UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de big data para evitar paradas e prejuízos no pagamento instantâneo Pix

Dênio Albaro de Lima Rodrigues

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP
SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de big data para evitar paradas e prejuízos no pagamento instantâneo Pix

Dênio Albaro de Lima Rodrigues

USP - São Carlos

2025

Dênio Albaro de Lima Rodrigues

Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de big data para evitar paradas e prejuízos no pagamento instantâneo Pix

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Me. Willian Dener de Oliveira

USP - São Carlos

2025

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi e
Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Rodrigues, Dênio Albaro de Lima
R696a Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de big data para evitar paradas e prejuízos no pagamento instantâneo Pix / Dênio Albaro de Lima Rodrigues; orientador Willian Dener de Oliveira. -- São Carlos, 2025.
78 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025.

1. Sicoob. 2. Pix. 3. Resiliência Cibernética. 4. Big data. 5. Inteligência Artificial. I. Oliveira, Willian Dener de, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

DEDICATÓRIA

Dedico essa obra para minha família e amigos, ao Sicoob, aos professores, tutores e orientadores que sempre me apoiaram na realização desse MBA nessa conceituada instituição de ensino.

A minha esposa Jackie pela compreensão, amor, carinho e apoio incondicional.

AGRADECIMENTOS

À Deus pela benção da vida. À minha família (esposa Jackie, filhos Ayla e Caio, genro Sávio e neto Noah) pelo apoio, carinho, parceria e pela constante torcida para que eu sempre alcance meus objetivos pessoais e profissionais.

À Dona Maria Ângela de Lima de Lima Rodrigues, minha grande e inspiradora professora no 2º ano primário do grupo escolar de Inhaúma-MG. Nas horas vagas ela era também esposa do meu pai Sebastião Rodrigues da Cunha (Bill), minha mãe e dos meus 7 irmãos: Denise, Vanderlei, Ângelo, Aroldo, Flávio, Rômulo e Guto.

Ao Sicoob, organização na qual desenvolvo minha carreira profissional há mais de 30 anos, compartilhando os mesmos valores e propósito, pelo patrocínio e incentivo em mais essa grande conquista.

À Profa. Dra. Solange Rezende pela gestão e organização do curso e pelo apoio incondicional aos alunos do MBA nessa conceituada instituição de ensino USP. Aos professores, tutores e, em especial, ao meu orientador Prof. Me. Willian Dener de Oliveira pelos ensinamentos, conselhos, esclarecimento de dúvidas e, acima de tudo, pelo profissionalismo e presteza no atendimento de minhas solicitações.

Aos amigos do trabalho Vilaça e Edson por temos juntos aceitado o desafio de fazer esse curso e sempre encorajando e motivando uns aos outros a seguir em frente e aproveitar ao máximo os ensinamentos e conhecimentos adquiridos.

Aos colegas de trabalho Rudge Moino, Franciscon Santos e demais profissionais do time técnico do Sicoob por terem tido papel relevante na montagem e liberação de infraestrutura tecnológica para realização desse TCC, envolvendo ambiente de estudos, desenvolvimento, experimentação e testes, bem como debates sobre as soluções implementadas.

RESUMO

RODRIGUES, D. A. L. **Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de Big Data para evitar paradas e prejuízos no pagamento instantâneo Pix.** 2025. Trabalho de Conclusão de Curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

O pagamento instantâneo brasileiro, também conhecido como Pix, é o meio de pagamento criado pelo Banco Central do Brasil em que os recursos são transferidos entre contas em poucos segundos a qualquer momento. Lançado oficialmente em novembro de 2020, o Pix se tornou o maior fenômeno de utilização de um produto/serviço do mercado financeiro nacional em todos os tempos. Nesse cenário, o maior desafio das instituições financeiras é manter um ambiente tecnológico para suportar a solução Pix, com volume gigantesco de transações funcionando 24h por dia, com elevados níveis de resiliência cibernética em seus 4 pilares (disponibilidade, desempenho, qualidade e segurança), pois a interrupção dessa solução acarreta prejuízos financeiros enormes. O objetivo desse trabalho foi utilizar técnicas de Big Data e Inteligência Artificial para analisar as informações contidas nos milhões de logs coletados nas transações financeiras realizadas com Pix e armazenados em ambiente de big data do Sistema de Cooperativas de Crédito do Brasil – Sicoob, com foco na geração de percepções, investigação e compreensão desses dados, visando criar um modelo que auxilie na identificação de padrões e comportamentos para detectação e classificação de anomalias do ambiente tecnológico, de forma a torná-lo cada vez mais resiliente, evitando falhas na estrutura que acarretam parada total ou parcial dos serviços Pix. Os resultados evidenciam que aplicar algoritmos não supervisionados de Aprendizado de Máquina, em fluxos de Big Data contendo logs transacionais em tempo real, auxilia na detecção de anomalias e na resiliência cibernética em um produto financeiro crítico como o Pix.

Palavras-chave: Sicoob; Pix; resiliência cibernética; big data; inteligência artificial.

ABSTRACT

RODRIGUES, D. A. L. **Análise de resiliência cibernética utilizando técnicas de inteligência artificial em ambiente de Big Data para evitar paradas e prejuízos no pagamento instantâneo Pix.** 2025. Trabalho Final de Curso – Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Brazilian instant payments, also known as Pix, are a payment method created by the Central Bank of Brazil that allows funds to be transferred between accounts in seconds at any time. Officially launched in November 2020, Pix has become the most popular product/service in the Brazilian financial market ever. In this scenario, the biggest challenge for financial institutions is maintaining a technological environment to support the Pix solution, with a massive volume of transactions operating 24/7, and with high levels of cyber resilience across its four pillars (availability, performance, quality, and security), as any interruption of this solution can result in significant financial losses. The objective of this work was to use Big Data and Artificial Intelligence techniques to analyze the information contained in the millions of logs collected from financial transactions carried out with Pix and stored in the big data environment of the Brazilian Credit Union System (Sicoob). The focus was on generating insights, investigating, and understanding this data. The goal was to create a model that helps identify patterns and behaviors for detecting and classifying anomalies in the technological environment, thereby making it increasingly resilient and avoiding structural failures that could lead to total or partial interruptions to Pix services. The results demonstrate that applying unsupervised Machine Learning algorithms to Big Data streams containing real-time transactional logs helps detect anomalies and improve cyber resilience in a critical financial product like Pix.

Keywords: Sicoob; Pix; cyber resilience; big data; artificial intelligence.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 Introdução

1.1 Contextualização

O pagamento instantâneo brasileiro, também conhecido como Pix, é o meio de pagamento criado pelo Banco Central (BC) em que os recursos são transferidos entre contas em poucos segundos a qualquer momento. É prático, rápido e seguro. O Pix pode ser realizado a partir de uma conta corrente, conta poupança ou conta de pagamento pré-paga (BANCO CENTRAL DO BRASIL, 2022).

Além de aumentar a velocidade em que pagamentos ou transferências são feitos e recebidos, o Pix tem o potencial de alavancar a competitividade e a eficiência do mercado; baixar o custo e promover a inclusão financeira da população (BANCO CENTRAL DO BRASIL, 2022).

Lançado oficialmente em novembro de 2020, o Pix se tornou o maior fenômeno de utilização de um produto/serviço do mercado financeiro nacional em todos os tempos. Em quatro anos, essa solução transformou o mercado de meios de pagamento. De seu início até 30 de setembro de 2024, foram 121,5 bilhões de transações feitas no sistema financeiro nacional, com valores transacionados atingindo R$ 52,6 trilhões. Em novembro de 2024, o Pix assumiu a dianteira entre as marcas mais fortes do Brasil (FORBES, 2024).

Nesse contexto, o maior desafio das instituições financeiras é manter um ambiente tecnológico para suportar a solução, com volume gigantesco de transações funcionando 24h por dia, com elevados níveis de resiliência cibernética em seus 4 pilares: disponibilidade, desempenho, qualidade e segurança. Além de manter a resiliência do ambiente atual, é imprescindível adequar o planejamento de capacidade para que esse ambiente propicie o crescimento expressivo das transações e a concentração desse movimento em dias e horários específicos.

1.2 Questão de pesquisa e objetivos

Esse trabalho se justifica pois, na maioria das vezes, as instituições financeiras não suportam esse crescimento contínuo e nem mesmo picos diários de movimento, causando a queda do Pix e, consequentemente, acarretando prejuízos financeiros enormes para a instituição e prestando um mau serviço para toda a população (ESTADÃO, 2025).

Durante as pesquisas, foram encontrados trabalhos correlatos referentes a gestão de infraestrutura tecnológica para tratar falhas em ativos de tecnologia da informação (TI) e evitar quedas de sistemas decorrentes de acessos ou movimentos atípicos (MEHTA, KOTHURI, GARCIA, 2018).

Como na era atual das Indústrias de TI, o gerenciamento de infraestrutura de TI desempenha um papel crucial. Como resultado, pesquisas substanciais estão sendo realizadas para melhorar a confiabilidade e a disponibilidade de ativos em infraestrutura de TI. (BHANAGE; PAWAR, 2020). A identificação e o tratamento de falhas são cruciais e desafiadores devido à complexidade da infraestrutura de TI. Os logs do sistema são a principal fonte de informações para diagnosticar e corrigir falhas. (BHANAGE; PAWAR, VISHAL, KOTECHA, 2021).

