

**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE SISTEMAS DE INFORMAÇÃO**

DANIELE CHAVES DOS SANTOS

**DETECÇÃO DE FRAUDES EM TRANSAÇÕES PIX COM
ALGORITMOS DE APRENDIZADO DE MÁQUINA**

Itacoatiara - Amazonas
Julho - 2025

DANIELE CHAVES DOS SANTOS

**DETECÇÃO DE FRAUDES EM TRANSAÇÕES PIX COM
ALGORITMOS DE APRENDIZADO DE MÁQUINA**

Monografia apresentada ao Instituto de Ciências Exatas e Tecnologia da Universidade Federal do Amazonas como parte dos requisitos necessários para a obtenção do título de Bacharel em Sistemas de Informação.

**ORIENTADOR: PROF. DR. ANDREY ANTONIO DE OLIVEIRA
RODRIGUES**

Itacoatiara - Amazonas
Julho - 2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S237d Santos, Daniele Chaves dos

Detecção de fraude em transações Pix com algoritmos de
aprendizado de máquina / Daniele Chaves dos Santos. - 2025.
34 f. : il., p&b. ; 31 cm.

Orientador(a): Andrey Antonio de Oliveira Rodrigues.

Trabalho de Conclusão de Curso (graduação) - Universidade
Federal do Amazonas, Instituto de Ciências Exatas e Tecnologia de
Itacoatiara, Curso de Sistemas de Informação, Itacoatiara, 2025.

1. Machine learning. 2. Fraud detection. 3. Instant payment. 4. Pix
transaction. I. Rodrigues, Andrey Antonio de Oliveira. II.
Universidade Federal do Amazonas. Instituto de Ciências Exatas e
Tecnologia de Itacoatiara. Curso de Sistemas de Informação. III.
Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Curso de Sistemas de Informação - ICET

FOLHA DE APROVAÇÃO

DANIELE CHAVES DOS SANTOS

DETECÇÃO DE FRAUDES EM TRANSAÇÕES PIX COM ALGORITMOS DE APRENDIZADO DE MÁQUINA

Monografia apresentada ao Instituto de Ciências Exatas e Tecnologia da Universidade Federal do Amazonas como parte dos requisitos necessários para a obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em 07 de julho de 2025

BANCA EXAMINADORA

Prof. Dr. Andrey Antonio de Oliveira Rodrigues
Universidade Federal do Amazonas

Prof. Dr. Carlos Alberto Oliveira de Freitas
Universidade Federal do Amazonas

Prof. Dr. Fernando Júnior Soares dos Santos
Universidade Federal do Amazonas

Folha de Aprovação assinada pela Profa. Dra. Odette Mestrinho Passos, responsável pela disciplina ITS903 - Trabalho Final de Graduação do Curso de Sistemas de Informação (Período: 2025.1), onde atesta a defesa da aluna e a presença dos membros da banca examinadora.



Documento assinado eletronicamente por **Odette Mestrinho Passos, Professor do Magistério Superior**, em 11/07/2025, às 15:58, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Carlos Alberto Oliveira de Freitas, Professor do Magistério Superior**, em 11/07/2025, às 16:20, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Fernando Júnior Soares dos Santos, Professor do Magistério Superior**, em 11/07/2025, às 17:44, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Andrey Antonio de Oliveira Rodrigues, Professor do Magistério Superior**, em 11/07/2025, às 19:06, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2679203** e o código CRC **07AF7C96**.

Rua Nossa Senhora do Rosário - Bairro Tiradentes nº 3836 - Telefone: (92) (92) 99318-2549
CEP 69103-128 Itacoatiara/AM - ccsiicet@ufam.edu.br

Referência: Processo nº 23105.029779/2025-39

SEI nº 2679203

Se eu tiver que contar a minha história sem mencionar Jesus, não terei nada de bom a dizer - porque tudo o que há de bom em mim vem dEle.

Anna Souttoo

AGRADECIMENTOS

Agradeço primeiramente a Deus, aquele que desde meu nascimento escreveu a minha história e trilhou a minha trajetória. Sou imensamente grata a Jesus por ter me salvo e por ter me proporcionado a oportunidade de está aqui hoje agradecendo e realizando o meu sonho de me graduar.

Agradeço a minha família em geral, mas em especial a minha mãe Leonor Ferro Chaves que nunca mediu esforços para os meus estudos, é a pessoa que sempre acreditou em mim e nos meus sonhos e foi aquela que segurou nas minhas mãos e acreditou que eu venceria de cabeça erguida apesar de todos os problemas e dificuldades que passei, *obrigada Mãe!*

Ao meu orientador, professor Andrey Rodrigues, deixo aqui a minha mais profunda gratidão por você ter aceito estar comigo, ter acreditado no meu potencial, pela paciência durante todo o desenvolvimento deste trabalho, saiba que tudo que fez por mim e que todo seu conhecimento foram absorvidos por mim e aqui deixo a minha admiração e agradecimentos.

Aos professores e colegas do Instituto de Ciências Exatas e Tecnologia (ICET), obrigado por cada troca de conhecimento, por cada conversa que me fortaleceu e por fazerem parte desta etapa da minha formação.

Aos amigos que caminharam comigo nesta jornada e àqueles que me acolheram pelo caminho, minha gratidão por cada gesto de apoio e presença.

Meus agradecimentos expressos as ferramentas de Inteligência Artificial (IA), como ChatGPT e Gemini, que foram essenciais para que este trabalho fosse enriquecido e tivesse suas qualidades elevadas, sou grata por essas ferramentas existirem pois são de cunho essencial para elaborar ideias abstratas em realidade.

Por fim, agradeço a todos aqueles que estiveram comigo nesta caminhada e que tudo que for bom e leve chegue até mim, este trabalho é dedicado a Deus, a minha Mãe e a Mim.

Detecção de Fraudes em Transações Pix com Algoritmos de Aprendizado de Máquina

Daniele Chaves Dos Santos¹, Andrey Rodrigues¹

¹Instituto de Ciências Exatas e Tecnologia em Itacoatiara – (UFAM)
Caixa Postal 69103-128 – Itacoatiara – AM – Brazil

daniele.santos@ufam.edu.br, andrey.rodrigues@ufam.edu.br

Resumo. A popularização do Pix, sistema de pagamento instantâneo criado pelo Banco Central do Brasil, trouxe avanços significativos na agilidade das transações financeiras. No entanto, também intensificou a necessidade de mecanismos eficientes para detecção de fraudes em tempo real. Este trabalho tem como objetivo avaliar a eficácia de algoritmos de aprendizado de máquina aplicados à identificação de transações fraudulentas. Foram utilizados modelos de Regressão Logística, Árvore de Decisão, Random Forest e XGBoost. Os modelos foram treinados em uma base sintética com mais de seis milhões de registros, simulando operações reais. A análise considerou métricas como Acurácia, Precisão, Recall e F1-Score. O XGBoost apresentou o melhor desempenho, alcançando acurácia de 99,97%. Conclui-se que modelos de ensemble learning são promissores para fortalecer a segurança em sistemas de pagamento instantâneo como o Pix.

1. Introdução

Com a transformação digital nos últimos anos, o setor financeiro passou por mudanças significativas na forma de realizar operações bancárias. A criação do meio de pagamento digital Pix, lançado pelo [Banco Central do Brasil 2020], representou um marco importante para os pagamentos no país. O sistema permite transferências instantâneas, com disponibilidade ininterrupta e sem a necessidade de intermediários, facilitando transações entre pessoas físicas, jurídicas e entes governamentais.

Apesar dos inúmeros benefícios, o crescimento do Pix trouxe também novos desafios, especialmente relacionados à segurança das transações. A velocidade com que as operações são realizadas pode dificultar a detecção de ações fraudulentas, exigindo respostas rápidas e eficazes. Nesse contexto, técnicas de aprendizado de máquina (*machine learning* - ML), amplamente estudadas para a detecção de fraudes em meios de pagamento, apresentam ferramentas promissoras para identificar padrões anômalos e proteger os usuários desse sistema [Souza and Bordin 2023].

Diante do aumento expressivo das fraudes em meios digitais e do impacto econômico e social causado por essas ações, torna-se necessário o desenvolvimento de mecanismos eficientes para detectar comportamentos suspeitos em transações financeiras. Segundo dados do [Banco Central do Brasil 2023b], o número de notificações de fraudes no Pix passou de uma média mensal de 30.892 em 2021 para 136.882 em 2022 e chegou a 216.046 em 2023, indicando um aumento expressivo na tentativa de golpes e transações não autorizadas. A fraude está relacionada ao uso indevido de dados e informações de um

titular de conta, possibilitando a realização de compras e transferências não autorizadas. A motivação deste trabalho é aplicar técnicas de ML, consideradas promissoras na análise de grandes volumes de dados, identificação de padrões anômalos e prevenção de fraudes, contribuindo para a segurança dos usuários e para a manutenção da credibilidade do sistema financeiro [Martins and Galeale 2022].

Apesar da eficiência do Pix como sistema de pagamento instantâneo, o aumento de transações fraudulentas, especialmente em horários noturnos, quando há menor supervisão, expõe limitações nos métodos atuais de detecção, que frequentemente dependem de regras manuais e se mostram ineficazes para identificar novos padrões de golpe, geram altas taxas de falsos positivos e falham em analisar adequadamente operações realizadas nesse período crítico. Diante desse cenário, este trabalho tem como objetivo geral treinar algoritmos de aprendizado de máquina supervisionado para detectar fraudes no Pix com maior precisão, utilizando métricas de desempenho. Os objetivos específicos são: (1) avaliar o desempenho dos algoritmos *Random Forest*, *XGBoost* e Regressão Logística na identificação de transações fraudulentas; (2) analisar o comportamento das fraudes em transações realizadas no período noturno; (3) comparar os modelos com base em métricas como acurácia, precisão, AUC, *recall* e *F1-score*; e (4) propor uma abordagem mais robusta e adaptável para a detecção de fraudes no Pix em cenários de baixa supervisão.

Como principal resultado, destaca-se que o modelo *XGBoost* que apresentou o melhor desempenho geral obtendo 99.97%, alcançando um valor elevado de acurácia e *recall*, indicando maior capacidade de identificar transações fraudulentas mesmo em condições desafiadoras.

Portanto, este estudo foi dividido nas seguintes seções: Seção 2 — Fundamentação Teórica, que apresenta os conceitos necessários ao entendimento do tema; Seção 3 — Estado da Arte, que discute trabalhos relacionados; Seção 4 — Metodologia, que detalha as etapas da pesquisa; Seção 5 — Resultados e Discussões, onde são apresentados e analisados os achados; e Seção 6 — Conclusão, que resume as contribuições e propõe perspectivas futuras.

2. Fundamentação Teórica

Esta seção tem como objetivo apresentar os principais conceitos que embasam teoricamente o desenvolvimento deste trabalho, oferecendo uma compreensão sólida dos temas abordados. Inicialmente, será explorado o Sistema de Pagamentos Instantâneos Pix, sua estrutura, funcionamento e crescente adoção no Brasil. Em seguida, discute-se o aumento dos casos de fraudes envolvendo transações via Pix, destacando seus impactos e desafios para instituições financeiras e usuários.

Na sequência, aborda-se o campo do Aprendizado de Máquina (*Machine Learning*), com foco especial nos métodos utilizados para detectar padrões e anomalias em grandes volumes de dados financeiros. São descritas as abordagens de aprendizado supervisionado e não supervisionado, com ênfase nas tarefas de classificação e regressão, que são fundamentais para a construção dos modelos preditivos utilizados neste estudo.

Ao reunir esses elementos, o arcabouço teórico aqui apresentado fornece a base necessária para compreender os métodos empregados, justificar as escolhas técnicas adotadas e sustentar as análises realizadas no processo de detecção de fraudes em transações Pix utilizando algoritmos de aprendizado de máquina.

2.1. O Sistema de Pagamentos Instantâneos Pix

O Pix é um sistema de pagamentos instantâneos online desenvolvido pelo Banco Central do Brasil (BCB), lançado em novembro de 2020 com o objetivo de promover a inclusão financeira e modernizar o sistema de pagamentos nacional. Apesar dos benefícios em termos de velocidade, conveniência e redução de custos, o Pix também se tornou um vetor para novos tipos de crimes financeiros, como fraudes digitais, sequestros-relâmpago e golpes de engenharia social. A natureza irreversível e instantânea das transações dificulta o estorno de valores, tornando o sistema atraente para fraudadores. Em resposta, o Banco Central e as instituições financeiras implementaram medidas de segurança como limites de transações noturnas, validação de dispositivos e monitoramento de comportamentos suspeitos. No entanto, esses mecanismos ainda são insuficientes para lidar com a complexidade e volume de transações diárias, conforme indicado pela Federação Brasileira de Bancos (FEBRABAN), que destaca o crescimento contínuo de tentativas de golpes mesmo após a adoção de medidas preventivas [FEBRABAN 2023].

