendo um ganho inicial e depois estabilizando. Entre os modelos, XGBoost e CatBoost alcançam os maiores níveis de F1 sob AL, a MLP fica em faixa intermediária e a Regressão Logística, em geral, abaixo dos demais.

Pela Tabela I, apresentando as métricas na última iteração, é possível concluir que o XGBoost–AL obteve o melhor F1 (0,870) e PR-AUC (0,943), seguido de perto por CatBoost–AL (F1 0,860 e PR-AUC 0,943). O *Label Propagation* apresentou desempenho intermediário e superou o *Pseudo-Labeling* de forma consistente. Como o ROC AUC ficou praticamente saturado (aproximadamente 0,999) em quase todos os cenários, a comparação deve priorizar F1 e PR-AUC, que refletem melhor a utilidade prática na triagem de casos, considerando o objetivo de apoiar a auditoria interna em que falsos positivos bem justificados têm valor operacional.

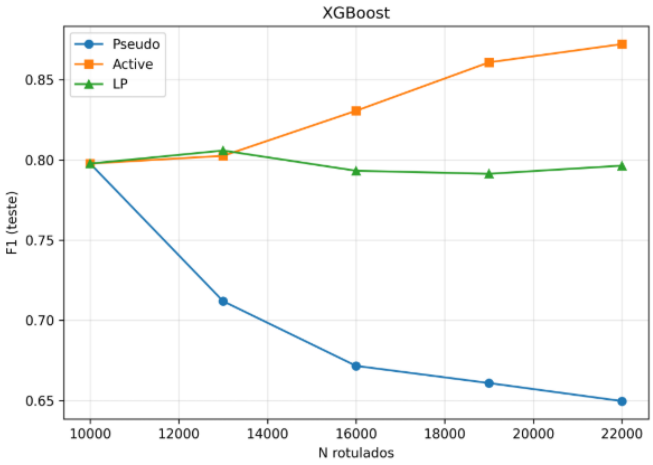
A Tabela II mostra que todas as combinações de modelo e estratégia apresentam desempenho interno muito alto na validação cruzada (ROC AUC próximo de 1,0 e PR-AUC/F1 também elevados), com desvios-padrão pequenos, indicando



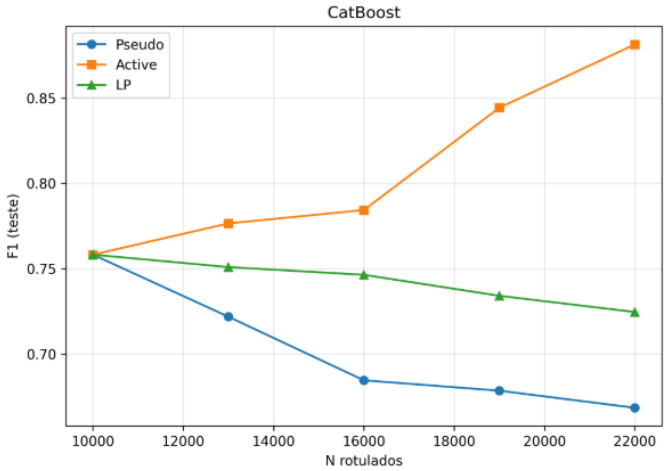
(a) Logistic Regression



(b) MLP



(c) XGBoost



(d) CatBoost

Figura 6: F1-score (teste) ao longo das iterações por modelo base, comparando PL, AL e LP.

Tabela I: Métricas no teste na **última iteração** por modelo e estratégia.

Modelo	Estratégia	F1	PR-AUC	ROC AUC
CatBoost	Active	0.860	0.943	0.999
CatBoost	LP	0.780	0.912	0.999
CatBoost	Pseudo	0.729	0.819	0.998
LogReg	Active	0.722	0.578	0.995
LogReg	LP	0.690	0.609	0.995
LogReg	Pseudo	0.686	0.594	0.995
MLP	Active	0.793	0.836	0.998
MLP	LP	0.747	0.805	0.998
MLP	Pseudo	0.704	0.723	0.997
XGBoost	Active	0.870	0.943	0.999
XGBoost	LP	0.806	0.924	0.999
XGBoost	Pseudo	0.768	0.868	0.999

estabilidade ao longo das iterações. As diferenças entre estratégias são discretas e, em alguns casos, PL/LP superam AL porque a CV mede o ajuste no conjunto rotulado atual e o limiar é escolhido por *fold*, o que tende a inflar levemente o F1. Essas métricas servem principalmente para acompanhar a aprendizagem e definir o limiar de corte.