Os logs do sistema registram todos os detalhes da operação executada e fornecem muitas informações dimensionais sobre ela. Os logs do sistema documentam a causa do problema dos componentes da infraestrutura de TI. O log do sistema é o primeiro lugar onde o administrador do sistema investiga problemas em alertas de falha. Consequentemente, os logs do sistema têm sido amplamente usados na detecção ou previsão de anomalias e falhas devido à sua franqueza e utilidade Sistemas de computador complexos registram uma coleção massiva de logs que podem disponibilizar informações úteis; por outro lado, analisar dados colossais é desafiador (BHANAGE; PAWAR; KOTECHA; ABRAHAM, 2023).

O objetivo desse trabalho é utilizar técnicas de Big Data e Inteligência Artificial (IA) para analisar as informações contidas nos milhões de logs coletados nas transações financeiras realizadas com Pix e armazenados em ambiente de big data do Sicoob, com foco na geração de percepções, investigação e compreensão desses dados, visando criar um modelo que auxilie na identificação de padrões e comportamentos para detectação e classificação de anomalias do ambiente tecnológico, de forma a torná-lo cada vez mais resiliente, evitando falhas na estrutura que acarretam parada total ou parcial dos serviços Pix.

1.3 Organização do trabalho

O capítulo 2 contém os conceitos para este trabalho e apresenta uma revisão dos trabalhos que foram estudados sobre o tema. O capítulo 3 traz a metodologia adotada, descrevendo as etapas utilizadas no fluxo e coleta, armazenamento, processamento dos logs do Pix em big data, montando um conjunto de dados (*dataset) c*om dados básicos que possam ser

avaliados por um modelo baseado em técnicas e ferramentas de IA, com treinamento e algoritmos com parâmetros definidos para detecção de anomalias e evitar quedas no serviço Pix. O capítulo 4 descreve os experimentos, análises, a relação com o objetivo proposto e resultados obtidos por meio de gráficos, tabelas e imagens. O capítulo 5 faz uma conexão com a introdução, apresentado a conclusão dos resultados, contribuições, inovações científicas da pesquisa, limitações e os trabalhos futuros.

2 Fundamentação teórica

2.1 Cooperativismo financeiro

O movimento do cooperativismo financeiro é uma iniciativa que promove a união de pessoas em torno de um objetivo comum: oferecer acesso a serviços financeiros de forma democrática, inclusiva e sustentável. Esse movimento surge na segunda metade do século XIX, em Rochdale, na Inglaterra e seus direcionadores filosóficos-doutrinários vem se propagando por valores e princípios de adoção universal (MEINEN; PORT, 2021). São seus valores basilares: solidariedade; liberdade; democracia; equidade; igualdade; responsabilidade; honestidade; transparência; e responsabilidade socioambiental; e seus princípios: adesão livre e voluntária; gestão democrática; participação econômica; autonomia e independência; educação, formação e informação; intercooperação; e interesse pela comunidade.

Por meio das cooperativas financeiras, o cidadão tem a oportunidade de obter atendimento personalizado para suas necessidades. O resultado positivo da cooperativa é conhecido como sobra e é repartido entre os cooperados em proporção com as operações que cada associado realiza com a cooperativa. Assim, os ganhos voltam para a comunidade dos cooperados. No entanto, assim como partilha das sobras, o cooperado está sujeito a participar do rateio de eventuais perdas, em ambos os casos na proporção dos serviços usufruídos.

Esse modelo tem se mostrado uma alternativa sólida às instituições financeiras tradicionais, especialmente em regiões onde o acesso a serviços bancários é limitado, contribuindo para o desenvolvimento econômico local e a inclusão financeira.

Estudo realizado pela Organização das Cooperativas Brasileiras (OCB) evidencia os benefícios significativos do cooperativismo de crédito no fortalecimento econômico e social das regiões onde está presente. Além de fomentar o empreendedorismo, gerar empregos e ampliar o acesso ao crédito, o cooperativismo impulsiona setores como agricultura, comércio e serviços, fortalecendo a economia local. Segundo a pesquisa, para cada R$ 1 concedido em crédito pelas cooperativas, há um efeito multiplicador de R$ 2,56 na atividade econômica da região, confirmando o papel estratégico dessas instituições na promoção do desenvolvimento sustentável e na redução das desigualdades regionais (ORGANIZAÇÃO DAS COOPERATIVAS BRASILEIRAS, 2024).

2.2 O Sistema de Cooperativas de Crédito do Brasil – Sicoob

O Sicoob é um sistema de cooperativas financeiras formado por 323 cooperativas singulares, 14 cooperativas centrais, 1 confederação e 9,1 milhões de cooperados, presente em 2.452 municípios com 4.685 pontos de atendimento, sendo a maior rede de atendimento do Sistema Financeiro Nacional (SFN), onde em 415 munícipios é a única instituição financeira presente para a população. É reconhecida como a 3ª melhor instituição financeira do Brasil segundo ranking "Melhores Bancos do Mundo 2023 e 2024" da Forbes (SICOOB, 2025).

O Sicoob é reconhecido por seu investimento estratégico em tecnologia e inovação. Pilares que sustentam seu crescimento e promovem a eficiência operacional. No centro da sua estratégia está a Plataforma de Serviços Financeiros do Sicoob conhecida por Sisbr, uma plataforma tecnológica moderna, segura e completa, que sustenta todas as operações financeiras, gerenciais, contábeis e administrativas das cooperativas. Entre os principais destaques está seu aplicativo financeiro móvel denominado SuperApp Sicoob, uma solução completa com mais de 300 transações disponíveis, sendo reconhecido como um dos aplicativos financeiros mais bem avaliados nas lojas virtuais.

A Plataforma de Serviços Financeiros do Sicoob – Sisbr é mais que um sistema de gestão e trata-se de um ERP cooperativo robusto e completo, uma das maiores iniciativas do mundo nesse formato. Ele se estrutura em seis plataformas integradas, como ilustrado na Figura 1 – Plataforma de Serviços Financeiros do Sicoob – Sisbr. A Plataforma de Negócios reúne produtos e serviços financeiros, como crédito, conta corrente, poupança, investimentos, seguros, previdência e consórcios. A Plataforma de Canais de Atendimento oferece atendimento completo por meios digitais e presenciais como terminais de caixa, caixas eletrônicos, *internet* e *mobile banking*. A Plataforma de Pagamentos e Recebimentos garante eficiência e segurança na realização de pagamentos com cartão de crédito, débito, voucher, Pix, boletos de cobrança, TED e cheques, além de maquininha de adquirência para pessoas jurídicas. A Plataforma de Gestão Empresarial sustenta os principais processos administrativos, operacionais e contábeis das cooperativas financeiras. A Plataforma de Risco e *Compliance* assegura a conformidade e a proteção do sistema abarcando soluções para a gestão do risco operacional, risco de mercado, risco de crédito, risco de liquidez e controles internos. A Plataforma Analítica potencializa as decisões estratégicas por meio de inteligência baseada em dados.

Figura 1 – Plataforma de Serviços Financeiros do Sicoob – Sisbr

Fonte: SICOOB (2024).

A estrutura tecnológica que sustenta a operação do Sistema Sicoob e propicia o crescimento dos seus negócios está instalada em 2 datacenters localizados em Brasília – DF, além das soluções em nuvem pública. O processamento dos dados é distribuído nesses 2 datacenters, que funcionam de forma síncrona com gravação de dados nos 2 sítios.

A infraestrutura de TI é composta de 3 grandes plataformas: Mainframe com sistema operacional Linux e banco de dados DB2, Intel com sistema operacional Windows e banco de dados SQL *server*, Hexadata com sistema operacional Windows e banco de dados Oracle. São mais de 4.000 servidores virtualizados nos 2 datacenters e em nuvem. São utilizados equipamentos de última geração e a implementação de tecnologias de automação, garantindo mais eficiência e escalabilidade para atender às demandas crescentes.

Em 2024 o Sicoob processou mais de 41 bilhões de transações de negócios por meio do Sisbr, contabilizando aquelas realizadas nos canais de atendimento, na retaguarda (*backoffice*), na emissão (cartões) e na operação de adquirência (Sipag 2.0). Os canais digitais foram responsáveis por 90% das transações financeiras realizadas, sendo 80% desse montante feitas pelos cooperados no SuperApp, atualmente o principal canal transacional do cooperado. Em volume financeiro, foram transacionados mais de R$ 3,27 trilhões nos canais e meios de pagamentos/recebimentos.

A produção, operação e monitoração tecnológica funcionam em regime 24x7 para garantir a resiliência cibernética do ambiente tecnológico. A gestão dos serviços de TI utiliza abordagem ITIL (*Information Technology Infrastructure Library*) para controlar e aumentar a eficiência dos processos tecnológicos.

2.3 Pix

O Pix é um meio digital distinto e "híbrido" de pagamento que combina as características funcionais das plataformas de pagamentos online com as propriedades jurídico-políticas de uma CBDC. Em todo caso, a sua criação expressa o intuito do Banco Central do Brasil em assegurar o seu controle político sobre o processo de digitalização dos meios de pagamento no Brasil (KOSINSKY, 2021).