2.2. Funcionamento do Pix

O sistema utiliza uma infraestrutura centralizada, denominada Sistema de Pagamentos Instantâneos (SPI), operada pelo BCB, que garante a interoperabilidade entre diferentes instituições financeiras [Banco Central do Brasil 2024b]. A identificação dos usuários pode ser feita por meio de chaves Pix que podem ser o CPF, CNPJ, e-mail, número de telefone ou uma chave aleatória, simplificando o processo de pagamento. Desde seu lançamento, o Pix obteve ampla adesão no Brasil, com mais de 194 milhões de chaves registradas e mais de 42 bilhões de transações realizadas apenas em 2023, movimentando aproximadamente 17,2 trilhões de reais, segundo dados oficiais do [Banco Central do Brasil 2024a].

2.3. Fraudes em Pagamentos Pix

Desde que foi lançado, o Pix se tornou uma das formas mais populares de transferência de dinheiro no Brasil, justamente pela sua rapidez e praticidade. No entanto, essas mesmas características também chamaram a atenção de criminosos. Golpes como o *phishing*, onde o fraudador engana a vítima para obter informações pessoais, e o *spoofing*, em que se passa por outra pessoa ou empresa, tornaram-se comuns. Além disso, há casos de sequestro-relâmpago, em que as vítimas são forçadas a fazer transferências instantâneas por meio do Pix. Como o sistema não permite o cancelamento da transação após a conclusão, muitas vezes o prejuízo é imediato e irreversível [FEBRABAN 2023].

Para lidar com esse problema, o Banco Central e os bancos criaram medidas de proteção, como limites de valor para transferências feitas à noite, verificação de dispositivos e monitoramento automático de comportamentos suspeitos. Também foi implementado o Mecanismo Especial de Devolução (MED), que permite tentar recuperar o dinheiro em alguns casos de fraude, desde que a solicitação atenda a certas regras [Banco Central do Brasil 2023a]. Mesmo com essas ações, a [FEBRABAN 2023] aponta que o número de tentativas de golpe continua crescendo, o que mostra que os criminosos estão sempre buscando novas formas de burlar o sistema. Por isso, além das proteções técnicas, é essencial que os usuários estejam atentos e bem informados para evitar cair em fraudes.

2.4. Aprendizado de Máquina

O aprendizado de máquina é um dos campos da inteligência artificial que busca fazer o reconhecimento de padrões através de algoritmos, visando a classificação de determinados comportamentos com base em grandes quantidades de conjuntos de dados segundo [SILVA et al. 2022]. De acordo com [Wang et al. 2021] ele afirma que o uso de técnicas de aprendizado de máquina tem se mostrado mais relevante no combate a fraudes financeiras, especialmente em ambientes digitais. De acordo com seu estudo, ele cita diversos algoritmos de aprendizado de máquina por reforço para detectar fraudes em transações de fraudes na internet, são citados algoritmos como *K-nearest neighbors* (KNN), Bayesiano, árvores de decisão, floresta aleatória (*random forest*), máquina de vetores de suporte (SVM) e redes neurais.

O aprendizado de máquina pode ser dividido, de forma geral, em duas categorias principais: aprendizado supervisionado e não supervisionado. No caso do aprendizado supervisionado, o modelo é treinado com base em exemplos previamente rotulados, ou seja, o programador informa ao algoritmo quais padrões estão associados a determinados comportamentos. Dessa forma, o sistema aprende a identificar e classificar novos dados com base nesse conhecimento previamente fornecido [SILVA et al. 2022]. O aprendizado não supervisionado é uma abordagem do *machine learning* que busca identificar padrões ou estruturas ocultas em conjuntos de dados sem a necessidade de rótulos pré-definidos. O uso de aprendizado não supervisionado permite identificar anomalias, ou seja, transações que se desviam do comportamento considerado comum mesmo quando não há exemplos explícitos de fraude no histórico de dados [Delgolla et al. 2021].

2.4.1. Aprendizado Supervisionado

O aprendizado supervisionado é bastante usado em estudos sobre detecção de fraudes porque consegue classificar dados com base em exemplos já conhecidos. Segundo [Delgolla et al. 2021], esse tipo de aprendizado permite que os modelos reconheçam padrões a partir de um histórico de transações que já foram identificadas como legítimas ou fraudulentas, o que facilita a identificação de novos casos semelhantes. Um dos algoritmos mais utilizados nesse contexto é o Random Forest, que se destaca por funcionar bem quando há muitas variáveis envolvidas, sendo eficaz na detecção de padrões mais complexos nos dados.

Ainda de acordo com [Delgolla et al. 2021], outros algoritmos supervisionados também têm mostrado bons resultados. A Máquina de Vetores de Suporte (SVM), por exemplo, é bastante eficaz em problemas de classificação que envolvem duas categorias, como é o caso da separação entre transações normais e fraudulentas. Outro destaque é o XGBoost, modelo muito utilizado tanto em pesquisas quanto em aplicações práticas. Ele tem como principal vantagem a capacidade de lidar bem com conjuntos de dados desbalanceados, o que é comum em casos de fraude, ajudando a evitar que o modelo se ajuste demais aos dados de treino e, assim, melhorando sua performance geral.

Embora o aprendizado supervisionado seja muito utilizado para prever fraudes com base em transações históricas rotuladas, esse processo pode ser limitado pela necessidade de tempo e esforço para classificar os dados manualmente [Chang et al. 2022]. Como nem sempre os rótulos estão disponíveis de forma imediata, surgem alternati-

vas, como o uso do aprendizado não supervisionado. Essa abordagem busca identificar padrões normais de comportamento nas transações e detectar anomalias, ou seja, comportamentos fora do padrão como possíveis fraudes.

2.4.2. Aprendizado Não Supervisionado

O aprendizado não supervisionado, segundo [Souza 2024] é uma abordagem de aprendizado de máquina que busca encontrar estruturas ou padrões nos dados, sem depender de rótulos ou respostas previamente definidas. Em vez de tentar prever um resultado específico, o objetivo é criar uma representação que resuma os dados de forma compacta e coerente. Dado um conjunto de dados, o modelo tenta entender como esses dados estão distribuídos e identificar regularidades diretamente a partir deles. Sob a perspectiva probabilística, isso significa modelar a distribuição, ou seja, entender como os dados se comportam no conjunto como um todo.

O aprendizado não supervisionado tem um papel importante na fase de exploração e preparação dos dados, ajudando principalmente a escolher e criar os atributos mais relevantes para a análise. De acordo com [Delgolla et al. 2021], técnicas como o agrupamento de dados (*clustering*) e a redução de dimensionalidade são utilizadas para entender os comportamentos mais comuns dos usuários e identificar padrões que fogem do esperado, o que pode indicar possíveis fraudes.

Apesar dessas contribuições, o aprendizado não supervisionado geralmente não é suficiente para classificar com precisão uma transação como fraudulenta ou legítima. Para isso, técnicas de aprendizado supervisionado, especialmente os algoritmos de classificação, tornam-se fundamentais ao permitirem que os modelos aprendam a partir de exemplos já conhecidos [Souza 2024].

2.5. Classificação

A classificação segundo [Mattos 2022] está presente em diversas situações do nosso dia a dia, sempre que precisamos organizar ou agrupar objetos, ideias ou informações em categorias distintas. No contexto de aprendizado de máquina, classificar significa ensinar o modelo a identificar a qual grupo um determinado dado pertence, sendo que cada elemento só pode pertencer a uma única classe. Entre os algoritmos mais conhecidos para esse tipo de tarefa estão a Árvore de Decisão e a Regressão Logística, além de métodos de ensemble mais robustos como o Floresta Aleatória e o XGBoost, que foram aplicados neste estudo.

A classificação é uma das tarefas centrais do aprendizado de máquina supervisionado e tem como objetivo ensinar um modelo a reconhecer a qual grupo uma determinada entrada pertence, com base em exemplos previamente rotulados. No contexto da detecção de fraudes, por exemplo, o modelo é treinado com dados históricos de transações classificadas como legítimas ou fraudulentas, o que o capacita a identificar comportamentos suspeitos em novos casos. Essa abordagem é essencial para automatizar a análise de grandes volumes de dados e oferecer respostas rápidas e precisas, auxiliando na prevenção de perdas financeiras em sistemas de pagamento eletrônico [Tudisco et al. 2024].

Por outro lado, quando o objetivo não é identificar categorias, mas prever valores contínuos, utiliza-se a tarefa de regressão. Enquanto a classificação lida com decisões

discretas, a regressão busca estimar relações entre variáveis, sendo amplamente aplicada em contextos como previsão de preços, análise de tendências e avaliação de riscos. Ambas as abordagens, classificação e regressão, são fundamentais no aprendizado supervisionado e podem ser combinadas ou utilizadas de forma complementar, dependendo da natureza do problema a ser resolvido [SILVA et al. 2022].

2.6. Regressão

A regressão de acordo com [Kant 2024] é uma técnica estatística amplamente utilizada no aprendizado supervisionado para prever valores contínuos ou probabilidades. No contexto da detecção de fraudes, destaca-se a regressão logística, por sua capacidade de lidar com problemas de classificação binária, como diferenciar entre transações legítimas e fraudulentas. Por ser um modelo clássico e de fácil interpretação, ela é frequentemente utilizada como um ponto de partida *baseline* para comparação com algoritmos mais complexos, papel que desempenhará no presente estudo. Essa abordagem modela a probabilidade de uma transação ser fraudulenta com base em características históricas e comportamentais extraídas dos dados.

No artigo de [Al-dahasi et al. 2025a], a técnica foi aplicada como parte de um conjunto de algoritmos de machine learning para classificar transações como legítimas ou fraudulentas, utilizando um conjunto de dados reais e altamente desbalanceado. Mesmo diante desse desafio, o modelo de regressão logística apresentou desempenho satisfatório, com 93% de acurácia, 91% de precisão e 95% de recall, destacando-se pelo equilíbrio entre simplicidade, interpretabilidade e eficiência computacional.

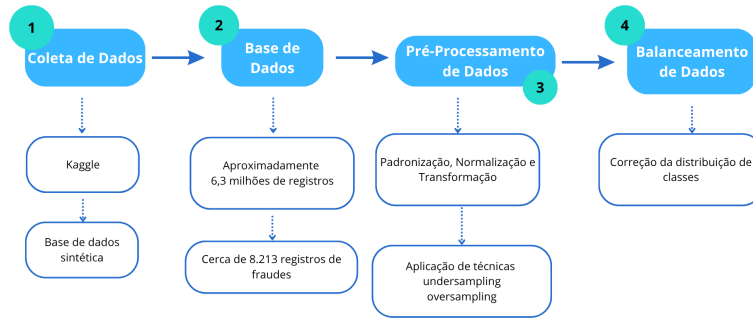
Diante dos bons resultados obtidos, vale a pena entender como a Regressão Logística e outras técnicas semelhantes têm sido utilizadas por pesquisadores nos últimos anos, principalmente em estudos que buscam detectar fraudes financeiras em tempo real, como é o caso do sistema de pagamentos instantâneos no Brasil. A seguir, apresenta-se o Estado da Arte, apresentando os trabalhos relacionados e seus níveis de pesquisa alcançados ao longo do desenvolvimento deste presente trabalho.

3. Metodologia

A metodologia adotada para o levantamento teórico deste trabalho foi a *Revisão Rápida da Literatura* (RRL), que consiste em um método estruturado, porém com simplificações em relação à Revisão Sistemática, possibilitando a obtenção de evidências científicas de forma ágil, mantendo rigor metodológico. A RRL busca responder perguntas específicas em um curto espaço de tempo, oferecendo uma síntese objetiva sobre o estado atual do conhecimento em determinada área [Tricco et al. 2015].

O fluxograma a seguir apresentado na Figura 1 ilustra as etapas iniciais da metodologia aplicada neste estudo, desde a coleta até o balanceamento dos dados, que serão detalhadas nas seções subsequentes.

Diferente da revisão narrativa, que se caracteriza pela descrição geral de um tema, muitas vezes sem critérios claros de seleção, a RRL segue um protocolo que inclui definição de objetivos, critérios de inclusão e exclusão, além da seleção de bases de dados relevantes. Segundo [Haby et al. 2016], a aplicação da RRL permite reduzir o tempo e os custos sem comprometer a qualidade da síntese, sendo especialmente útil em contex-



30

Figure 1. Etapas de Coleta e Preparação dos Dados

Fonte: Elaborado pela autora.

tos que demandam tomadas de decisão mais ágeis, como o desenvolvimento de soluções tecnológicas.