Tabela II: Métricas de validação cruzada por modelo e estratégia (média ± desvio ao longo das iterações).

Modelo	Estratégia	Métricas de CV		
		F1 (CV)	PR-AUC (CV)	ROC AUC (CV)
CatBoost	Active	0.959 ± 0.008	0.986 ± 0.001	0.998 ± 0.000
CatBoost	LP	0.971 ± 0.001	0.989 ± 0.001	0.999 ± 0.000
CatBoost	Pseudo	0.983 ± 0.007	0.990 ± 0.002	0.998 ± 0.000
LogReg	Active	0.901 ± 0.043	0.926 ± 0.023	0.991 ± 0.003
LogReg	LP	0.946 ± 0.005	0.953 ± 0.001	0.996 ± 0.001
LogReg	Pseudo	0.978 ± 0.013	0.978 ± 0.014	0.997 ± 0.001
MLP	Active	0.920 ± 0.019	0.959 ± 0.006	0.994 ± 0.001
MLP	LP	0.948 ± 0.002	0.971 ± 0.002	0.997 ± 0.001
MLP	Pseudo	0.975 ± 0.015	0.983 ± 0.009	0.997 ± 0.001
XGBoost	Active	0.961 ± 0.006	0.987 ± 0.002	0.998 ± 0.000
XGBoost	LP	0.971 ± 0.002	0.988 ± 0.001	0.999 ± 0.000
XGBoost	Pseudo	0.981 ± 0.007	0.991 ± 0.002	0.998 ± 0.000

O modelo XGBoost com *Active Learning* alcançou o melhor desempenho global, identificando corretamente 34.855 de 35.000 transações de teste (99,6%). Foram detectadas 495 fraudes (99% de recall), com apenas 5 casos fraudulentos não

identificados. Além disso, o modelo gerou 140 falsos positivos (0,4% do total), o que é considerado aceitável na prática de auditoria, já que contribui para a filtragem e priorização de casos suspeitos a serem analisados pelos auditores.

VI. Conclusão

Este trabalho apresentou estratégias de aprendizado semi-supervisionado para auxílio na detecção de fraudes em dados contábeis. Partindo de um conjunto de dados rotulado inicial reduzido e de um grande *pool* não rotulado, sendo comparadas três estratégias: *Active Learning*, *Pseudo-Labeling* e *Label Propagation*. O protocolo incluiu validação cruzada estratificada a cada iteração para definição de limiar e um conjunto de teste fixo para aferição final, sempre com pré-processamento ajustado somente nos dados rotulados vigentes.

Os resultados indicaram que o *Active Learning* apresentou crescimento consistente de F1 ao longo das iterações e desempenho final superior, sobretudo nos modelos baseados em árvores (XGBoost e CatBoost). O *Label Propagation* obteve desempenho intermediário, com ganhos iniciais e posterior estabilização. Já o *Pseudo-Labeling*, como estratégia isolada, tendeu a degradar ou estagnar o F1 em iterações posteriores, sinalizando propagação de ruído de pseudo-rótulos. Em termos práticos para a auditoria, a combinação de AL e modelos de árvores ofereceu a melhor relação entre cobertura e precisão, com PR-AUC e F1 elevados no teste. Como o objetivo é priorizar casos para investigação, a presença controlada de falsos positivos é aceitável e até desejável quando acompanhada de alta precisão no topo do ranking.

Entretanto, vale mencionar que um ponto favorável de *Label Propagation* e *Pseudo-Labeling* é que não exigem validação humana durante o ciclo de treinamento, o sistema pode evoluir e ampliar o conjunto rotulado sem consultar um especialista a cada iteração. Mesmo com desempenho inferior ao *Active Learning*, essas abordagens reduzem custo operacional, funcionam bem em cenários de pouca disponibilidade do "oráculo" e podem servir como etapa inicial de *bootstrapping* antes de ativar AL para refinar os casos mais ambíguos.

Algumas limitações devem ser registradas, como a validação do AL ser feita de forma simulada revelando o rótulo oficial, não substituindo a validação humana em fluxo real. Além disso, a eficácia de PL/LP depende de hipóteses de estrutura nos dados (aglomeração e baixa densidade) que podem não se sustentar uniformemente.