A grande diferença para outros meios de pagamento é a rapidez e a disponibilidade: enquanto existem restrições de dias e horários para enviar quantias através de TED e para realizar pagamentos de contas por boletos, o Pix permite transações 24 horas por dia, sete dias por semana, incluindo até feriados nacionais. Em outras palavras, o dinheiro cai na hora, a qualquer dia da semana ou horário (NUBANK, 2025).

O meio de pagamento criado pelo Banco Central (BC) em que os recursos são transferidos entre contas em poucos segundos, a qualquer hora ou dia. É prático, rápido e seguro. O Pix pode ser realizado a partir de uma conta corrente, conta poupança ou conta de pagamento pré-paga (BANCO CENTRAL DO BRASIL, 2025).

O Pix funciona sem intermediação de terceiros: o dinheiro sai de uma conta e vai diretamente para a conta de quem vai receber os valores. Esse é o mesmo sistema usado nas transferências entre contas de um mesmo banco, que são instantâneas, podem ser feitas a qualquer momento e são gratuitas. O Pix funciona por meio do Sistema de Pagamentos Instantâneos (SPI). Ele é gerido e operado pelo Banco Central, por meio do Departamento de Operações Bancárias e de Sistema de Pagamentos (Deban). O SPI é o responsável pelo processamento das transações e por conectar todos os participantes do sistema, incluindo bancos, fintechs e outras instituições financeiras, permitindo a transferência de dinheiro em tempo real (NUBANK, 2025).

Qualquer transação de pagamento, de qualquer valor, pode ser feita usando o Pix. A única condição é que o recebedor aceite este meio de pagamento. É possível fazer e receber um Pix em diversas situações: transferências entre pessoas; pagamento em estabelecimentos comerciais, incluindo lojas físicas e comércio eletrônico; pagamento de prestadores de serviços;

pagamento entre empresas, como pagamentos de fornecedores, por exemplo; recolhimento de receitas de órgãos públicos federais como taxas (custas judiciais, emissão de passaporte etc), aluguéis de imóveis públicos, serviços administrativos e educacionais, multas, entre outros (esses recolhimentos poderão ser feitos por meio do PagTesouro); pagamento de cobranças; pagamento de faturas de serviços públicos, como energia elétrica, telecomunicações (telefone celular, internet, TV a cabo, telefone fixo) e abastecimento de água; e recolhimento de contribuições do FGTS e da Contribuição Social (BANCO CENTRAL DO BRASIL, 2025).

O Pix deve crescer 35% ao ano até 2027 e se consolidar como o método de pagamento de maior expansão no e-commerce de mercados emergentes (ÉPOCA NEGÓCIOS, 2025). De acordo com estatísticas do Banco Central (BANCO CENTRAL DO BRASIL, 2025), o volume de transações continua a ter um crescimento expressivo e deverá crescer ainda mais.

Em virtude da facilidade de operação do produto e da bancarização da população brasileira, o Pix se tornará cada vez mais o principal serviço do sistema financeiro do país, forçando a infraestrutura tecnológica das instituições financeiras a se tornarem cada vez mais resilientes e ininterruptas durante 24 horas e 7 dias por semana. Os valores financeiros aumentarão com o aumento do volume transacional e, como consequência, a queda dos serviços de Pix acarretará prejuízos financeiros e de imagem ainda maiores para instituição financeira.

2.4 Resiliência Cibernética

A resiliência cibernética ou digital é a capacidade das organizações de aceitar, sobrepor-se, recuperar-se e superar-se (evoluir) diante dos riscos e ataques cibernéticos, exigindo, portanto, que elas sejam capazes de manter suas operações, mudarem e recuperarem-se rapidamente ante a qualquer adversidade que atente contra suas operações e as respectivas tecnologias da informação que as suportam (SILVA, 2023).

Ser resiliente em termos cibernéticos pode ser definido, portanto, com a capacidade organizacional de identificar e gerenciar os riscos e eventos adversos de origem cibernética. Esta capacidade é resultante da combinação de processos, práticas, mentalidades e tecnologias que envolvem toda a organização de forma integrada e tem por objetivo garantir que esta previna, responda e se recupere de ataques cibernéticos, com foco no menor comprometimento possível em termos operacionais, financeiros, de relacionamento, de reputação, de direitos de terceiros, dentre outros (SILVA, 2023).

Nesta era digital, as informações são continuamente adquiridas e processadas em todos os lugares, de muitas fontes e para muitos propósitos e setores. Nesse sentido, o Aprendizado

de Máquina (ML) e a IA estão desempenhando um papel decisivo na conversão de informações brutas em previsões e recomendações úteis para melhorar a resiliência das operações de negócio e a vida geral dos cidadãos (MARTIN; LANGENDOERFER; ZARRIN; DÍAZ; BARTOLOMÉ, 2022).

A evolução da Internet das Coisas (IoT), resultando no aumento de informações físicas monitoradas, também promoveu o crescimento de outras áreas que são diretamente dependentes das informações disponíveis, como inteligência artificial e aprendizado de máquina. O maior desafio das empresas é construir processos e ferramentas mais eficazes, não só para gerenciar a informação, mas também para facilitar a transformação dessa informação em conhecimento (BRAGA; GOMES, 2017).

A identificação de eventos raros é um componente importante monitoramento de sistema e detecção de eventos. Anomalias podem representar situações problemáticas em que percepções iniciais, precisos e acionáveis são essenciais para fazer avaliações situacionais em caso de condições inesperadas. Para muitos problemas, o estado da arte em aprendizado de máquina é o aprendizado em lote. No entanto, treinar modelos em um contexto de lote pode representar um custo de oportunidade. Por exemplo, em aplicações como detecção de fraude, o tempo e o investimento gastos no treinamento de modelos em lote, com conjuntos de dados cada vez maiores e infraestrutura complexa, podem ser gastos na detecção de comportamento aberrante. Algoritmos de detecção de anomalias *online* oferecem acesso rápido a percepções úteis com menos requisitos de capacidade de computação e geralmente precisam ser integrados a fluxos de dados existentes ou legados na empresa (BELACEL; RICHARD; RANGAVAJJALA; ADHADUK, 2021).

O monitoramento de infraestrutura de TI tornou-se uma tarefa desafiadora devido ao aumento das complexidades infraestrutura e sua utilização. Qualquer falha por um pequeno número de tempo leva a perdas significativas para uma organização. Assim, é essencial evitar tais condições de falha (BHANAGE; PAWAR; VISHAL; KOTECHA, 2021).

O tempo de inatividade de um ambiente tecnológico empresarial agora é mais caro do que nunca: cerca de 44% das empresas indicam que os custos de tempo de inatividade por hora excedem US$ 1 milhão a mais de US$ 5 milhões, excluindo quaisquer taxas legais, multas ou penalidades. Além disso, 91% das organizações disseram que uma única hora de tempo de inatividade que coloca o hardware e os aplicativos de servidor de missão crítica offline, em média, ultrapassa US$ 300.000,00 devido à perda de negócios, interrupções de produtividade e esforços de remediação (TECHCHANNEL, 2021).

No Sicoob, os principais indicadores de resiliência cibernética, apurados durante os 360 dias de 2024, superaram as metas estabelecidas no acordo de níveis de serviço (SLAs) com as cooperativas. O indicador de disponibilidade alcançou a marca de 99,824% (meta: 99,20%), o desempenho atingiu 98,949% (meta: 96,00%) e a qualidade chegou a 99,885% (meta: 99,30%).

Mesmo tendo alcançado as metas, o percentual de inatividade dos serviços consolidados no SLA, causa grandes prejuízos financeiros e de imagem para o Sicoob. Como exemplo, se considerarmos que em 1 ano com 360 dias, o percentual de 0,176% de inatividade corresponde a 15,21 horas. Se são transacionados R$ 3,27 trilhões nos canais digitais do Sisbr no ano, uma projeção de valor financeiro perdido nessa indisponibilidade é de R$ 5,76 bilhões.

Diante desse cenário, o Conselho de Administração da instituição solicitou a imediata ampliação contínua dos percentuais de indicadores de resiliência e um plano de ações para que os indicadores estejam cada vez próximos de 100,00%, o que evitará perdas financeiras, de imagem e não prejudicará a população de modo geral. Este é um dos grandes desafios para área de tecnologia do Sicoob: coletar milhões de logs das transações financeiras de forma a identificar anomalias, causas de subidas abruptas de movimento que causam degradação do ambiente tecnológico, com um objetivo de criar um modelo que utilize recursos de big data e inteligência artificial para gerar alertas sobre os componentes do ecossistema de TI e que evitem instabilidades ou indisponibilidades do serviço de Pix.

2.5 Big data e Inteligência Artificial

Em um mundo digital, a relação entre Big Data e Inteligência Artificial é uma das mais modernas e transformadoras. Nos últimos anos, o termo Big Data tem ganhado destaque em diversas áreas, especialmente no contexto empresarial. Refere-se ao conjunto de dados que é tão volumoso, variado e veloz que se torna difícil de ser processado por métodos tradicionais. A IA, por sua vez, é um ramo da ciência da computação que busca desenvolver sistemas capazes de simular a inteligência humana. Os avanços teóricos e práticos de metodologias e técnicas utilizadas no desenvolvimento desses Sistemas Inteligentes desempenham um papel cada vez maior na sociedade da informação (REZENDE, 1994).