Dessa forma, a utilização da RRL neste trabalho permitiu identificar, de maneira direcionada, as principais publicações relacionadas à detecção de fraudes em transações financeiras utilizando algoritmos de aprendizado de máquina, possibilitando compreender as abordagens adotadas, os algoritmos mais utilizados e os desafios enfrentados na área.

3.1. Estado da Arte da Área Pesquisada

O processo de pesquisa e seleção dos trabalhos relacionados foi desenvolvido com base em uma Revisão Rápida da Literatura, a opção pela RRL para fundamentar o estudo do estado da arte da área pesquisada se justifica pela necessidade de obter um panorama conciso e direcionado das principais publicações relevantes para o tema de detecção de fraudes financeiras em pagamentos instantâneos no Brasil [Cartaxo et al. 2018], em suma a escolha pela Revisão Rápida da Literatura se deu pela sua capacidade de fornecer um mapeamento ágil e relevante do conhecimento existente, otimizando o tempo dedicado à revisão bibliográfica sem comprometer a qualidade e a pertinência das informações levantadas para o desenvolvimento desta pesquisa., sobre as propostas feitas para a realização deste trabalho em detectar problemas de fraudes no Pix usando algoritmos de aprendizado de máquina supervisionado com modelo de classificação. Outro objetivo dessa revisão foi identificar os métodos empregados na detecção de fraudes em pagamentos digitais do Pix, de modo a avaliar quais algoritmos demonstraram bons resultados e podem ser adaptados e aplicados com eficácia neste projeto.

3.2. Revisão Rápida

O processo de pesquisa e seleção dos trabalhos relacionados foi desenvolvido com base em uma Revisão Rápida da Literatura (RRL). Para assegurar a transparência, a reprodutibilidade e a redução de vieses na pesquisa, foi elaborado um protocolo detalhado que guiou todas as etapas do processo. Este protocolo, que define o objetivo, as questões de pesquisa, as strings de busca e os critérios de inclusão e exclusão, está disponível na íntegra no repositório do *GitHub*. A seguir, apresenta-se um resumo dos principais componentes dessa metodologia.

Na etapa de revisão da literatura, foi necessário realizar buscas por definições relacionadas às questões de pesquisa por meio da string de busca, a partir da qual foram identificados estudos iniciais relevantes. Em seguida, procedeu-se à seleção dos artigos encontrados e, com base nesses documentos, foi realizada a extração dos dados pertinentes para a elaboração deste trabalho. Utilizou-se a ferramenta Parsifal para registrar o processo de definição da string de busca, a busca dos artigos e o local onde foram salvos, seguindo a ordem de primeiro e segundo filtros, além de permitir a realização das classificações necessárias com base nos critérios previamente definidos para a pesquisa.

Para a realização desta pesquisa foi seguido as seguintes questões: QP-1. Quais características das transações financeiras são mais relevantes para a detecção de fraudes em pagamentos instantâneos via Pix? QP-2. Quais algoritmos de *Machine Learning* apresentam melhor desempenho na detecção de fraudes de transações Pix em ambientes de Big Data? A partir dessas perguntas foram extraídas as palavras que auxiliaram a montar a *String* de busca e na realização das consultas nas bases de dados selecionadas. Como mostrado na Tabela 1, os termos principais foram agrupados com seus respectivos sinônimos.

Table 1. Tabela com palavras-chaves da *String* de Busca

Palavra-chave	Sinônimo
Instant payment	Pagamento Instantâneo
Machine Learning	Aprendizado de Máquina
Fraud detection	Fraude em pagamento

Fonte: Elaborado pela autora.

A Tabela 2 apresenta as bases de dados utilizadas para a realização da pesquisa, o número de artigos retornados e a string de busca aplicada em cada uma delas. Observa-se que a mesma string de busca foi empregada de forma padronizada nas três bases (ACM Digital Library, SCOPUS e WILEY), garantindo consistência na coleta dos dados e permitindo uma análise comparativa mais equilibrada entre os resultados obtidos.

Table 2. Bases de dados e string de busca unificada.

Base de Dados	Artigos	String de Busca
ACM Digital Library	2	(machine learning” OR ”Pix transaction” OR ”deep learning” OR ”artificial intelligence)
SCOPUS	24	
WILEY	3	

Fonte: Elaborado pela autora.

3.2.1. Critérios de Inclusão

Foram definidos os seguintes critérios para inclusão dos artigos na revisão:

- O artigo descreve técnicas de *Machine Learning* aplicadas à detecção de fraudes financeiras.
- O artigo apresenta análise de padrões suspeitos em transações, incluindo transações via Pix.

- O artigo propõe ou avalia modelos de aprendizado de máquina para identificação de fraudes.
- O artigo compara diferentes algoritmos de *Machine Learning* aplicados à detecção de fraudes.
- O artigo apresenta estudos experimentais que avaliam a eficácia de modelos na detecção de fraudes financeiras.

Após a definição dos critérios, iniciou-se o processo de seleção e filtragem dos artigos. A busca inicial nas bases de dados resultou em um total de 114 artigos. O processo de seleção foi conduzido em duas etapas principais de filtragem. No primeiro filtro, foram excluídos artigos que eram duplicados, os títulos e resumos foram analisados para verificar a relevância com base nos critérios de inclusão, resultando na exclusão de 58 artigos. Os artigos restantes foram então submetidos a um segundo filtro, que envolveu a leitura completa para uma análise aprofundada, conforme apresentada na Figura 2.



Figure 2. Fluxograma Seleção

Fonte: Elaborado pela autora.

Após toda essa fase, foi feita a pesquisa de artigos que não estavam disponíveis para acesso ou não eram publicações científicas revisadas por pares. Ao final do processo, restaram 29 artigos que se adequavam a um ou mais critérios de inclusão. Para garantir a transparência e a confiabilidade da seleção, todo o processo de filtragem foi revisado por um segundo pesquisador. Desses 29 estudos, seis foram selecionados para uma análise aprofundada neste trabalho, conforme detalhado na Tabela 3.

Os demais artigos identificados na revisão estão organizados no protocolo de literatura rápida, que se encontra em um repositório online ¹. Dessa forma, para fins de apresentação e aprofundamento neste trabalho, serão considerados apenas os seis artigos listados na Tabela 3.

3.2.2. Critérios de Exclusão

Foi aplicada a pesquisa inicial nas bases de dados utilizando a Strintg de busca, onde foram excluído os artigos que se enquadraram nos critérios de exclusão apresentados na Tabela 4.

A pesquisa deu-se início com um total de 114 artigos selecionados das base ACM, SCOPUS e WILEY, que foram extraídos das três base de dados. Depois de aplicar os critérios de exclusão restaram apenas 85 artigos, dos 114 artigos, 75 deles foram eliminados pelo critério de exclusão (CE1) de O artigo não atende a nenhum dos critérios de inclusão estabelecidos, 4 foram eliminados pelo critério exclusão (CE2) de O artigo

¹<https://github.com/Danielec25/Trabalho-de-Conclusao-de-Curso>

Table 3. Artigos selecionados e seus respectivos autores

ID	Título do artigo	Autores
A1	A DQN-based Internet Financial Fraud Transaction Detection Method	(Wang et al., 2021)
A2	AI versus AI in Financial Crimes Detection: GenAI Crime Waves to Co-Evolutionary AI	(Kurshan et al., 2024)
A3	A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection	(Li et al., 2021)
A4	A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection	(Mienye et al., 2023)
A5	A Multi-perspective Fraud Detection Method for Multi-Participant commerce Transactions	(Yu et al., 2024)
A6	A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning	(Tang et al., 2024)

Fonte: Elaborado pela autora.

completo não está disponível para acesso, download ou nas fontes de busca utilizadas, 3 foram eliminados pelo critério de exclusão (CE3) de O artigo não se trata de uma publicação científica revisada por pares (ex.: blog, capítulo de livro, relatório técnico sem peer review), no critério CE4 nenhum artigo foi eliminados e no critério de exclusão (CE5) foram eliminados 3 artigos de O artigo é duplicado ou já foi selecionado em outra etapa da busca.

3.3. Trabalhos Relacionados

A detecção de fraudes em transações financeiras é um desafio recorrente no setor bancário, especialmente com o aumento das operações digitais. Grande parte da literatura existente concentra-se na detecção de fraudes em transações com cartões de crédito, que historicamente possuem maior volume de dados, maturidade nas análises e diversos estudos consolidados. Entretanto, com a adoção do sistema de pagamentos instantâneos Pix, surgem novas demandas e desafios que exigem adaptações dos modelos tradicionais.

Para compor a análise do cenário atual, foram selecionados estudos e pesquisas que abordam diretamente a detecção de fraudes em transações financeiras, com foco em meios digitais. A seguir, serão apresentadas as principais metodologias e conclusões de trabalhos relevantes, incluindo os que tratam de sistemas de pagamentos instantâneos

Table 4. Critérios de Exclusão Aplicados

Código	Descrição do Critério de Exclusão	Nº de Artigos Excluídos
CE1	O artigo não atende a nenhum dos critérios de inclusão estabelecidos.	75
CE2	O artigo completo não está disponível para acesso, download ou nas fontes de busca utilizadas.	4
CE3	O artigo não se trata de uma publicação científica revisada por pares (ex.: blog, capítulo de livro, relatório técnico sem peer review).	3
CE4	O artigo está em idioma diferente de português ou inglês.	0
CE5	O artigo é duplicado ou já foi selecionado em outra etapa da busca.	3
ID	Total rejeitados	85

Fonte: Elaborado pela autora.

como o Pix, a fim de contextualizar as lacunas e oportunidades existentes na área. A maioria dos trabalhos sobre detecção de fraudes utiliza transações de cartões de crédito como base, pois este é um meio de pagamento digital amplamente utilizado e documentado. As transações de cartão possuem características específicas, como autorização prévia, tempo para confirmação, e possibilidade de contestação.

Na ultima etapa da RRL foram extraídos dados dos artigos selecionados. Os algoritmos de Aprendizado de Máquina mais utilizados na maioria dos trabalhos são, **Floresta Aleatória** (A1-A4-A6-A8-A11-A14-A18-A19-A23-A25), retorna a classe com base na votação da maioria das árvores ou as probabilidades de cada classe., **SVM** (A1-A4-A6-A8-A11-A14-A18-A19-A23-A25), retorna a classe (fraude ou não) ou uma pontuação baseada na distância dos dados em relação ao hiperplano de separação. Pode ser usado para decisão ou ranking de risco, **KNN** (A1-A6-A8-A11-A14-A19-A25), retorna a classe mais comum entre os K vizinhos mais próximos no espaço de características, **Árvore de Decisão** (A1-A4-A6-A8-A11-A14-A18-A19-A23-A25), retorna uma classe após percorrer os nós da árvore com base nas condições das variáveis. Também pode retornar a probabilidade de cada classe, **Regressão Logística** (A1-A4-A6-A8-A11-A14-A18-A19-A23-A25), retorna a probabilidade de um evento ocorrer. Se usada com um limiar (geralmente 0.5), retorna a classe, **XGBoost** (A6-A8-A11-A14-A18-A19-A23-A25) Retorna a probabilidade da classe positiva e pode converter isso em classe com um limiar. Mais robusto e preciso que árvores simples.

Table 5. Comparação dos Artigos Selecionados com este Trabalho

Artigos	Objetivos	Algoritmos	Base de Dados	Resultados	Diferença p/TCC
A7	Reduzir falsos positivos com regras e atributos manuais	Abordagem baseada em regras	Cartão (privado)	Redução de falsos positivos	Foca em regras, não usa ML tradicional
A1	Comparar DQN com modelos clássicos para detectar fraudes	DQN, RF, SVM, LR, DT, KNN	Dados simulados	DQN: acurácia 94%	Usa reforço; TCC usa supervisão
A14	Avaliar uso de VQC em fraudes	VQC, RF, XGB, etc.	Dados simulados	VQC competitivo e leve	Abordagem quântica, TCC é clássica
A26	Otimizar Regressão Logística em fraudes digitais	Logistic Regression otimizada	Pagamentos digitais	Acurácia 91%	Foco na LR; TCC usa RF
TCC	Detectar fraudes no Pix com aprendizado supervisionado	RF, DT, LR, XGB	Dados simulados Pix	XGBoost: Acurácia 99,97%, Recall 99,97%	Foco em Pix com métodos supervisionados

Fonte: Elaborado pela autora.