Como trabalhos futuros, pretende-se integrar o *loop* de AL ao processo de auditoria, avaliando sua efetividade de forma qualitativa, bem como testar variantes de AL com diversidade no lote e alocação ciente de classe, explorar PL/LP como apoio e testes com diferentes abordagens de engenharia de *features*. Com essas evoluções, o sistema tende a ampliar o valor para a auditoria interna, filtrando com mais assertividade os casos que merecem investigação.

Agradecimentos

Este trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por meio do aporte financeiro à esta pesquisa. Agradecemos também ao apoio financeiro da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) para a participação no congresso.

Referências

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, 07 2009.

[2] A. Samagaio and T. A. Diogo, "Effect of computer assisted audit tools on corporate sustainability," *Sustainability*, vol. 14, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/2/705

[3] N. Khorunzhak and I. Lukanovska, "Accounting in the digital economy: Problems and prospects," *Black Sea Economic Studies*, 2019.

[4] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," *CoRR*, vol. abs/1709.05254, 2017. [Online]. Available: http://arxiv.org/abs/1709.05254

[5] M. J. Rahman and H. Zhu, "Detecting accounting fraud in family firms: Evidence from machine learning approaches," *Advances in Accounting*, vol. 64, p. 100722, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0882611023000810

[6] F. Carcillo, Y.-A. L. Borgne, O. Caelen, and G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization," *International Journal of Data Science and Analytics*, vol. 5, no. 4, pp. 285–300, 2018.

[7] V. V. Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "AFRAID: Fraud detection via active inference in time-evolving social networks," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2015, pp. 659–666.

[8] S. Karlos, G. Kostopoulos, S. Kotsiantis, and V. Tampakas, "Using active learning methods for predicting fraudulent financial statements," in *Artificial Intelligence Applications and Innovations*. Springer, 2017, pp. 351–362.

[9] S. Kim, T. Mai, T. Khanh, S. Han, S. Park, K. Singh, and M. Cha, "Take a chance: Managing the exploitation-exploration dilemma in customs fraud detection via online active learning," *arXiv preprint arXiv:2010.14282*, 2020. [Online]. Available: https://arxiv.org/abs/2010.14282

[10] J. Zhang, "Research on medical insurance fraud identification method based on multi-source datasets," *Journal of Electronics and Information Science*, vol. 9, no. 3, pp. 53–61, 2024.

[11] D. Hawkins, *Identification of Outliers*, ser. Monographs on applied probability and statistics. Chapman and Hall, 1980. [Online]. Available: https://books.google.com.br/books?id=fb0OAAAAQAAJ

[12] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: A review of anomaly detection techniques and recent advances," *Expert Systems with Applications*, vol. 193, p. 116429, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421017164

[13] F. Carcillo, Y. Borgne, O. Caelen, and G. Bontempi, "An assessment of streaming active learning strategies for real-life credit card fraud detection," *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 631–639, 2017.

[14] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[15] H. Cevikalp, M. Elmas, and S. Ozkan, "Large-scale image retrieval using transductive support vector machines," *Computer Vision and Image Understanding*, vol. 173, pp. 2–12, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314217301364

[16] R. Fu, C. Chen, S. Yan, X. Wang, and H. Chen, "Consistency-based semi-supervised learning for oriented object detection," *Knowledge-Based Systems*, vol. 304, p. 112534, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705124011687

[17] Z. Zhao, L. Zhou, L. Wang, Y. Shi, and Y. Gao, "Lassl: Label-guided self-training for semi-supervised learning," *The Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 9208–9216, 2022.

[18] M. I. J. Guo, Z. Liu, S. M. I. T. Zhang, and F. I. C. L. P. Chen, "Incremental self-training for semi-supervised learning," *ArXiv*, vol. abs/2404.12398, 2024.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot,

and E. Duchesnay, *scikit-learn: Machine Learning in Python - Semi-supervised learning*, 2011, accessed: 2025-03-27. [Online]. Available: https://scikit-learn.org/stable/modules/semi_supervised.html

[20] H. Ye, P. Wu, Y. Huo, X. Wang, Y. He, X. Zhang, and J. Gao, "Bearing fault diagnosis based on randomized fisher discriminant analysis," *Sensors*, vol. 22, no. 21, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8093

[21] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[22] B. Frascaroli, "Utilização de redes neurais artificiais para classificação de ratings de risco soberano," 01 2006.

[23] C. M. Bishop, *Neural networks for pattern recognition*. USA: Oxford university press, 1995.

[24] M. Desai and M. Shah, "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn)," *Clinical eHealth*, vol. 4, pp. 1–11, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2588914120300125

[25] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[26] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ: Wiley, 2013.