Big Data é caracterizado por 3 vértices: volume, variedade e velocidade. O volume refere-se à quantidade de dados gerados, que cresce exponencialmente a cada dia. A variedade diz respeito aos diferentes tipos de dados, que podem ser estruturados, semiestruturados ou não estruturados. A velocidade refere-se à rapidez com que esses dados são gerados e precisam ser

processados. A importância do Big Data reside na sua capacidade de fornecer informações que podem ser utilizadas para a tomada de decisões estratégicas, identificação de tendências de mercado e melhoria de processos (GOOGLE CLOUD, 2025).

A IA envolve a criação de algoritmos e modelos que permitem que máquinas aprendam com os dados e realizem tarefas que normalmente requerem inteligência humana, como reconhecimento de padrões, tomada de decisões e previsão de resultados. As aplicações de IA são vastas e incluem desde assistentes virtuais, como *chatbots*, até sistemas de recomendação em plataformas de *e-commerce*. A IA é fundamental para a análise de Big Data, pois permite que as organizações extraiam percepções significativas de grandes volumes de informações (GOLDENCLOUD.TECH, 2021).

No Sicoob, o ambiente de Big Data é hospedado em nuvem pública, seguindo as regras de segurança da informação e normativos dos órgãos reguladores. Atualmente estão armazenados mais de 600 terabytes de dados – estruturados e não estruturados, sendo acessados diariamente por 7.000 usuários. São executados por mês 214.000 relatórios e painéis de gestão para análise de dados e tomada de decisão.

Todos os dias no Sicoob são gerados milhões de logs das transações financeiras que são armazenados em big data para monitoração, consultas e auditoria. A coleta e armazenamento desses grandes volumes de logs é uma prática essencial para análise do ambiente tecnológico e de comportamento do cliente. Com o aumento exponencial de dados gerados por transações em tempo real, precisamos de ferramentas eficazes para processar e analisar essas informações. A análise permite identificar padrões e distorções que podem indicar anomalias ou comportamentos comuns do sistema financeiro. Outra vantagem de análise de logs de transações é a identificação de transações que se desviam do padrão normal, permitindo que as áreas de tecnologia e negócios respondam rapidamente a possíveis quedas do ambiente tecnológico.

Com o uso de tecnologias avançadas, como inteligência artificial e aprendizado de máquina, a análise de grandes volumes de dados poderá se tornar cada vez mais eficiente, possibilitando uma resposta proativa aos aumentos inesperados de movimentos nos sistemas transacionais, garantido níveis cada vez mais elevados de resiliência do ambiente cibernético.

2.6 Observabilidade e monitoramento em TI

A observabilidade em TI é a aplicação de um sistema de constante monitoramento, rastreamento, análises e diagnósticos de um sistema, fazendo uso de algumas ferramentas de

software. A observabilidade permite investigar e entender de maneira mais aprofundada como um sistema comporta. A partir das ferramentas de observabilidade, é possível coletar e analisar um grande espectro de dados, detectar problemas, entender as causas dos problemas identificados e traçar um plano de ação para aprimorar o funcionamento dos sistemas. Dessa forma, a observabilidade ajuda a resolver problemas antes que eles afetem os *Keys Performance Indicators* (KPIs) da empresa. Além disso, transformam um problema incompreensível em algo mais detectável, visível e investigável (SOFTPLAN, 2025).

Algumas vantagens de implementar a observabilidade são: disponibilidade, pois facilita a identificação e correção de erros, minimizando o tempo de inatividade e o impacto nos usuários finais; confiabilidade, pois, quando usada de forma preventiva, a observabilidade é capaz de mitigar problemas de escalabilidade ou performance de uma aplicação antecipadamente; rastreabilidade do uso de funcionalidades, de forma a se obter insumos para decisões estratégicas que podem ser tomadas no âmbito do negócio; produtividade, pois a observabilidade permite acompanhar mudanças em tempo real, aumentando a capacidade produtiva do seu time na hora de suprir as demandas diárias, auxiliando as equipes de DevOps, infraestrutura ou outras; previsibilidade, pois quando a cultura do time está focada em rastrear e monitorar os sistemas em tempo real, a empresa consegue prever mais facilmente quedas de sistemas (SOFTPLAN, 2025).

No conceito de observabilidade em TI, temos 3 pilares: logs de evento, métricas e rastreamento. Logs de evento são registros imutáveis que apresentam hora e data dos eventos. As métricas são valores definidos de acordo com o objetivo. Elas são usadas para investigar o comportamento de um evento em um intervalo de tempo específico e fornecem dados como data, nome, hora e KPIs. O rastreamento é o pilar que detalha todas as operações, desde seu início até o fim. Isso acontece com o monitoramento da jornada completa para entender se o sistema está em normalidade ou corre algum risco de queda (DYNATRACE, 2025).

2.7 Métricas de validação de disponibilidade do ambiente de TI

As métricas de validação da disponibilidade no ambiente tecnológico são fundamentais para gestão dos ativos do TI e manutenção da sua resiliência. Definição de algumas métricas muito utilizadas (OPSERVICES, 2015).

Os KPIs são métricas quantificáveis que refletem o desempenho de um determinado processo ou sistema. Para medir a disponibilidade do ambiente tecnológico, alguns KPIs importantes incluem:

O *Service Level Agreement* (SLA) de Disponibilidade é o KPI mais direto para medir a disponibilidade. Ele representa a porcentagem do tempo total em que o sistema ou serviço esteve operacional e disponível para uso. Fórmula padrão.

$$\text{Disponibilidade}(\%) = \frac{\text{Tempo Toral de Operação} - \text{Tempo Total de Indisponibilidade}}{\text{Tempo Total da Operação}} X100$$

O MTBF (*mean time between failures* ou tempo médio entre falhas) e MTTR (*mean time to repair* ou tempo médio para reparo) são dois indicadores relacionado à disponibilidade de uma aplicação. Apesar de sua relevância no desempenho de processos, muitos gestores subutilizam esses KPIs em suas atividades de controle.

O MTBF é a métrica que se refere à média de tempo transcorrido entre uma irregularidade e o próximo lapso. Essas falhas de tempo podem ser prognosticadas por meio de uma fórmula. Já o indicador de MTTR refere-se ao prazo médio que demora para realizar uma correção depois da eventualidade, o erro. Ou seja, é o tempo gasto durante a intervenção em um determinado processo. Os dois índices são empregados como base referencial para tomar decisões nas organizações. O objetivo de uma operação é sempre aumentar o MTBF e diminuir o MTTR (OPSERVICES, 2015).

$$\text{MTBF} = \frac{\text{Tempo Total de Operação}}{\text{Número Total de Falhas}} \qquad \text{MTTR} = \frac{\text{Tempo Total de Reparo}}{\text{Número Total de Reparos}}$$

Vale salientar que estamos lidando com instalações, sistemas, processos ou equipamentos que possam ser reparados. Caso nos referíssemos a algo irreparável, o KPI exato seria MTTF (*mean time to failure* ou tempo médio para falhas). Diferenciar esses conceitos é essencial para empresas de todos os segmentos, especialmente as que trabalham com ambientes de alta disponibilidade nas quais as falhas podem gerar grandes prejuízos com vendas não concretizadas ou com perda de confiança na entrega dos serviços.

As diferenças são conceituais, portanto temos fórmulas distintas. O MTBF que ao longo de um determinado período disponível para atuar, foi notado como período integral disponível para operar = 24 horas. Aconteceram 3 paralisações a cada uma delas: 1 hora, 2 horas e 30 minutos (0,5 horas).

O MTBF agora calculado com o exemplo acima:

MTBF = [24 – (1 + 2 + 0,5)] / 3 = 6,8333 horas ou 410 minutos

O MTTR usando mesmo exemplo da MTBF:

MTTR = (1 + 2 + 0,5) / 3 = 1,1666 horas ou 70 minutos

Gerando uma relação com os dois índices, pode-se concluir que, a cada 2 horas, o programa se encontrará indisponível por 15 minutos. Quanto menor o MTTR, mais eficiente é a equipe de manutenção (OPSERVICES, 2015).

Com o acompanhamento dos indicadores de tempo médio entre as falhas e tempo médio para reparo consegue-se aumentar a cultura da empresa sobre seus procedimentos. Isso é particularmente crítico em um contexto no qual as equipes de manutenção necessitam se manter reduzidas e são formadas por colaboradores que se oferecem a várias áreas simultaneamente.

Logo após indicar uma frequência otimizada para trabalhar os dois indicadores, inicie a implantação das seguintes etapas: identificação da falha — com o relatório, pode-se notar as ocorrências por padrão de falha e o impacto em tempo da pausa para esses casos. Dessa forma, consegue-se restringir a classe de problema e reconhecer se ele refere a alguma parte específica que pode ter adversidade; diminuição do tempo de inatividade — nesse instante, você sabe qual componente traz mais impasses para corrigi-lo e elabore um processo de manutenção preditiva não pratique a falha de sempre executar manutenções corretivas sem conseguir compreender a razão do problema.

Classificar as condições desses dispositivos exige uma tarefa detalhada de análise e coleta de informações dos segmentos monitorados, introduzindo nesse contexto a manutenção preditiva como instrumento primordial. Como pode ser percebido, MTTR e MTBF são dois indicadores de performance poderosos e que devem ser utilizados para ampliar o conhecimento da empresa sobre seus processos e reduzir perdas de produtividade ou qualidade nos produtos oferecidos (OPSERVICES, 2015).