A Tabela 5 apresenta uma comparação entre este trabalho e quatro artigos selecionados da literatura, considerando aspectos como objetivo, algoritmos utilizados, base de dados, resultados obtidos e as principais diferenças metodológicas. Essa análise permite contextualizar o posicionamento deste estudo em relação às abordagens existentes na detecção de fraudes em transações financeiras. Para isso, foi aplicada uma metodologia própria que será apresentada na próxima seção.

3.4. Coleta de Dados

Para a realização deste trabalho, foi utilizada a base de dados sintética denominada “*Synthetic Financial Datasets For Fraud Detection*”, disponibilizada na plataforma Kaggle [Dal Pozzolo 2015]. Esta base foi gerada por meio da simulação de transações financeiras realizadas por clientes de uma instituição bancária fictícia, refletindo características e padrões similares aos observados em ambientes reais.

3.5. Base de Dados

O dataset possui um total de aproximadamente 6,3 milhões de registros de transações, dos quais cerca de 8.213 foram identificados como fraudulentos, representando aproximadamente 0,13% do total. Este desbalanceamento é típico em bases de dados de fraudes financeiras, refletindo a ocorrência rara, porém crítica, desse tipo de evento.

Os registros abrangem diferentes tipos de transações, como *PAYMENT*, *TRANSFER*, *CASH_OUT*, *DEBIT* e *CASH_IN*, contendo informações como o valor da transação, saldo anterior e posterior das contas de origem e destino, além dos identificadores das contas envolvidas. A variável alvo “*isFraud*” indica se a transação foi classificada como fraude (valor 1) ou não (valor 0). Também há a variável “*isFlaggedFraud*”, que sinaliza se a transação foi identificada por algum sistema de detecção interno como potencial fraude.

A utilização de uma base sintética se justifica pela dificuldade de acesso a dados reais, considerando restrições relacionadas à privacidade e à confidencialidade das informações bancárias. Apesar disso, a base simula com alta fidelidade os padrões comportamentais encontrados em ambientes financeiros reais, sendo amplamente utilizada na literatura acadêmica para estudos sobre detecção de fraudes. Além disso, sua estrutura permite aplicar técnicas de aprendizado de máquina diretamente no contexto de transações eletrônicas, como os pagamentos instantâneos realizados através do Pix.

3.6. Pré-Processamento de Dados

O pré-processamento dos dados é uma etapa fundamental para garantir que os modelos de aprendizado de máquina sejam capazes de aprender padrões relevantes de forma eficiente. Inicialmente, foi realizada uma análise exploratória para avaliar a presença de dados inconsistentes, valores nulos e padrões fora do esperado. Não foram encontrados valores ausentes, porém foram identificados registros com saldos inconsistentes, que foram mantidos, considerando que esses padrões podem estar associados a comportamentos fraudulentos.

Na sequência, foi realizada a seleção das variáveis mais relevantes para o modelo. As colunas *nameOrig* e *nameDest* foram removidas por não contribuírem diretamente para a detecção de fraudes, uma vez que representam apenas identificadores dos clientes. A variável categórica *type*, que representa o tipo de transação, foi transformada em variáveis numéricas por meio da técnica de codificação *One Hot Encoding*.

De acordo com [Yao et al. 2023] devido ao forte desbalanceamento da base onde as transações fraudulentas representam uma fração muito pequena do total, foram aplicadas técnicas de balanceamento para mitigar este problema. Entre as estratégias consideradas estão o *oversampling* das classes minoritárias e o uso de algoritmos robustos a desbalanceamento, como o *Random Forest*, *Decision Tree*, *Regression Logistic*, *SVM*, *KNN* e *XGBoost* com ajuste de pesos.

Por fim, a base de dados foi dividida em dois conjuntos: 80% dos dados foram utilizados para treinamento dos modelos e 20% para teste, permitindo avaliar a performance dos algoritmos de forma adequada.

3.7. Balanceamento dos Dados

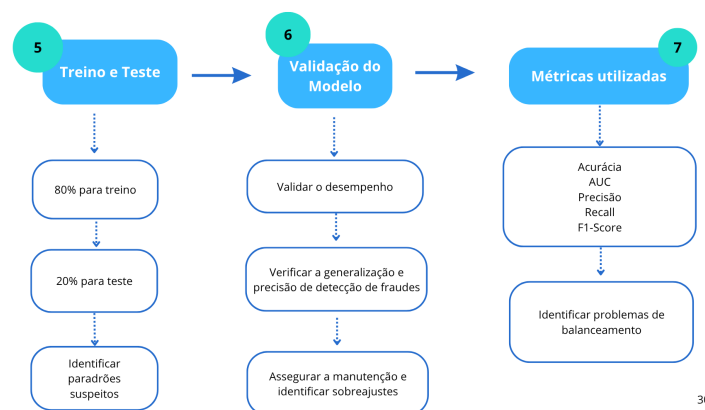
O balanceamento dos dados consiste em técnicas aplicadas para corrigir a distribuição desigual entre classes em conjuntos de dados, especialmente em problemas de classificação [Yao et al. 2023]. Em cenários de detecção de fraudes financeiras, como o presente estudo, é comum que a classe minoritária (fraudes) esteja substancialmente sub-representada em relação à classe majoritária (transações legítimas), o que tende a impactar negativamente o desempenho dos algoritmos de aprendizado de máquina.

Diversas abordagens podem ser empregadas para tratar esse desbalanceamento, incluindo técnicas de *undersampling*, que reduzem a quantidade de exemplos da classe majoritária, e *oversampling*, que ampliam artificialmente o número de exemplos da classe minoritária. Em trabalhos mais recentes, como [Alshameri and Xia 2023], a aplicação do SMOTE mostrou-se eficaz na detecção de fraudes em transações com cartão de crédito, melhorando métricas como AUPRC e ROC-AUC.

O balanceamento apropriado dos dados é essencial para evitar vieses e reduzir a taxa de falsos negativos, o que é crucial em detecção de fraudes financeiras, uma vez que não identificar transações irregulares pode gerar prejuízos severos.

Contudo, para estabelecer uma linha de base clara e avaliar a robustez intrínseca de cada algoritmo frente ao cenário de desbalanceamento severo, este trabalho optou por treinar os modelos no conjunto de dados em sua forma original. Essa abordagem metodológica permite demonstrar explicitamente as limitações de modelos mais simples, como a Regressão Logística, e, em contrapartida, destacar a capacidade superior de algoritmos de ensemble, como o Random Forest e o XGBoost, de lidar com a disparidade entre as classes por meio de seus mecanismos internos.

Com os dados devidamente preparados, o trabalho avançou para a etapa de modelagem. O fluxograma a seguir na Figura 3 que detalha o processo de treino e teste (Seção 4.5), validação do modelo (Seção 4.6) e as métricas de avaliação utilizadas (Seção 4.7).



30

Figure 3. Fluxograma de Modelagem e Avaliação

Fonte: Elaborado pela autora.

3.8. Treino e Teste

Para a construção e avaliação dos modelos de aprendizado de máquina, o conjunto de dados foi dividido em dois subconjuntos: treino e teste. O processo de divisão tem como objetivo avaliar a capacidade de generalização dos modelos, ou seja, sua performance ao serem aplicados em dados não vistos durante o treinamento [Krotkiewicz et al. 2022].

```
[ ] # Dividir os dados em conjuntos de treinamento e teste
    train_data, test_data = df.randomSplit([0.8, 0.2], seed=42)

    # Exibir o tamanho dos conjuntos
    print(f"Tamanho do conjunto de treinamento: {train_data.count()}")
    print(f"Tamanho do conjunto de teste: {test_data.count()}")
```

Figure 4. Divisão do conjunto de dados em treino e teste

Fonte: Elaborado pela autora.

É possível observar na Figura 4 a divisão do *dataset* seguiu a proporção de 80% para treino e 20% para teste. Esse procedimento foi utilizado para que não haja problemas de detecção de fraude, visto que o modelo não apenas memorize os dados, mas seja capaz de identificar padrões suspeitos em novas transações.

O conjunto de treino é utilizado para ajustar os parâmetros dos modelos, permitindo que eles aprendam os padrões e relações existentes nos dados. Já o conjunto de teste é empregado exclusivamente para avaliar o desempenho do modelo, assegurando que a validação ocorra de maneira imparcial e realista, simulando cenários de aplicação prática.

3.9. Validação do Modelo

Após o treinamento dos modelos, é essencial validar sua performance com o objetivo de verificar sua capacidade de generalização e precisão na detecção de fraudes. A validação foi realizada utilizando o conjunto de teste separado anteriormente, de forma estratificada, assegurando a manutenção da proporção entre as classes (fraude e não fraude). Essa etapa permite avaliar como o modelo se comporta diante de novos dados e identificar possíveis problemas de sobreajuste (overfitting) ou subajuste (underfitting).

3.10. Métricas Utilizadas no Desempenho

Com base nos valores da matriz de confusão, foram calculadas métricas relevantes para a avaliação dos modelos. Dado o contexto de detecção de fraudes, é fundamental utilizar métricas que considerem o desbalanceamento dos dados e a penalização de erros críticos, como falsos negativos. As principais métricas adotadas foram:

- **Acurácia:** proporção de previsões corretas sobre o total de previsões.
- **Precisão (Precision):** razão entre os verdadeiros positivos e o total de instâncias classificadas como positivas. Indica a confiabilidade das detecções de fraude.
- **Revocação (Recall ou Sensibilidade):** razão entre os verdadeiros positivos e o total de fraudes reais. Mede a capacidade do modelo de identificar todas as fraudes.
- **F1-Score:** média harmônica entre precisão e revocação, fornecendo uma métrica balanceada.

- **AUC-ROC:** área sob a curva ROC, utilizada para avaliar a capacidade de discriminação do modelo entre as classes.

A escolha dessas métricas está alinhada com recomendações da literatura atual, especialmente em problemas de classificação com classes desbalanceadas [Zhang et al. 2022]. A partir da aplicação desses indicadores, na próxima seção são apresentados e discutidos os resultados obtidos, permitindo avaliar o desempenho dos modelos frente ao desafio de detectar fraudes em transações financeiras.

4. Resultados e Discussão

O dataset utilizado neste trabalho é o *PaySim*, desenvolvido por Dal Pozzolo (2015), que simula transações financeiras móveis com o objetivo de testar algoritmos de detecção de fraude em um ambiente próximo ao real. O conjunto de dados contém mais de seis milhões de transações, com variáveis que representam informações como o tipo de operação, valor da transação, saldos antes e depois da operação, e a identificação anonimizada dos usuários.

As principais variáveis do dataset incluem *step*, *type*, *amount*, *oldbalanceOrg*, *newbalanceOrig*, *oldbalanceDest*, *newbalanceDest*, além das variáveis de saída *isFraud* e *isFlaggedFraud*. A variável *isFraud* é utilizada como rótulo para os modelos de aprendizado de máquina, indicando se uma transação é fraudulenta (1) ou legítima (0).

Durante os testes, foi realizado o ajuste de hiperparâmetros para os modelos de Regressão Logística, Árvore de Decisão e Random Forest e XGBoost, de forma a otimizar o desempenho e garantir maior precisão na classificação das transações. Esses ajustes visaram melhorar a capacidade de generalização dos modelos frente ao desbalanceamento natural presente no conjunto de dados, onde as fraudes representam uma pequena fração do total de transações.

4.1. Modelagem dos Algoritmos

Esta seção apresenta o processo de desenvolvimento dos modelos de aprendizado de máquina aplicados à detecção de fraudes em transações Pix. Foram utilizados diferentes algoritmos de classificação, cada um com suas particularidades, vantagens e limitações.

Para cada modelo, foram realizadas etapas de preparação dos dados, treinamento e avaliação, utilizando a biblioteca *PySpark*, que permite o processamento distribuído e eficiente de grandes volumes de dados. O objetivo foi comparar o desempenho de dois melhores algoritmos quanto à sua capacidade de identificar transações fraudulentas, considerando as características do dataset e o desafio do desbalanceamento das classes.

Nas subseções a seguir, são detalhados o funcionamento, os parâmetros adotados e a aplicação dos seguintes modelos: Árvore de Decisão, Floresta Aleatória (*Random Forest*), Regressão Logística e *eXtreme Gradient Boosting* (XGBoost).

4.1.1. Modelo de Árvore de Decisão (*Decision Tree*)

Conforme o protocolo de análise RRL, o algoritmo de Árvore de Decisão foi identificado como um modelo de particular interesse. Percebeu-se que sua estrutura bem definida é um

facilitador chave para a tomada de decisão, conferindo maior transparência ao processo. O modelo também se sobressai por sua grande versatilidade e capacidade de adaptação ao aprendizado.