2.8 Detecção de anomalias com Isolation Forest em Python e Scikit-learn

Combater fraudes, defender redes, identificar discrepâncias e sinalizar equipamentos defeituosos podem parecer problemas distintos. No entanto, eles compartilham uma característica comum: não há rótulos ou definições claras sobre o que constitui uma anomalia. Em muitos casos, as anomalias são raras e sutis, dificultando sua identificação em meio a vastas quantidades de dados normais. Uma maneira de desenvolver sistemas de detecção de anomalias é usar aprendizado de máquina preditivo (O´SULLIVAN, 2024).

Isolation Forest é um algoritmo de aprendizado de máquina não supervisionado que identifica anomalias ou outliers em dados isolando-os por meio de um processo de particionamento aleatório dentro de uma coleção de árvores de decisão. No seu funcionamento, o Isolation Forest é composto por um conjunto de árvores de decisão (chamadas árvores de isolamento). Cada árvore isola os pontos de dados através de partições aleatórias. Anomalias geralmente requerem menos partições para serem separadas, resultando em caminhos mais curtos nas árvores. A pontuação de anomalia de um ponto é baseada no comprimento médio do caminho necessário para isolá-lo em todas as árvores da floresta. Pontos com pontuações de anomalia mais altas são considerados mais prováveis de serem anomalias. Como vantagens, o Isolation Forest tem uma complexidade de tempo linear e baixo uso de memória, tornando-o adequado para grandes conjuntos de dados; não requer rótulos de anomalias durante o treinamento, o que é útil quando as anomalias são raras ou desconhecidas; e é geralmente mais rápido do que outros algoritmos de detecção de anomalias. Quanto às limitações, pode não ser tão eficaz em conjuntos de dados pequenos, onde o processo aleatório pode levar a resultados inconsistentes e identifica potenciais anomalias, mas não explica por que elas são anômalas, exigindo análise adicional (O´SULLIVAN, 2024).

Scikit-learn é uma biblioteca em Python com ferramentas simples e eficientes para análise preditiva de dados. Tem uma série de características que a tornam muito útil, dentre as quais destacamos: facilidade de uso, porque possui uma interface consistente e intuitiva, tornando mais fácil para iniciantes e especialistas a implementação de algoritmos de aprendizado de máquina; ampla gama de algoritmos de algoritmos supervisionados e não supervisionados, permitindo que os usuários escolham a melhor abordagem para seus problemas; funciona perfeitamente com outras bibliotecas Python populares como NumPy, SciPy e Matplotlib, facilitando o processamento e a visualização de dados; é mantido por uma comunidade ativa de desenvolvedores e pesquisadores, o que garante atualizações frequentes e suporte abrangente; e sendo de código aberto, qualquer pessoa pode contribuir para o projeto e

acessar seu código-fonte. Com ela você pode identificar e categorizar dados em classes pré-definidas, como classificar e-mails como spam ou não spam; prever valores numéricos contínuos, como prever o preço de uma casa com base em suas características. agrupar dados semelhantes em clusters, como agrupar clientes com base em seus comportamentos de compra; e reduzir o número de variáveis em um conjunto de dados, facilitando a visualização e a análise (SCIKIT-LEARN, 2024).

2.9 Deficiências e lacunas das atuais tecnologias

A resiliência cibernética é essencial para garantir a segurança e continuidade operacional das infraestruturas tecnológicas de grandes bancos. A análise dos documentos estudados revela desafios críticos relacionados às tecnologias empregadas, bem como soluções potenciais para mitigar riscos e melhorar a segurança operacional. Atualmente, existem várias deficiências das tecnologias empregadas que vamos citar 3 adiante.

A primeira é a análise de logs e detecção de falhas. Apesar do avanço no uso de modelos de aprendizado profundo como BERT e LSTM para análise de logs, ainda existem limitações significativas, tais como a falta de padronização dos logs, onde diferentes sistemas geram logs em formatos variados, dificultando sua análise unificada, dados desbalanceados, pois a maioria dos logs são normais, o que dificulta a detecção de anomalias raras e relevantes e latência na detecção de falhas, pois os modelos atuais podem não ser rápidos o suficiente para detectar e mitigar ataques em tempo real (BHANAGE; PAWAR; KOTECHA; ABRAHAM, 2023).

Depois, o processamento de dados em tempo real, pois muitos frameworks de machine learning não foram projetados para lidar com fluxos de dados contínuos. Isso gera desafios como: baixa integração com sistemas legados onde a implementação de soluções de streaming requer adaptação significativa dos sistemas tradicionais, escalabilidade limitada quando aumento no volume de transações bancárias pode sobrecarregar soluções centralizadas; e falta de automação na gestão de modelos, pois a maioria das plataformas exige intervenção manual para atualização e melhoria de modelos de IA (MARTIN; LANGENDOERFER; ARRIN; DÍAZ; BARTOLOMÉ, 2022).

Por fim, o diagnóstico de falhas e resiliência, pois muitas abordagens de detecção de falhas ainda dependem de processos reativos. Entre as limitações estão o diagnóstico impreciso pois a análise de eventos de falha com causas raiz ainda apresenta desafios, a dependência de processos manuais quando a recuperação de falhas muitas vezes exige análise manual, reduzindo a eficiência operacional e a falta de padronização na segurança da nuvem, sendo que

muitas soluções carecem de uma abordagem unificada para resiliência em ambientes distribuídos (BHANAGE; PAWAR; VISHAL; KOTECHA, 2021).

3 Metodologia

3.1 Visão Geral da Metodologia

A resiliência cibernética de uma grande instituição financeira depende da capacidade de detecção de falhas, processamento de dados em tempo real e recuperação rápida de incidentes. A implementação de soluções baseadas em aprendizado de máquina, monitoramento automatizado e arquiteturas distribuídas são essenciais para garantir a segurança e estabilidade da infraestrutura tecnológica. ML e IA dependem de fontes de dados para treinar, melhorar e fazer previsões por meio de seus algoritmos, em um mundo onde as informações estão mudando de dados estáticos para fluxos de dados contínuos. No entanto, a maioria das estruturas de ML/IA usadas atualmente não está preparada para essa revolução, conforme pode ser verificado nas pesquisas contidas no item 2.9 – Deficiências e lacunas das atuais tecnologias.

No Sicoob, temos um ambiente robusto de big data com bilhões de informações das transações financeiras realizadas pelos nossos cooperados. Com ferramentas automatizadas e atualizadas, quando ocorrem quedas das soluções tecnológicas, conseguimos analisar as causas dessas quedas, causadoras de prejuízos financeiros enormes, e tomar ações para reestabelecer o ambiente ao seu estado operacional na maior brevidade possível. A grande dor é termos esse volume gigante de informações organizadas em big data, mas sempre estamos tomando ações de forma reativa, sem conseguir evitar a queda do ambiente tecnológico e os prejuízos decorrentes dessas paradas.

A metodologia a ser elaborada envolverá conceitos e técnicas pesquisadas para criação de modelo para coleta de dados contidos nos milhões de logs envolvidos nas transações de Pix realizados no Sicoob. Com as informações contidas nos logs criaremos um *dataset* on-line com informações das transações: tempos de resposta, sucesso ou perda, *timestamp* etc. Esse conjunto de dados será projetado para operar com grandes volumes de dados (big data), oferecendo suporte tanto a análises em tempo real quanto a análises históricas.

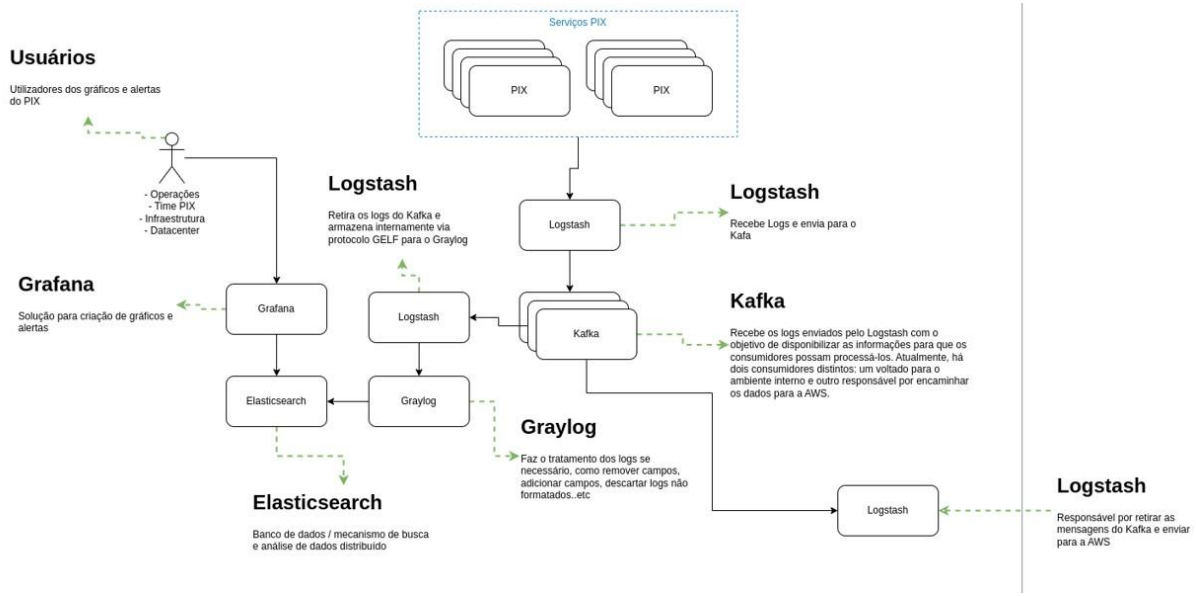
A partir de técnicas avançadas de IA em ambiente Big Data, poderemos aprimorar a análise desse *dataset* em tempo de execução, realizar análise exploratória dos dados com foco na geração de percepções, investigar, compreender os dados coletados e implementar inferências em tempo real, objetivando identificar de padrões e comportamentos com o intuito de detectar e classificar anomalias do ambiente tecnológico de forma a torná-lo cada vez mais resiliente, evitando falhas na estrutura que acarretam parada total ou parcial dos serviços Pix.

Além disso, gerar alertas para sermos mais proativos na tomada de decisões e melhorar o diagnóstico e o tempo para recuperação de falhas.

3.2 Fluxo de logs do Pix no ambiente de Big Data do Sicoob

A seguir diagrama do fluxo de logs do Pix no ambiente de Big Data do Sicoob.

Figura 2 – Fluxo de logs do Pix no Ambiente de Big Data do Sicoob



Fonte: (AUTOR, 2025).

A Figura 2 – Fluxo de logs do Pix no Ambiente de Big Data do Sicoob mostra o fluxo de logs do Pix no ambiente de Big Data do Sicoob, desde sua coleta, inserção, armazenamento, tratamento e visualização. O sistema de monitoramento de logs do Pix no Sicoob foi estruturado com o objetivo de garantir a rastreabilidade, o tratamento e a visualização eficiente dos eventos gerados pelos serviços Pix. O fluxo é composto por diversas tecnologias que atuam desde a coleta até a apresentação final dos dados, bem como a exportação para a nuvem.

a) Geração e Coleta Inicial dos Logs

Os logs são inicialmente gerados pelos diversos serviços que compõem a plataforma PIX. Esses registros, brutos no formato JSON, contêm informações detalhadas sobre o funcionamento interno, erros, métricas e eventos dos serviços. Para garantir o envio padronizado desses dados a infraestrutura de observabilidade, os serviços utilizam o protocolo

GELF (Graylog Extended Log Format) sobre JSON, um formato amplamente adotado por ferramentas de análise de logs.

Esses logs JSON são enviados a uma instância do Logstash, ferramenta *open source*, parte integrante do ecossistema Elastic Stack (ELK), desenvolvida para processar, transformar e encaminhar dados de log de diversas fontes para múltiplos destinos. A principal função é atuar como um pipeline de coleta e processamento de dados, permitindo que informações brutas (*raw data*), como logs dos micros serviços, métricas de sistemas ou eventos de segurança, sejam recebidas de forma eficiente e flexível. A arquitetura do Logstash é baseada em uma cadeia de processamento modular composta por três estágios principais: entrada, filtro e saída.

No estágio de entrada, o Logstash recebe os logs de diferentes fontes de dados utilizando plugins. Entre os plugins de entrada mais utilizados estão os beats, utilizados para receber dados do Filebeat, o syslog, para integrar com sistemas que enviam logs via protocolo Syslog, o http, para ingestão de dados via chamadas REST e na arquitetura atual, comumente utilizada para integrar com sistemas que emitem logs no formato GELF. As diversas entradas permitem que o Logstash opere como um concentrador de dados com suporte a múltiplos protocolos.

Uma vez recebidos, os dados passam pela etapa de filtro, onde são transformados, enriquecidos ou reestruturados conforme as necessidades do ambiente, sendo neste caso, utilizado no processo de sanitização dos dados, removendo campos desnecessários. Os filtros utilizam plugins como grok para parsing de expressões regulares, mutate para remoção ou renomeação de campos, date para padronização de timestamps, e geoip para enriquecer logs com informações geográficas baseadas em endereços IP. Essa etapa é especialmente importante para padronizar e qualificar os dados antes de serem encaminhados, garantindo a consistência e utilidade para consumo analítico posterior.

No estágio final de saída, os dados processados são enviados para um ou mais destinos. Os destinos mais comuns incluem o Elasticsearch, utilizado para indexação e busca de logs em tempo real, o Kafka, para integração com pipelines de dados distribuídos e também sistemas como Amazon S3, bases de dados relacionais, e até mesmo arquivos locais. A capacidade do Logstash de encaminhar dados para múltiplas saídas simultaneamente o torna altamente flexível para arquiteturas híbridas ou distribuídas.

Por ser extensível e altamente configurável, o Logstash é amplamente utilizado em ambientes corporativos para consolidar logs de aplicações, servidores, containers e dispositivos de rede.

b) Transporte de Logs com Apache Kafka

O Apache Kafka é utilizado como barramento de dados, garantindo o transporte confiável e escalável dos logs. Os dados recebidos do Logstash são disponibilizados para múltiplos consumidores. Entre os consumidores existentes, destacam-se: um responsável por processar e disponibilizar os dados no ambiente interno e outro dedicado ao envio de dados para a nuvem AWS (*Amazon Web Services*). Para compreender melhor a solução adotada, é fundamental apresentar, de forma conceitual, o funcionamento dos produtores, tópicos, partições e consumidores — elementos centrais para entender o papel do Kafka na arquitetura.

Os produtores são os clientes responsáveis por publicar mensagens nos tópicos do Kafka. Eles são tipicamente aplicações ou sistemas que geram dados, como servidores de aplicação, coletores de logs (neste caso o Logstash), sensores IoT ou APIs. Os produtores se conectam ao cluster Kafka e, ao publicar mensagens, podem definir qual partição do tópico receberá a mensagem, seja por meio de uma chave de particionamento (*partition key*) ou por balanceamento automático. A eficiência dos produtores é aumentada pelo modelo assíncrono de publicação do Kafka, permitindo o envio em lotes (batch) e compressão de mensagens, além de reduzir o uso de rede e obter maior performance como resultado.

Dentro da arquitetura do Apache Kafka, os tópicos são entidades lógicas que funcionam como canais de comunicação para onde os dados são enviados e posteriormente consumidos. Todo dado publicado no Kafka precisa ser associado a um tópico. Internamente, cada tópico pode ser dividido em múltiplas partições, o que permite o processamento paralelo e a distribuição das mensagens entre diferentes nós (*brokers*) do cluster. Essa divisão por partições é fundamental para a escalabilidade horizontal do Kafka, pois permite que múltiplos consumidores leiam diferentes partes de um mesmo tópico simultaneamente, maximizando, mais uma vez, a performance.

Para garantir maior segurança do dado, no que diz respeito a perda de informações, os tópicos são replicados, ou seja, dentro do ambiente existem cópias das partições de um tópico que são mantidas em diferentes nós dentro do cluster Kafka. O objetivo principal da replicação é garantir alta disponibilidade e tolerância a falhas. Para cada partição, uma das réplicas é eleita como líder, sendo responsável por todas as operações de leitura e escrita. As demais réplicas atuam como seguidores, mantendo-se sincronizadas com a partição líder. Caso o nó líder falhe, um dos seguidores é automaticamente promovido, assegurando a continuidade do serviço sem perda de dados, desde que o fator de replicação e a política de sincronização estejam corretamente configurados.

Por outro lado, os consumidores são as aplicações ou serviços que se conectam ao Kafka com o objetivo de ler e processar as mensagens armazenadas nos tópicos, sendo novamente o Logstash, conforme Figura 2. Eles são organizados em grupos de consumidores (*consumer groups*), o que permite que diferentes instâncias de um mesmo serviço compartilhem a carga de leitura de forma coordenada. Cada partição de um tópico é atribuída a apenas um consumidor dentro de um grupo, evitando leitura duplicada e garantindo paralelismo sem inconsistência. Os consumidores mantêm o controle de sua posição de leitura por meio de offsets, permitindo retomar o processamento a partir de um ponto específico, mesmo após falhas ou reinicializações.

Com base na compreensão do funcionamento do Kafka, é possível afirmar que, na arquitetura atual, existe um grupo de instâncias do Logstash atuando como produtores, responsáveis por receber os dados dos micros serviços por meio de uma entrada e enviá-los ao Kafka através de uma saída. Esses dados são armazenados em tópicos replicados no Kafka e, em seguida, são consumidos por um segundo grupo de instâncias do Logstash, que atua como consumidor. Nessa segunda etapa, os dados passam por filtros de sanitização antes de serem encaminhados, também via saída do Logstash, ao Graylog.

Com base no entendimento do funcionamento do Logstash e do Kafka, é possível afirmar que, na topologia atual, o Logstash é empregado em dois pontos distintos do fluxo de logs. O primeiro grupo de instâncias atua como produtor, sendo responsável por coletar os logs gerados pelos micros serviços por meio de uma entrada do tipo Graylog Extended Log Format (GELF)). Nessa etapa, os dados são recebidos sem qualquer transformação ou filtragem e são encaminhados diretamente, por meio de uma saída, para um tópico no Kafka. Esses dados ficam armazenados em tópicos replicados, conforme a arquitetura distribuída do Kafka. Posteriormente, um segundo grupo de instâncias do Logstash entra em ação, atuando como consumidor. Esse grupo consome as mensagens do Kafka, aplica filtros de sanitização — removendo informações irrelevantes ou sensíveis — e, em seguida, envia os logs tratados para o Graylog por meio de uma nova saída.

c) Processamento de Logs com Graylog

Após os logs serem consumidos do Kafka por novas instâncias do Logstash, eles são enviados ao Graylog utilizando o protocolo GELF. Embora o Graylog possua funcionalidades avançadas para manipulação de dados, como remoção ou inclusão de campos e enriquecimento

com metadados, neste cenário ele atua como ponto de ingestão, já que o tratamento dos dados foi previamente realizado pelo Logstash.