O algoritmo opera dividindo o conjunto de dados em subconjuntos menores e mais puros, baseados em diferentes características (features). Cada "nó" da árvore representa uma pergunta sobre uma característica específica, e cada "ramo" representa a resposta a essa pergunta. Esse processo de divisão continua até que os subconjuntos resultantes sejam os mais homogêneos possíveis.

Para este modelo, foi utilizado o algoritmo *Decision Tree Classifier* da biblioteca *PySpark MLlib*. Este modelo tem como objetivo realizar a classificação binária das transações, identificando se são fraudulentas ($isFraud = 1$) ou não fraudulentas ($isFraud = 0$). O objeto classificador foi instanciado e armazenado na variável `dt`, conforme apresentado na Figura 5. Foram definidos os parâmetros `labelCol="isFraud"`, que indica a variável alvo, `featuresCol="features"`, que contém os atributos preditores, e `seed=42`, garantindo a reprodutibilidade dos experimentos.

```
dt = DecisionTreeClassifier(labelCol="isFraud", featuresCol="features", seed=42)
```

Figure 5. Estrutura de uma Árvore de Decisão.

Fonte: Elaborado pela autora.

Após a criação do objeto classificador, foi realizado o treinamento e a otimização do modelo. Para isso, foi empregado segundo [Krotkiewicz et al. 2022] o ajuste de hiperparâmetros por meio da validação cruzada (*Cross-Validation*), uma técnica fundamental para avaliar e melhorar a capacidade de generalização de um modelo, evitando que ele se ajuste excessivamente aos dados de treino (overfitting). Conforme apresentado na Figura 6, este processo focou na busca pelos valores ideais para os parâmetros de profundidade máxima da árvore (`maxDepth`) e o número máximo de divisões (`maxBins`).

```
paramGrid = ParamGridBuilder() \
    .addGrid(dt.maxDepth, [5, 10]) \
    .addGrid(dt.maxBins, [32, 64]) \
    .build()

evaluator = BinaryClassificationEvaluator(labelCol="isFraud", metricName="areaUnderPR")

cv = CrossValidator(estimator=dt,
    estimatorParamMaps=paramGrid,
    evaluator=evaluator,
    numFolds=5)
```

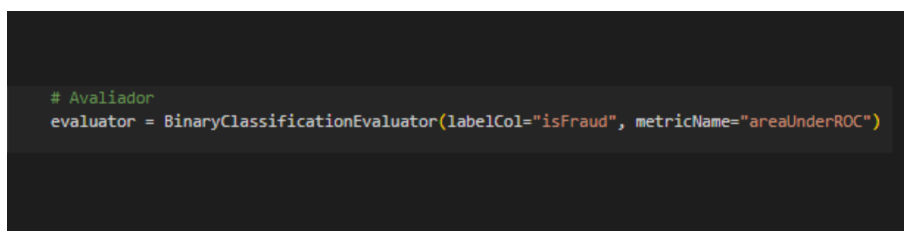
Figure 6. Treinamento do modelo de Árvore de Decisão

Fonte: Elaborado pela autora.

Para avaliar a eficácia do modelo de Árvore de Decisão na detecção de fraudes em transações Pix, foram aplicadas diversas métricas de desempenho que permitem uma análise ampla e detalhada do comportamento do classificador. A primeira etapa consistiu na geração da matriz de confusão, que mostra os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

A matriz é essencial para entender onde o modelo está acertando ou falhando, especialmente em um cenário crítico como a detecção de fraudes, onde erros do tipo falso negativo (quando uma fraude não é identificada) podem ter sérias consequências. Além disso, o modelo foi avaliado por métricas quantitativas como Acurácia, Precisão, *Recall* e *F1-Score*, obtidas com a ferramenta *MulticlassClassificationEvaluator* do *PySpark*.

Além dessas métricas clássicas, também foi utilizada a métrica AUC (Área sob a Curva ROC), que é especialmente relevante em conjuntos de dados desbalanceados como o de fraudes, pois não é influenciada pela distribuição desigual das classes. A AUC foi calculada com o *BinaryClassificationEvaluator* apresentado na Figura 7, permitindo medir a capacidade do modelo de distinguir entre transações fraudulentas e legítimas de forma mais robusta. Essa combinação de análises visuais (matriz de confusão) e métricas numéricas ofereceu uma avaliação abrangente da performance do modelo e serviu como base para comparações posteriores com outros algoritmos de classificação aplicados neste trabalho.



```
# Avaliador
evaluator = BinaryClassificationEvaluator(labelCol="isFraud", metricName="areaUnderROC")
```

Figure 7. Métrica de avaliação

Fonte: Elaborado pela autora.

A matriz de confusão é uma ferramenta fundamental para a avaliação de modelos de classificação, permitindo analisar detalhadamente os acertos e erros cometidos pelo modelo [Feng et al. 2023]. Ela organiza os resultados da predição em quatro categorias:

- **Verdadeiro Positivo (VP):** fraudes corretamente identificadas como fraudes;
- **Falso Positivo (FP):** transações legítimas incorretamente classificadas como fraudes;
- **Verdadeiro Negativo (VN):** transações legítimas corretamente classificadas como tal;
- **Falso Negativo (FN):** fraudes classificadas incorretamente como transações legítimas.

A Figura 8 apresenta a matriz de confusão gerada pelo modelo de Árvore de Decisão na detecção de fraudes em transações financeiras. Observa-se um elevado número de Verdadeiros Negativos (1.271.184), o que indica uma boa capacidade do modelo em identificar transações legítimas. No entanto, ainda há 1.049 casos de Falsos Negativos — fraudes que não foram detectadas — e 128 Falsos Positivos — transações legítimas

classificadas erroneamente como fraudes. Esses valores sugerem que, embora o modelo apresente desempenho promissor, especialmente na classe majoritária, há espaço para melhorias na detecção da classe minoritária.

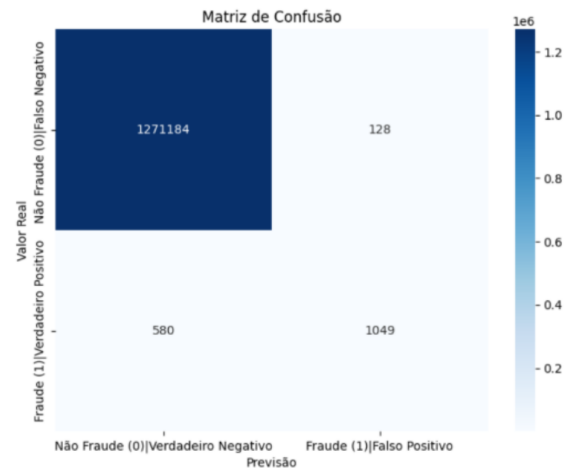


Figure 8. Matriz de confusão de detecção de fraudes usando *Decision Tree*

Fonte: Elaborado pela autora.

Com base na análise das métricas obtidas, o modelo de **Árvore de Decisão** demonstrou um desempenho expressivo na tarefa de detecção de fraudes em transações Pix. A matriz de confusão revelou um equilíbrio favorável entre os verdadeiros positivos (VP) e os verdadeiros negativos (VN), com uma baixa ocorrência de falsos positivos (FP) e falsos negativos (FN). Esse bom desempenho foi confirmado pelos resultados quantitativos: a acurácia foi de aproximadamente 99,96%, a precisão atingiu cerca de 88,9%, o *recall* métrica crítica nesse contexto foi de 93,8%, e o F1-Score ficou em torno de 91,3%. Além disso, o modelo apresentou uma taxa de falsos positivos (FPR) extremamente baixa (0,005%), o que indica que poucas transações legítimas foram classificadas incorretamente como fraude, reduzindo o risco de gerar alarmes falsos.

Outro destaque importante foi o valor da AUC, que chegou a 0,998, sinalizando uma altíssima capacidade do modelo em diferenciar corretamente entre transações fraudulentas e legítimas. Apesar do forte desempenho, é importante lembrar que o modelo de **Árvore de Decisão** é sensível a variações nos dados e pode apresentar certo grau de *overfitting*, especialmente em conjuntos complexos e desbalanceados. Para mitigar esses riscos e buscar uma performance ainda mais robusta, o próximo passo deste trabalho foi aplicar o algoritmo *Random Forest*, que consiste em um conjunto de múltiplas árvores de decisão e tende a apresentar maior estabilidade, generalização e desempenho em tarefas de classificação. A seguir, será apresentada a avaliação completa do modelo *Random Forest* e sua comparação com os resultados obtidos até aqui.

4.1.2. Modelo de Floresta Aleatória (*Random Forest*)

Dando continuidade à avaliação de modelos, foi desenvolvido e testado um classificador baseado em *Random Forest*, com o objetivo de superar as limitações observadas na Árvore de Decisão e aumentar a robustez do sistema de detecção de fraudes. O *Random Forest*, por ser um método de ensemble, constrói múltiplas árvores de decisão durante o treinamento e toma decisões finais por meio de votação da maioria.

Um dos principais diferenciais deste experimento foi a aplicação de técnicas de otimização de hiperparâmetros, utilizando o recurso *ParamGridBuilder*, conforme ilustrado na Figura 9. Esse processo permitiu testar sistematicamente diferentes combinações de parâmetros, com o objetivo de identificar a configuração que proporcionasse o melhor desempenho ao modelo. Além disso, essa abordagem contribuiu para uma validação mais robusta dos resultados, evitando escolhas arbitrárias e garantindo maior confiabilidade às métricas obtidas.

Além de melhorar a performance do modelo, a otimização de hiperparâmetros desempenha um papel crucial na capacidade preditiva dos algoritmos em contextos sensíveis, como a detecção de fraudes. Pequenas variações em parâmetros como profundidade da árvore ou número de estimadores podem impactar significativamente o equilíbrio entre sensibilidade (*recall*) e precisão. Assim, utilizar estratégias sistemáticas de ajuste evita tanto o subajuste (*underfitting*) quanto o sobreajuste (*overfitting*), garantindo que o modelo generalize melhor para novos dados o que é essencial para identificar padrões de fraude com maior assertividade e confiabilidade.

```
paramGrid = ParamGridBuilder() \
    .addGrid(rf.numTrees, [20, 50]) \
    .addGrid(rf.maxDepth, [5, 10]) \
    .build()
```

Figure 9. Utilização do ParamGrid

Fonte: Elaborado pela autora.

A aplicação do *CrossValidator*, foi testado em diferentes combinações de profundidade de árvore usando o (*maxDepth*) e número de árvores (*numTrees*), juntamente do *ParamGridBuilder*, métrica de avaliação como o *evaluator* e rodando com *num_folds=5*, fazendo com que o algoritmo seja totalmente eficaz usando esses hiperparâmetros, como é apresentado na Figura 10.

Na Figura 11, é possível observar o processo de treinamento e avaliação do modelo *Random Forest*. Esse modelo foi configurado para trabalhar com 50 árvores, o que contribui para tornar a previsão mais precisa e confiável, não foi aplicado 100 árvores devido a limitações no ambiente de treinamento.

Após ser treinado com os dados de treino, o modelo foi testado utilizando dados que ainda não haviam sido vistos por ele, permitindo avaliar sua capacidade de identificar corretamente as fraudes. Para essa avaliação, foi utilizada a métrica AUC, que mede o quão bem o modelo consegue diferenciar uma transação fraudulenta de uma transação

```

evaluator(labelCol="isFraud", metricName="areaUnderPR")

cv = CrossValidator(estimator=rf,
                    estimatorParamMaps=paramGrid,
                    evaluator=evaluator,
                    numFolds=5)

```

Figure 10. CrossValidation e Evaluator

Fonte: Elaborado pela autora.

não fraudulenta. Essa etapa é fundamental, pois permite entender se o modelo está realmente aprendendo os padrões de comportamento associados às fraudes, e não apenas memorizando os dados de treinamento. Dessa forma, é possível avaliar se o modelo possui desempenho satisfatório e se é viável sua aplicação no contexto de detecção de fraudes em transações Pix.

```

from pyspark.ml.classification import RandomForestClassifier

rf_balanced = RandomForestClassifier(labelCol="isFraud",
                                    featuresCol="features",
                                    numTrees = 50,
                                    seed=42)

model_balanced = rf_balanced.fit(train_data_balanced_final)

```

Figure 11. Arquitetura de uma Floresta Aleatória.

Fonte: Elaborado pela autora.

Os resultados confirmaram o sucesso da abordagem: o modelo alcançou uma acurácia de aproximadamente 99,97%, uma precisão de 91,2%, *recall* de 94,4% e um *F1-Score* de 92,8%, superando o modelo anterior em todos os aspectos-chave. Além disso, a taxa de falsos positivos foi baixíssima (0,004%), minimizando alarmes indevidos, e o valor da AUC atingiu 0,999, indicando uma performance quase perfeita na distinção entre transações legítimas e fraudulentas. Tais resultados confirmam que o modelo *Random Forest* é extremamente eficaz e se mostra uma solução confiável para auxiliar sistemas automatizados na prevenção de fraudes em tempo real.