O Graylog recebe os logs já formatados e os envia diretamente para o Elasticsearch, onde são indexados para posterior análise e visualização. Além de centralizar a busca e visualização de logs em uma interface mais amigável, o Graylog oferece vários recursos administrativos que simplificam o gerenciamento do ambiente, reduzindo a carga administrativa para os responsáveis pela solução.

Por meio do Graylog é possível gerenciar os índices do Elasticsearch de forma centralizada, definindo políticas de rotação por retenção, tempo ou tamanho dos índices. Isso ajuda a manter o cluster do Elasticsearch saudável, evitando o crescimento descontrolado do armazenamento, ou seja, governança dos logs. O Graylog também permite a criação de index sets para segmentação de log, facilitando o controle de acesso, a organização dos dados.

Além disso, o Graylog permite a configuração de serviços automatizados, como tarefas de arquivamento, geração de alertas baseados em padrões específicos de logs, e o envio de notificações por e-mail ou via integrações com ferramentas externas. Esses recursos ampliam significativamente as capacidades da plataforma no que diz respeito à governança, rastreabilidade e auditoria dos eventos registrados.

No entanto, na arquitetura atual, embora o Graylog disponha dessas funcionalidades, a responsabilidade por alertas e notificações está centralizada no Grafana, que assume esse papel com base em métricas e logs processados em tempo real.

d) Armazenamento com Elasticsearch

Após o processamento e sanitização do dado feito pelo Logstash, e a governança feita pelo Graylog, os logs são armazenados no Elasticsearch, que atua como um mecanismo de busca e banco de dados distribuído, do tipo séries temporais (*time series*). Sua arquitetura foi projetada para lidar com grandes volumes de dados, com alta disponibilidade, escalabilidade e rapidez nas buscas, mesmo em ambientes com alto tráfego de informações. Essa performance se deve a forma como o Elasticsearch estrutura os dados em índices, que são automaticamente particionados e replicados entre os nós do cluster.

A replicação entre os índices no Elasticsearch é feita utilizando pedaços (*shards*) primários e réplicas. Cada índice é dividido em pedaços, que são distribuídos entre os nós do cluster. Para cada pedaço primário, é possível configurar um ou mais pedaços de réplica, que atuam como cópias exatas dos dados. Esses pedaços de réplica são alocados automaticamente

em nós diferentes dos pedaços primários, garantindo tolerância a falhas e continuidade da operação mesmo que um nó seja perdido. Esse mecanismo também permite distribuir as consultas entre múltiplos pedaços, melhorando o desempenho de leitura e a capacidade de resposta do sistema.

Ao mesmo tempo, a interface do Graylog torna todo esse processo de administração de índice, pedaços, replicações e monitoração mais amigáveis, pois que o administrador do ambiente, implementar todos os controles necessários sem a necessidade de lidar diretamente com os comandos administrativos complexos do Elasticsearch.

No contexto do gerenciamento de logs, essa soma de desempenho, escalabilidade e integração com o Graylog torna o Elasticsearch fundamental para o acesso rápido aos dados utilizando a interface do Graylog, ou com outras soluções que podem ser acopladas ao Elasticsearch, a exemplo do Grafana.

e) Visualização com Grafana

A camada de visualização é fornecida pelo Grafana, uma ferramenta que se conecta ao Elasticsearch para criação de painéis de gráficos dinâmicos e alertas personalizados. Os usuários dessa camada incluem equipes de operações de TI, time Pix e equipes de infraestrutura. Esses usuários utilizam o Grafana para acompanhar o comportamento do sistema em tempo real e receber alertas automáticos em caso de mudanças de comportamento das transações Pix.

A integração entre o Grafana e o Elasticsearch é feita por meio de fontes de dados, que são conectores utilizados para acessar diferentes repositórios de dados. O Elasticsearch é um dos vários tipos de fontes de dados disponíveis no Grafana, a solução também permite a criação de fontes de dados para conexões com bancos de dados relacionais (como PostgreSQL e MySQL), sistemas de séries temporais (como Prometheus e InfluxDB), entre outros. Fazendo com que o Grafana seja flexível, permitindo que uma mesma interface seja usada para centralizar a visualização de dados provenientes de diferentes fontes.

Além da visualização em tempo real, o Grafana permite construir painéis históricos que ajudam a identificar tendências, gargalos e padrões de comportamento ao longo do tempo. Isso é possível graças à flexibilidade de suas queries, que podem extrair e combinar dados de diferentes fontes, com filtros por intervalo de tempo, tipo de transação ou origem da requisição. A integração com o Elasticsearch garante que os dados apresentados estejam sempre atualizados e que a navegação entre períodos e métricas seja simplificada, mesmo em ambientes

com grande volume de logs. Dessa forma, o Grafana atua não apenas como uma ferramenta de monitoramento, mas também como uma plataforma de análise e suporte à tomada de decisão.

f) Exportação para a Nuvem

Paralelamente ao consumo interno, há uma instância dedicada do Logstash que consome dados do Kafka e realiza o envio estruturado dos logs para a nuvem AWS, permitindo que os dados sejam armazenados ou processados em nuvem para fins de backup, auditoria ou análise adicional.

3.3 Integração e funcionamento do Ambiente de Big Data

No tópico anterior, foram apresentados individualmente os principais componentes do ambiente de Big Data do Pix no Sicoob, destacando suas funções e detalhes técnicos de cada produto utilizado na arquitetura. De qualquer forma, compreender cada tecnologia isoladamente não é suficiente. É importante entender como essas tecnologias se conectam, formando um fluxo contínuo de dados, e quais desafios surgem nesse nas integrações.

O fluxo de dados do Pix funciona como um encadeamento de componentes, no qual cada etapa se comunica com o próximo componente por meio de protocolos e mecanismos específicos, garantindo que os dados circulem de forma segura, ou seja, sem perda ou duplicidade. A seguir, o funcionamento de forma integrada do fluxo de dados, etapa por etapa.

Na etapa Geração e Coleta Inicial, os serviços Pix produzem registros em JSON seguindo o padrão GELF. Esses logs são enviados para o Logstash, entrando pela entrada específica do tipo GELF. Nesse ponto, o Logstash atua como concentrador de dados, recebendo registros de diferentes serviços. Em seguida, envia os logs para o Kafka via saída, atuando como produtor de mensagens.

Depois, na etapa de Transporte via Kafka, as mensagens publicadas pelos produtores (Logstash) são armazenadas em tópicos replicados. Cada tópico pode conter múltiplas partições, garantindo escalabilidade e paralelismo no consumo. A replicação entre brokers assegura alta disponibilidade e tolerância a falhas. Consumidores, incluindo outro grupo de instâncias do Logstash, podem ler simultaneamente os dados, respeitando os offsets para evitar duplicidade ou perda de mensagens. Este componente é importante para a operação dos próximos elementos do fluxo, pois permite que eles sejam desligados temporariamente para

manutenção ou atualização sem interromper a ingestão de dados. Para isso, o Kafka armazena e enfileira as mensagens até que os componentes seguintes sejam restabelecidos, garantindo a continuidade e integridade do fluxo de logs.

Como Processamento e Sanitização, o segundo grupo de Logstash atua como consumidor do Kafka. Ele lê os dados dos tópicos, aplica filtros de sanitização, enriquecimento e transformação adicionais e encaminha os logs via saída para o Graylog. Essa etapa é crítica, pois a configuração dos filtros e a capacidade de consumo determinam se os dados chegam íntegros e confiáveis à camada seguinte.

Na etapa Armazenamento e Indexação, o Graylog recebe os logs processados e os envia para o Elasticsearch, que indexa e armazena os registros em índices distribuídos com shards primários e réplicas. A integração entre Graylog e Elasticsearch facilita a administração, permitindo o gerenciamento centralizado de índices, políticas de retenção e segmentação de dados.

E, finalmente, na etapa Visualização e Monitoramento, o Grafana consome os dados do Elasticsearch por meio de fontes de dados configurados, transformando-os em dashboards interativos e alertas. A integração garante que as métricas e logs estejam sempre atualizados, permitindo acompanhamento em tempo real, análise histórica e suporte à tomada de decisão. Na Exportação para a Nuvem, de forma paralela, outra instância do Logstash consome dados do Kafka e envia para a AWS, garantindo backup, redundância e análises externas.

Em suma, a integração entre os componentes do fluxo de logs é organizada como uma cadeia contínua: a entrada do Logstash recebe os dados iniciais, que são processados e enviados via saída para tópicos no Kafka, consumidos novamente pelo Logstash e encaminhados para armazenamento e visualização. Cada acoplamento entre etapas possui um desafio operacional diferente, exigindo atenção à configuração, monitoramento e ajustes finos (*tuning)* para garantir integridade, escalabilidade e capacidade de ingestão dos dados.