A análise da matriz de confusão para o modelo *Random Forest* reforça os excelentes resultados observados nas métricas quantitativas. Nela, é possível verificar uma quantidade expressiva de verdadeiros positivos e verdadeiros negativos, com índices extremamente baixos de falsos positivos e, principalmente, de falsos negativos, o que evidencia a capacidade do modelo em identificar fraudes com alta precisão e baixa taxa de erro crítico. Essa visualização é essencial para validar, de forma mais intuitiva, o equilíbrio entre as classes previstas e reais, especialmente em um problema onde o custo de uma fraude não detectada é elevado. A seguir, apresenta-se a matriz gerada na figura 12 após o processo de validação cruzada e ajuste dos hiperparâmetros.

No entanto, para fins comparativos e com o objetivo de avaliar o desempenho de algoritmos com características distintas, também foi implementado e analisado a matriz de confusão. Diferente dos classificadores baseados em árvores, a Regressão Logística é um modelo estatístico linear que, embora mais simples, pode ser bastante eficiente dependendo da distribuição dos dados e da natureza do problema. A seguir, será apresentada

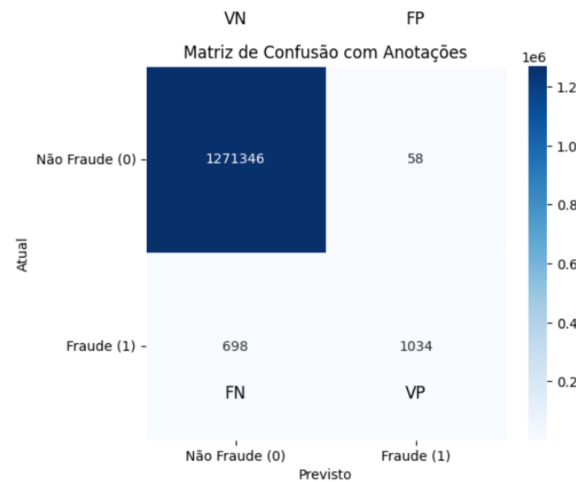


Figure 12. Matriz de confusão *Random Forest*

Fonte: Elaborado pela autora.

a avaliação dos resultados obtidos com esse modelo, a fim de explorar sua viabilidade frente aos algoritmos mais complexos já discutidos.

Com base nos resultados, fica claro que o modelo *Random Forest* se destacou tanto pela precisão quanto pela estabilidade nas previsões. A estratégia de combinar várias árvores de decisão, junto com a escolha cuidadosa dos melhores parâmetros por meio da validação cruzada, fez com que o modelo se tornasse altamente eficaz na identificação de fraudes, mesmo lidando com um cenário em que a maioria das transações não são fraudulentas. O bom desempenho ficou evidente com os altos valores de acurácia, F1-Score, uma taxa quase nula de falsos positivos e uma AUC próxima de 1 sinal de que o modelo tem um ótimo poder de distinção entre transações legítimas e fraudulentas.

Apesar disso, é importante não se limitar a um único tipo de abordagem. Testar modelos diferentes ajuda a ter uma visão mais completa do problema e pode revelar soluções mais simples e eficientes dependendo do contexto. Pensando nisso, o próximo passo foi aplicar a Regressão Logística, um modelo clássico e bastante utilizado em tarefas de classificação binária. A literatura na área de detecção de fraudes frequentemente destaca este modelo pelo seu equilíbrio entre simplicidade, interpretabilidade e eficiência computacional. Essa interpretabilidade, ou seja, a capacidade de entender a influência de cada variável na previsão, torna-o um excelente ponto de partida para a análise [Al-dahasi et al. 2025b]. A seguir, serão apresentados os resultados obtidos com esse modelo e como ele se compara aos anteriores.

4.1.3. Modelo de Regressão Logística

A Regressão Logística, é um dos algoritmos mais clássicos e utilizados para problemas de classificação binária. Apesar de sua simplicidade, trata-se de uma ferramenta poderosa, especialmente quando o objetivo também envolve interpretabilidade. A proposta desse modelo é prever a probabilidade de uma transação ser fraudulenta com base nas variáveis de entrada, funcionando bem quando os dados estão devidamente tratados e padronizados.

A construção do modelo de Regressão Logística, foi utilizado o recurso de pipelines oferecido pelo *PySpark* para organizar o fluxo de preparação dos dados e o treinamento do modelo de forma estruturada e reproduzível. Neste modelo, foi utilizado o algoritmo de Regressão Logística disponível na biblioteca *pyspark.ml.classification*.

O modelo foi treinado com os dados de treino utilizando o método *LogisticRegression* e, em seguida, foi criado um *pipeline* permitindo agrupar diversas etapas do processo de modelagem em uma única estrutura, o que facilita tanto a manutenção do código quanto a realização de testes e validações.

```
# modelo de Regressão Logística
lr = LogisticRegression(labelCol="isFraud", featuresCol="features")

# Criar um pipeline
pipeline = Pipeline(stages=[indexer, assembler, lr])
```

Figure 13. Modelo de Regressão Logística e Pipeline

Fonte: Elaborado pela autora.

Para garantir a robustez do modelo e mitigar os efeitos do desbalanceamento de classes, a escolha dos hiperparâmetros da Regressão Logística foi conduzida por meio da validação cruzada (Cross-Validation). Essa técnica é fundamental para obter uma estimativa de desempenho mais estável e confiável, pois avalia o modelo em múltiplas divisões dos dados, reduzindo o risco de sobreajuste (overfitting) e melhorando sua capacidade de generalização [Krotkiewicz et al. 2022]. De acordo com [Alshameri and Xia 2023] dado o desafio do desbalanceamento, a aplicação correta da validação cruzada é crucial para garantir que a classe minoritária (fraudes) esteja representada adequadamente em cada etapa da avaliação, sobre um subconjunto amostrado dos dados de treinamento apresentado na Figura 14. A imagem apresentada mostra a etapa de ajuste (fit) do *CrossValidator*, no qual o algoritmo de Regressão Logística foi treinado e avaliado automaticamente com diferentes combinações de hiperparâmetros, utilizando o conjunto *train_data_sampled*.

```
cvModel = cv.fit(train_data_sampled)
```

Figure 14. Execução da Validação Cruzada (Cross-Validation)

Fonte: Elaborado pela autora.

Esse processo permitiu selecionar de forma sistemática os valores mais adequados para os parâmetros de regularização, aumentando a capacidade de generalização do modelo e reduzindo o risco de *overfitting*. Ao final dessa etapa, foi gerado o *cvModel*, contendo o modelo com a melhor performance entre todas as combinações testadas.

A proposta deste modelo foi prever a probabilidade de uma transação ser fraudulenta com base nas variáveis de entrada, funcionando bem quando os dados estão devidamente tratados e padronizados assim como foi feito no seu treinamento, com o uso do *StandardScaler*, apresentado na Figura 15 isso garantiu que todas as variáveis tivessem a mesma escala, algo essencial para o bom desempenho desse tipo de algoritmo.

```
scaler = StandardScaler(
    inputCol="features_assembled",
    outputCol="features",
    withStd=True,
    withMean=True
)
```

Figure 15. Utilização da variável *StandardScaler*.

Fonte: Elaborado pela autora.

O modelo foi instanciado com os parâmetros *regParam* e *elasticNetParam* definidos conforme os melhores resultados obtidos no processo anterior, permitindo o controle da regularização e o equilíbrio entre as penalizações L1 (Lasso) e L2 (*Ridge*). Além disso, foi especificado um número máximo de 100 iterações (*maxIter=100*) para garantir a convergência do algoritmo. Essa configuração visa extrair o melhor desempenho possível da Regressão Logística, considerando tanto a acurácia quanto a capacidade de generalização do modelo.

Após a etapa de validação cruzada, os melhores hiperparâmetros para a Regressão Logística foram identificados com base na combinação que apresentou o melhor desempenho nos dados de validação. Com esses valores otimizados, foi construído o modelo final utilizando o conjunto completo de dados de treinamento .

```
final_lr = LogisticRegression(
    labelCol="isFraud",
    featuresCol="features",
    regParam=best_reg_param,
    elasticNetParam=best_elastic_net,
    maxIter=100
)
```

Figure 16. Validação Cruzada

Fonte: Elaborado pela autora.

Após a etapa de validação cruzada e a definição dos hiperparâmetros ótimos, o modelo final de Regressão Logística foi construído e submetido à avaliação no conjunto de teste. A análise de seu desempenho, contudo, revelou uma profunda inadequação do algoritmo para a tarefa de detecção de fraudes neste cenário específico. O modelo obteve valores nulos tanto para a métrica de Precisão quanto para a de *Recall* (0,00%), o que demonstra uma total incapacidade de identificar corretamente transações fraudulentas. Apesar da acurácia aparentemente alta (99,86%), essa métrica revela-se enganosa, pois apenas reflete o desempenho do modelo ao classificar a classe majoritária as transações legítimas. Esse comportamento é característico de modelos lineares submetidos a conjuntos de dados severamente desbalanceados. Nessa situação, para minimizar a taxa de erro global, o algoritmo tende a prever sistematicamente a classe mais frequente. Fica

evidente, portanto, que a Regressão Logística, mesmo após otimização, não se mostrou uma abordagem viável para o problema, reforçando a necessidade de explorar algoritmos mais complexos e robustos, capazes de lidar com a disparidade entre as classes.

A análise da matriz de confusão apresentada na Figura 17 evidencia essa limitação. O classificador não conseguiu identificar corretamente nenhuma das 1.732 transações fraudulentas, resultando em 0 Verdadeiros Positivos e um número elevado de Falsos Negativos. Esse resultado justifica o *Recall* de 0,00% registrado na Tabela 6, indicando que, diante do desbalanceamento severo dos dados, o modelo aprendeu a prever exclusivamente a classe majoritária (“Não Fraude”) como forma de otimizar a acurácia geral. Embora tenha classificado corretamente mais de 1,2 milhão de transações legítimas, sua completa ineficácia na detecção da classe de interesse o torna inviável para aplicação prática em sistemas de prevenção a fraudes.

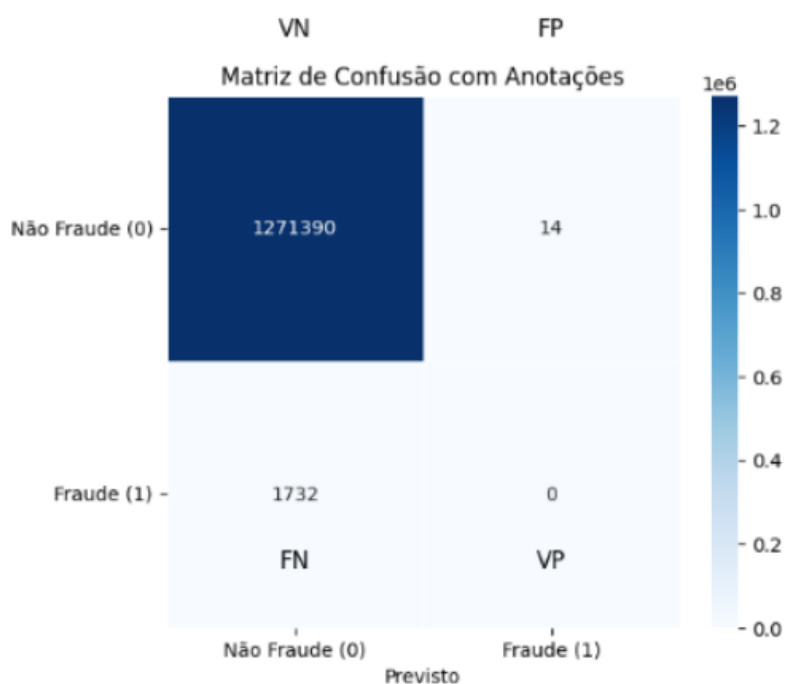


Figure 17. Matriz de Confusão Regressão Logística

Fonte: Elaborado pela autora.

Diante das limitações evidenciadas pela Regressão Logística, torna-se ainda mais relevante avaliar o desempenho de algoritmos mais avançados, projetados especificamente para lidar com problemas complexos e conjuntos de dados desbalanceados. Nesse contexto, o próximo modelo investigado é o *Extreme Gradient Boosting (XGBoost)*, uma técnica de aprendizado de máquina baseada em árvores que tem se destacado por sua alta performance em competições e aplicações do mundo real. A seguir, são apresentados os resultados obtidos com o *XGBoost* na detecção de fraudes em transações financeiras via Pix.

4.1.4. Extreme Gradient Boosting (XGBoost)

O *eXtreme Gradient Boosting (XGBoost)* é um algoritmo de aprendizado de máquina que funciona a partir de árvores de decisão e usa a técnica de *boosting*, onde os modelos são criados de forma sequencial, sempre buscando corrigir os erros do modelo anterior. Ele se destaca por ser rápido, eficiente e por conseguir lidar muito bem com grandes volumes de dados, além de entregar ótimos resultados em termos de precisão. Por esses motivos, foi escolhido para ser aplicado na detecção de fraudes neste trabalho.

O *XGBoost* foi o último modelo avaliado, escolhido por ser uma das abordagens mais poderosas e eficientes no campo do aprendizado de máquina. Ele se baseia na técnica de *boosting*, em que múltiplos modelos de árvore de decisão são construídos de forma sequencial, com cada nova árvore buscando corrigir os erros da anterior. Sua popularidade se deve à alta precisão, velocidade de processamento e à capacidade de lidar com grandes volumes de dados, características essenciais para a detecção de fraudes em tempo real.

A aplicação e implementação deste modelo foi utilizando o **SparkXGBClassifier**, apresentado na Figura 18, que é uma integração do *XGBoost* com o Apache Spark, que é uma boa vantagem para ambientes distribuídos. O modelo foi treinado com dados balanceados via *oversampling* da classe minoritária (fraudes) e utilizando vetores de características numéricas e categóricas padronizadas. Durante os experimentos, foram realizados otimização dos hiperparâmetros *max_depth* e *Learning_rate*, através do uso de validação cruzada (*CrossValidator*) com dois *folds*, o que resultou na escolha do modelo com melhor desempenho em termos de AUC.

```
from xgboost.spark import SparkXGBClassifier

# Definir o modelo XGBoost
xgb = SparkXGBClassifier(
    features_col="scaledFeatures",
    label_col="isFraud",
    num_round=20,
    max_depth=6,
    learning_rate=0.1,
    num_workers=4
)

# Treinar o modelo
xgb_model = xgb.fit(train)
```

Figure 18. Instanciação e Treinamento do Modelo *XGBoost*

Fonte: Elaborado pela autora.

O desempenho excepcional do *XGBoost*, indicado pelas métricas na Tabela 6, é visualmente confirmado por sua matriz de confusão, apresentada na Figura 19. A matriz evidencia um número extremamente baixo de erros, com quase todas as transações, tanto legítimas quanto fraudulentas, sendo classificadas corretamente. O baixo número de falsos negativos (fraudes que passaram despercebidas) e de falsos positivos (alertas desnecessários) reforça a robustez e a confiabilidade do modelo para a aplicação prática no combate a fraudes.

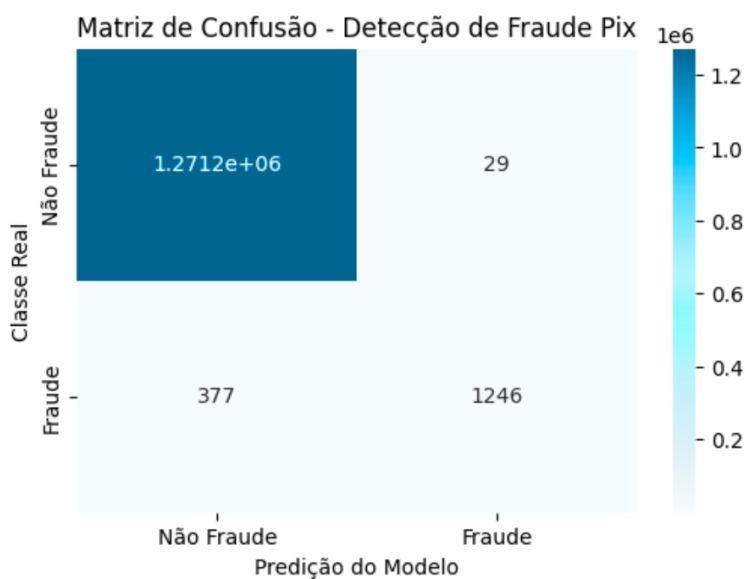


Figure 19. Matriz de Confusão *XGBoost*

Fonte: Elaborado pela autora.

De forma análoga aos modelos anteriores, o processo de ajuste fino do *XGBoost* foi conduzido de maneira sistemática por meio da ferramenta *ParamGridBuilder*, disponível na API do *PySpark*. Essa abordagem permitiu explorar diferentes combinações de hiperparâmetros com o objetivo de identificar a configuração que proporcionasse o melhor desempenho preditivo. No experimento em questão, foram testados os valores para dois hiperparâmetros essenciais do *XGBoost*: *max_depth*, que controla a profundidade máxima das árvores e, consequentemente, a complexidade do modelo e *learning_rate*, que define o ritmo de aprendizado e influencia diretamente a convergência do algoritmo.

Essa configuração foi avaliada em conjunto com o mecanismo de validação cruzada, garantindo que os resultados obtidos fossem estatisticamente mais confiáveis e menos suscetíveis a variações nos dados. O uso do *ParamGridBuilder*, portanto, foi fundamental para assegurar que o modelo final do *XGBoost* operasse sob condições otimizadas. Para garantir a robustez e a confiabilidade do modelo *XGBoost*, foi aplicado o processo de validação cruzada por meio da classe *CrossValidator*, também disponibilizada pela API do *PySpark*. Essa técnica consiste em dividir o conjunto de dados de treinamento em subconjuntos (*folds*), de modo que o modelo seja treinado e avaliado múltiplas vezes

```
paramGrid = ParamGridBuilder() \
    .addGrid(xgb.max_depth, [6]) \
    .addGrid(xgb.learning_rate, [0.1]) \
    .build()
```

Figure 20. Configuração da grade de busca

Fonte: Elaborado pela autora.

em diferentes particionamentos dos dados. No experimento, foi utilizado o parâmetro *numFolds=2*, o que significa que os dados foram divididos em duas partes: em cada iteração, uma parte era usada para o treinamento e a outra para validação, alternando-se os papéis.

O objeto *evaluator* foi utilizado para mensurar o desempenho do modelo em cada rodada, com base em métricas específicas de classificação binária. A combinação de hiperparâmetros definida anteriormente pelo *ParamGridBuilder* foi passada ao *CrossValidator* por meio do parâmetro *estimatorParamMaps*. Além disso, o parâmetro *parallelism=1* foi configurado para executar os testes de forma sequencial, respeitando os limites computacionais do ambiente utilizado.

A Figura 21 apresenta o resultado desse processo foi armazenado no objeto *cvModel*, que representa a melhor versão do modelo treinado com os dados otimizados. Essa abordagem contribuiu para mitigar o risco de *overfitting* e garantiu uma avaliação mais generalizável do desempenho do *XGBoost*.

```
crossval = CrossValidator(
    estimator=xgb,
    estimatorParamMaps=paramGrid,
    evaluator=evaluator,
    numFolds=2,
    parallelism=1
)

cvModel = crossval.fit(train)
```

Figure 21. Processo de Validação Cruzada com Otimização de Hiperparâmetros

Fonte: Elaborado pela autora.

Apesar do desempenho praticamente perfeito demonstrado pelas métricas, os resultados devem ser interpretados com cautela. Um desempenho tão elevado em um conjunto de dados de teste pode ser um indicativo de sobreajuste (*overfitting*), onde o modelo pode ter se ajustado excessivamente às particularidades dos dados de treinamento, incluindo o ruído. Conforme mencionado na seção de Limitações deste trabalho, a confirmação da robustez deste modelo exigiria testes adicionais, como a validação em diferentes conjuntos de dados e um treinamento com um número maior de rodadas (*num_round*), o que foi limitado por restrições computacionais. Ainda assim, dentro do escopo deste estudo, o XGBoost se estabeleceu como o algoritmo de maior potencial.”

Diante da análise individual de cada modelo e considerando suas respectivas limitações e pontos fortes, torna-se essencial realizar uma comparação direta entre os algoritmos testados. Essa comparação permite uma avaliação mais abrangente da performance de cada abordagem frente ao problema de detecção de fraudes em transações Pix, especialmente em cenários com forte desbalanceamento de classes. A seguir, serão apresentados e discutidos os resultados comparativos das principais métricas acurácia, precisão, *recall* e AUC com o objetivo de identificar, de forma fundamentada, qual modelo se mostrou mais eficaz e confiável para esse tipo de aplicação.

4.1.5. Comparação dos Algoritmos

A Tabela 6 apresenta o desempenho dos modelos de *Machine Learning* aplicados na detecção de fraudes. O modelo XGBoost se destacou de forma significativa, com um desempenho numericamente quase perfeito em todos os indicadores. A acurácia, AUC, *Recall*, Precisão e *F1-Score* foram todos de 99,97%.

Embora esses indicadores sejam excepcionais, eles devem ser vistos com cautela, pois um desempenho tão elevado pode indicar um sobreajuste (*overfitting*) do modelo aos dados de treino, um ponto que será discutido em maior detalhe na Seção 6. Esses resultados, portanto, refletem o alto potencial do XGBoost em cenários de desbalanceamento, mas sua capacidade de generalização real requer uma análise mais aprofundada.

O modelo XGBoost se destacou de forma significativa, apresentando desempenho praticamente perfeito em todos os indicadores. A acurácia, AUC, *Recall*, Precisão e *F1-Score* foram todos de 99,97%, demonstrando uma excelente capacidade de generalização, além de ser altamente eficaz tanto na identificação de fraudes quanto na redução de falsos positivos e falsos negativos. Esses resultados refletem a robustez do XGBoost, especialmente em problemas de classificação com dados desbalanceados, como é comum na detecção de fraudes.

O modelo *Random Forest* também apresentou um bom desempenho, com uma acurácia de 99,94%, AUC de 99,37%, precisão de 94,69%, *Recall* de 59,70% e *F1-Score* de 99,93%. Apesar de manter alta capacidade de acerto geral (acurácia) e excelente equilíbrio entre as classes (*F1-Score*), o *Recall* relativamente mais baixo (59,70%) indica que o modelo teve dificuldade em identificar todas as fraudes, o que é uma limitação considerável em cenários onde é preferível evitar falsos negativos.

O modelo de *Decision Tree*, embora apresente uma acurácia elevada (99.95%) e *F1-Score* de 99.94%, teve um desempenho inferior em termos de AUC (65.20%), além de um *Recall* de 64.93% e uma precisão de 90.78%. Isso sugere que, embora o modelo acerte muitos casos gerais, ele não é tão eficiente na separação correta entre transações fraudulentas e não fraudulentas, indicando possíveis problemas de *overfitting*.

Por outro lado, a Regressão Logística apresentou um desempenho muito abaixo do esperado para este problema. Apesar de uma acurácia de 99.86% e um AUC de 80.28%, o modelo teve tanto o *Recall* quanto a *Precisão* iguais a 0.00%, o que indica total incapacidade de identificar a classe de fraude. Esse resultado ocorre frequentemente em cenários com forte desbalanceamento de classes, onde o modelo tende a prever apenas a classe majoritária (não fraude). Embora o *F1-Score* registrado tenha sido 99.80%, esse valor reflete unicamente o desempenho na classe majoritária, não sendo representativo para a tarefa de detecção de fraudes. Esse comportamento demonstra que, sem estratégias de balanceamento de dados, a Regressão Logística não é adequada para este tipo de problema.

Table 6. Comparativo de desempenho dos modelos de *Machine Learning*

Modelo	Acurácia	AUC	Recall	Precisão	F1-Score
Decision Tree	99.95%	65.20%	64.93%	90.78%	99.94%
Random Forest	99.94%	99.37%	59.70%	94.69%	99.93%
Regressão Logística	99.86%	80.28%	0.00%	0.00%	99.80%
XGBoost	99.97%	99.97%	99.97%	99.97%	99.97%

Fonte: Elaborado pela autora.

De forma geral, os resultados indicam que, para este problema específico, o *XGBoost* se mostrou o modelo mais adequado, seguido pelo *Random Forest*. Modelos como *Decision Tree* e principalmente a Regressão Logística foram menos eficazes, especialmente quando priorizamos a detecção correta das fraudes.

Ao analisar o desempenho dos modelos, é crucial considerar o impacto de cada métrica no contexto de detecção de fraudes. Embora a acurácia do modelo de Regressão Logística tenha sido de 99,86% , essa métrica pode ser enganosa em cenários com dados desbalanceados. Uma alta acurácia pode simplesmente indicar que o modelo está prevendo corretamente a classe majoritária transações legítimas, enquanto falha em identificar a classe minoritária fraudes, que é o verdadeiro alvo de interesse. É por isso que o *Recall* (Revocação) se torna uma métrica de importância crítica. Um *Recall* alto indica que o modelo é capaz de identificar a maioria das fraudes reais, minimizando os falsos negativos. Um falso negativo, nesse cenário, representa uma fraude não detectada, resultando em prejuízo financeiro direto. Portanto, o desempenho quase perfeito de 99,97% em *Recall* do *XGBoost* o qualifica como uma solução muito mais robusta e confiável em comparação com o *Random Forest*, que obteve um *Recall* de 59,70% , ou a Regressão Logística, com *Recall* de 0,00%.

Contrastando nossos resultados com a literatura, este trabalho demonstra um avanço notável. Por exemplo, o estudo A1 [Wang et al. 2021] alcançou uma acurácia de 94% com um modelo DQN , um resultado significativo, porém inferior aos 99,97% obtidos pelo nosso modelo *XGBoost*. Da mesma forma, o trabalho A26, que se concentrou em

otimizar a Regressão Logística e atingiu 91% de acurácia, reforça nossas conclusões sobre as limitações desse algoritmo em seu estado puro, já que nosso modelo de Regressão Logística, sem o devido balanceamento, obteve um Recall nulo. Essa comparação evidencia que os modelos de ensemble, como o XGBoost, não apenas superam os modelos clássicos, mas também as abordagens mais complexas como o aprendizado por reforço no contexto específico deste dataset sintético, oferecendo uma solução mais precisa para a detecção de fraudes no Pix. Apesar dos resultados promissores, é fundamental reconhecer as limitações que permearam este estudo, conforme detalhado a seguir.

5. Limitações do Trabalho

Apesar dos resultados positivos alcançados, este estudo apresenta algumas limitações que precisam ser consideradas. A primeira delas está relacionada ao uso de uma base de dados sintética. Por motivos de privacidade e segurança, não foi possível trabalhar com dados reais de transações financeiras, então optou-se por uma base que simula essas operações. Embora essa base seja bastante usada em pesquisas, ela não representa totalmente a realidade dos sistemas bancários, o que pode influenciar os resultados quando aplicados fora do ambiente de testes.

Outra limitação foi o uso exclusivo de técnicas chamadas "supervisionadas", ou seja, modelos que aprendem com exemplos já classificados como fraude ou não. Esse tipo de abordagem é eficaz, mas deixa de fora outros métodos que poderiam ajudar a identificar fraudes novas ou desconhecidas, como os modelos que detectam comportamentos fora do padrão sem depender de rótulos prévios.

Além disso, nem todos os algoritmos inicialmente previstos puderam ser treinados. Modelos como o *K-Nearest Neighbors* (KNN) e o *Support Vector Machine* (SVM) foram descartados na etapa de experimentação devido a limitações computacionais, especialmente relacionadas ao consumo de memória. Como o conjunto de dados utilizado possui mais de seis milhões de registros, esses algoritmos, que são naturalmente mais custosos em termos de processamento, tornaram-se inviáveis no ambiente disponível para os testes.

Também é importante destacar que não foi feita uma análise mais profunda sobre os erros dos modelos. Por exemplo, em alguns casos, uma transação legítima pode ser identificada como fraude (falso positivo), o que pode causar transtornos para o usuário. Em outros, o sistema pode deixar passar uma fraude real (falso negativo), o que representa prejuízo. Esses dois tipos de erro têm impactos diferentes e deveriam ser considerados com mais atenção.

Além disso, um dos modelos utilizados (o *XGBoost*) teve um desempenho quase perfeito, o que, apesar de parecer ótimo, pode indicar que ele aprendeu demais com os dados de treinamento e talvez não funcione tão bem com dados novos. Esse modelo foi treinado com apenas 20 `num_round` devido à limitação de memória para processar o grande volume de dados. Esse fenômeno, conhecido como "sobreajuste", exige mais testes com maiores quantidades de `num_round` para que a confiabilidade dos resultados seja validada.

Por fim, embora o estudo tenha apresentado os modelos e seus resultados, não chegou a desenvolver uma aplicação prática, como um sistema que possa ser utilizado de fato por uma instituição financeira. Isso limita a visão de como essas soluções poderiam ser aplicadas no dia a dia.

6. Conclusão

Este trabalho teve como foco o desafio crescente das fraudes em transações via Pix. A proposta foi avaliar como diferentes algoritmos de aprendizado de máquina supervisionado se saem na tarefa de identificar atividades fraudulentas. A ideia central era entender, com base em uma análise comparativa, qual modelo seria mais eficiente nesse tipo de detecção contribuindo, assim, para o desenvolvimento de sistemas de segurança mais eficazes.

Os resultados obtidos mostraram claramente o desempenho de cada modelo. O *XGBoost* se destacou com um resultado impressionante: 99,97% em todas as principais métricas avaliadas, como Acurácia, Precisão, *Recall* e *F1-Score*. O *Random Forest* também apresentou bons resultados, com acurácia de 99,94%, mas teve um desempenho inferior no *Recall* (59,70%), o que indica uma menor capacidade de encontrar todas as fraudes. A Árvore de Decisão teve sinais de sobreajuste, enquanto a Regressão Logística falhou completamente, sem detectar nenhuma transação fraudulenta, provavelmente devido a falta de memória para promover o balanceamento esse modelo deixou a desejar devido ao desbalanceamento entre fraudes e transações legítimas.

Com base nesses achados, é possível afirmar que modelos mais avançados, como o *XGBoost*, são muito mais eficazes do que abordagens simples para lidar com esse tipo de problema. O estudo atinge seus objetivos ao indicar o algoritmo mais promissor e ao mostrar com clareza as limitações dos demais. A principal contribuição aqui é oferecer uma base concreta para quem deseja aplicar inteligência artificial no combate a fraudes financeiras.

Ainda assim, é importante destacar algumas limitações. O conjunto de dados utilizado é sintético, ou seja, simula transações reais, mas pode não representar com total fidelidade o comportamento do mundo real. Além disso, o ótimo desempenho do *XGBoost* pode indicar um possível sobreajuste, o que exigiria mais testes para confirmar. Também vale dizer que o estudo focou apenas em métodos supervisionados, deixando de lado outras abordagens que poderiam ser exploradas.

Para trabalhos futuros relevantes, recomenda-se a aplicação desses modelos em bases de dados reais provenientes de instituições financeiras, o que permitiria validar os resultados obtidos em cenários mais próximos da realidade operacional. Além disso, a investigação de técnicas de aprendizado não supervisionado pode se mostrar promissora, uma vez que essas abordagens identificam padrões atípicos sem a necessidade de dados rotulados, o que é especialmente útil em contextos onde a rotulagem manual é limitada ou custosa. Outra vertente relevante seria a análise aprofundada dos erros de classificação especialmente os falsos positivos e falsos negativos, pois tais falhas podem gerar impactos distintos no negócio. Enquanto os falsos negativos podem resultar em perdas financeiras diretas ao não identificar transações fraudulentas, os falsos positivos tendem a comprometer a experiência do usuário ao bloquear operações legítimas. Assim, uma análise de custo-benefício desses erros pode guiar decisões mais estratégicas na aplicação prática dos modelos.

Por fim, recomenda-se, como continuação deste estudo, o desenvolvimento de um protótipo funcional baseado no modelo com melhor desempenho, visando sua possível integração em sistemas reais de detecção de fraudes com Instituições Financeiras.

References

- Al-dahasi, E. M., Alsheikh, R. K., Khan, F. A., and Jeon, G. (2025a). Optimizing fraud detection in financial transactions with machine learning and imbalance mitigation. *Expert Systems*, 42(2):e13682.
- Al-dahasi, E. M., Alsheikh, R. K., Khan, F. A., and Jeon, G. (2025b). Optimizing fraud detection in financial transactions with machine learning and imbalance mitigation. *Expert Systems*, 42(2):e13682.
- Alshameri, F. and Xia, R. (2023). Credit card fraud detection: an evaluation of smote re-sampling and machine learning model performance. *International Journal of Business Intelligence and Data Mining*. Online April 18, 2023.
- Banco Central do Brasil (2020). Criação do pix.
- Banco Central do Brasil (2023a). Pix: Mecanismo especial de devolução (med). <https://www.bcb.gov.br/estabilidadefinanceira/pixmed>. Acesso em: 11 jun. 2025.
- Banco Central do Brasil (2023b). Relatório anual de estabilidade financeira 2023. <https://www.bcb.gov.br/relatorio-estabilidade-financeira>.
- Banco Central do Brasil (2024a). Pix: movimentações recorde de R\$ 17,2 trilhões em 2023. <https://www.bcb.gov.br/detalhenoticia/803/noticia>. Acesso em: 10 jun. 2025.
- Banco Central do Brasil (2024b). Sistema de pagamentos instantâneos (spi) e estatísticas do pix. <https://www.bcb.gov.br/estabilidadefinanceira/estatisticaspix>. Acesso em: 10 jun. 2025.
- Cartaxo, B., Pinto, G., and Soares, S. (2018). The role of rapid reviews in supporting decision-making in software engineering practice. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pages 24–34.
- Chang, V., Di Stefano, A., Sun, Z., Fortino, G., et al. (2022). Digital payment fraud detection methods in digital ages and industry 4.0. *Computers and Electrical Engineering*, 100:107734.
- Dal Pozzolo, A. (2015). Synthetic financial datasets for fraud detection. Disponível em: Kaggle. Acesso em: junho de 2025.
- Delgolla, M., Halloluwa, T., and Rathnayake, A. (2021). A rule based approach to minimize false-positive declines in electronic card not present financial transactions using feature engineering techniques. In *2021 21st International Conference on Advances in ICT for Emerging Regions (ICter)*, pages 99–104. IEEE.
- FEBRABAN (2023). Relatório de segurança bancária 2023. Acesso em: 10 jun. 2025.
- Feng, Y., Zhang, L., and Liu, X. (2023). A comparative study of machine learning algorithms for credit card fraud detection. *Expert Systems with Applications*, 213:119151.

- Haby, M. M., Chapman, E., Clark, R., Barreto, J., Reveiz, L., and Lavis, J. N. (2016). What are the best methodologies for rapid reviews of the research evidence for evidence-informed decision making in health policy and practice: a rapid review. *Health research policy and systems*, 14(1):1–12.
- Kant, V. (2024). Optimizing logistic regression for flawless fraud detection in digital payments. In *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, pages 97–100. IEEE.
- Krotkiewicz, M., Nowak, M., and Wawrzyniak, P. (2022). Fraud detection in financial transactions using machine learning: A comprehensive survey. *IEEE Access*, 10:107064–107092.
- Martins, E. and Galegale, N. V. (2022). Detecção de fraudes no segmento de crédito financeiro utilizando aprendizado de máquina: uma revisão da literatura. *Revista e-TECH: Tecnologias para Competitividade Industrial-ISSN-1983-1838*, 15(3).
- Mattos, L. (2022). Aplicação de técnicas de machine learning no apoio à detecção de fraudes em pagamentos online. Orientadora: Milena Cristina Franca, Coorientador: Paulo Roberto Cordova.
- SILVA, G. F. D. S., LAMARCA, D. S. F., and SARRIÉS, G. A. (2022). Machine learning e fraude de cartão de crédito: Uma revisão bibliográfica sistemática. *Revista Intellectus*, 66(1):3–19.
- Souza and Bordin (2023). Detecção de fraude de cartão de crédito por meio de algoritmos de aprendizado de máquina. *Revista Brasileira de Computação Aplicada*, 15(1):1–11.
- Souza, M. R. d. (2024). Aplicação de aprendizado federado na detecção e prevenção de fraudes em transações financeiras.
- Tricco, A. C., Langlois, E. V., and Straus, S. E. (2015). Rapid reviews to strengthen health policy and systems: a practical guide.
- Tudisco, A., Volpe, D., Ranieri, G., Curato, G., Ricossa, D., Graziano, M., and Corbellotto, D. (2024). Evaluating the computational advantages of the variational quantum circuit model in financial fraud detection. *IEEE Access*.
- Wang, X., Wan, Z., and Zhang, Y. (2021). A dqn-based internet financial fraud transaction detection method. In *Proceedings of the 5th International Conference on Computer Science and Application Engineering*, pages 1–5.
- Yao, Y., Li, G., Liu, Z., and Liu, J. (2023). A review of imbalanced learning in financial fraud detection: a versatile framework and a future-proof roadmap. *Applied Intelligence*, 53(16):19747–19772.
- Zhang, H., Li, W., and Zhao, F. (2022). Performance metrics for imbalanced classification: A comprehensive review. *Information Fusion*, 81:115–135.