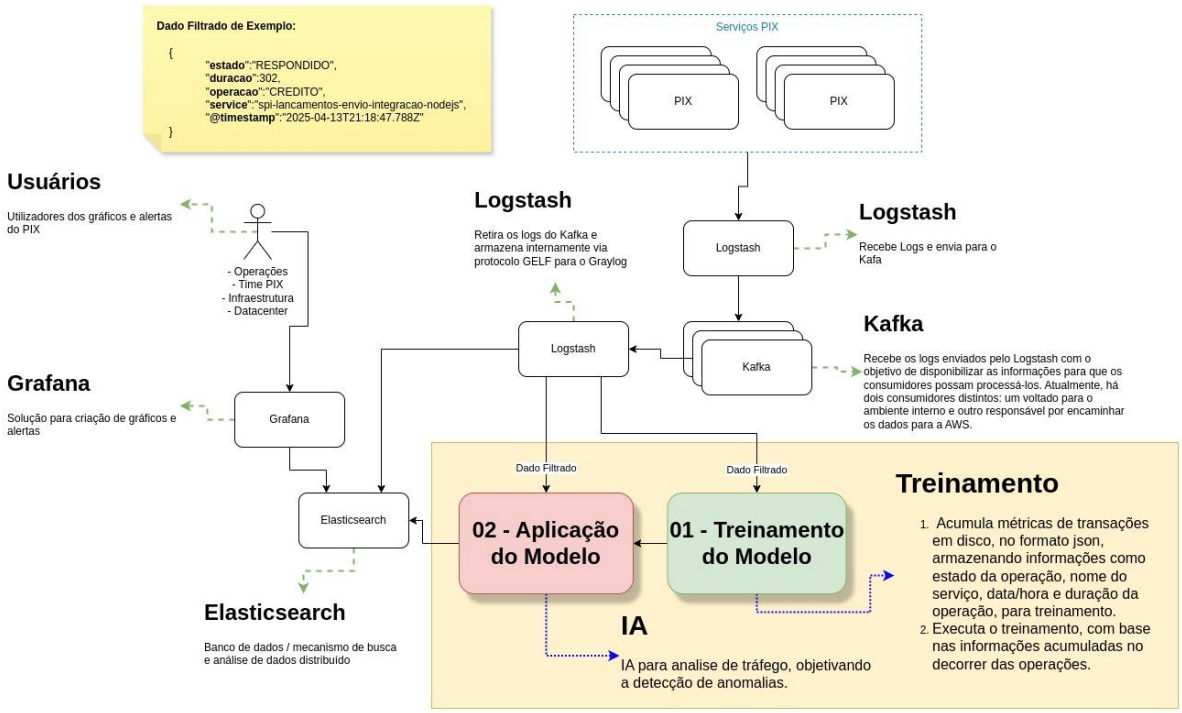
3.4 Preparação de modelo com IA no ambiente de Big Data

A principal questão a ser endereçada nesse ambiente Pix demonstrado na Figura 2 da seção anterior é detectar problemas antes de impactar ou causar a queda dos serviços de Pix. Nesse contexto, propõe-se a introdução, no fluxo de logs do Pix Sicoob, de duas novas etapas ao processo: a etapa de treinamento e a etapa de análise de dados, objetivando a detecção automática de problemas e anomalias em tempo real na infraestrutura tecnológica, para gerar

alertas para os times técnicos e para a própria infraestrutura, de forma que sejam tomadas ações rápidas para evitar a interrupção dos serviços.

A Figura 3 – Ambiente Big Data do Pix com introdução de IA mostra a proposta de implementação de IA no ambiente Big Data do Pix do Sicoob.

Figura 3 – Ambiente Big Data do Pix com introdução de IA



Fonte: AUTOR, 2025.

O desafio é a implementação de um modelo baseado em IA, que se acopla ao ambiente atual para aprendizado e detecção de anomalias, operando de forma complementar ao processo atual.

A primeira etapa é o treinamento do modelo, no qual os dados coletados precisam ser preparados e organizados para formar a base de aprendizado. Essa fase é fundamental para que o sistema adquira capacidade de reconhecer padrões e gerar previsões consistentes. Concluído o treinamento, inicia-se a segunda etapa, a aplicação do modelo, em que o conhecimento adquirido passa a ser utilizado em situações reais para apoiar análises, prever cenários e orientar decisões. A seguir descrevemos as etapas de implementação do modelo proposto para treinamento e detecção de anomalias no ambiente de Big Data do Pix do Sicoob.

a) Treinamento do Modelo (Verde – Etapa 1)

Nesta seção, são definidas as etapas com suas atividades fundamentais para aquisição de dados e para o treinamento do modelo. Para a aquisição de dados, foi criado um arquivo de configuração no segundo grupo de Logstash já existente, que espelha o tópico no Kafka, remove campos desnecessários, tendo como resultado um *dataset* com nome específico e salvo em disco local. A Figura 4 – Arquivo de configuração do Logstash representa o arquivo de configuração utilizado para geração do *dataset*.

Figura 4 – Arquivo de configuração do Logstash

```
input {
  kafka {
    bootstrap_servers => "kafp2001:9092,kafp2002:9092,kafp2003:9092"
    codec => json
    topics => "mon-pix"
    consumer_threads => 3
    group_id => "ai-training"
    auto_commit_interval_ms => "5000"
    client_id => 'ai-training'
  }
}

filter {
  if ([service] and [operacao] == "CREDITO" and [duracao]) {
    mutate {
      remove_field => [
        "instOrigem", "instDestino", "short_message", "source_host",
        "id", "message", "full_message", "host",
        "@version", "tags", "type", "source", "gl2_*", "facility",
        "file", "stack", "environment", "valor", "descLevel", "method",
        "prioridade", "relativePath", "level", "line", "pos", "version",
        "agendado", "boolIntercredis", "idRequisicaoPagamento", "tipoQrcode"
      ]
    }
  } else {
    drop { }
  }
}

output {
  file {
    path => "/media/ai/01-massa-de-dados.json"
    codec => json_lines
  }
}
```

Fonte: AUTOR, 2025.

O arquivo de configuração é composto por três blocos principais. O primeiro bloco é a entrada, que corresponde à conexão com o Kafka, onde o Logstash se conecta para realizar a aquisição dos dados. Nesse bloco são definidos os servidores Kafka, o formato das mensagens,

o tópico ao qual o Logstash irá se conectar e o grupo de consumo utilizado para identificar a instância do Logstash no Kafka.

Depois, o segundo bloco é o filtro que é responsável pela etapa de sanitização dos dados, ou seja, a remoção de campos desnecessários. Essa filtragem resulta em uma massa de dados enxuta, reduzindo o tamanho final do *dataset*.

No último bloco que é a saída, temos a etapa final, na qual é gerado o *dataset* com os dados necessários para o treinamento do modelo. O resultado é salvo em um único arquivo, em que cada linha corresponde a um log distinto. Essa estrutura é garantida pelo uso do codec json_lines, que armazena cada registro em formato JSON em apenas uma linha.

Durante esse processo de geração do *dataset*, apenas os campos relevantes para o treinamento foram mantidos, armazenados em um *dataset* cujo nome físico é "01-massa-de-dados.json". Esse *dataset* é composto pelos campos chamados estado (respondido ou não), duração (tempo de resposta em milissegundos), operação (tipo de serviço Pix podendo ser débito ou crédito), serviço (qual serviço envolvido na transação e *timestamp* (data e hora da operação). Um exemplo de carga no *dataset* é ("RESPONDIDO", 302, "CREDITO", "SPI-LANÇAMENTOS-ENVIO-INTEGRAÇÃO-NODEJS", "2025-04-17T21:18:47.7882").

Concluída a atividade de aquisição de dados, iniciamos o treinamento do modelo com base no *dataset* gerado, sendo necessária a utilização de um script Python, que irá utilizar as bibliotecas para o treinamento. As figuras a seguir detalham o script com seus respectivos arquivos de entrada e saída, divididos em 5 passos, para melhor compreensão do treinamento.

Figura 5 – Importação das bibliotecas

```
# ------------------------------------------------------------------------
# ETAPA 1: Importação das bibliotecas necessárias
# ------------------------------------------------------------------------

# Importa a biblioteca pandas, utilizada para leitura e manipulação de dados tabulares (DataFrames)
import pandas as pd

# Importa o algoritmo IsolationForest para detecção de anomalias, da biblioteca scikit-learn
from sklearn.ensemble import IsolationForest

# Importa o codificador de variáveis categóricas para números inteiros, da biblioteca scikit-learn
from sklearn.preprocessing import LabelEncoder

# Importa a função dump da biblioteca joblib, que permite salvar objetos Python em arquivos
from joblib import dump

# Importa o módulo datetime para manipulação de datas e horas
from datetime import datetime

# Importa a biblioteca json, usada para carregar arquivos em formato JSON
import json
```

Fonte: AUTOR, 2025.

Nesse bloco do script, as bibliotecas necessárias são importadas, conforme Figura 5 – Importação das bibliotecas. São elas a Pandas, utilizada para manipulação de grandes volumes de dados, o json, para tratamento de dados no formato JSON; a datetime, para registro e identificação temporal e, principalmente, a biblioteca sklearn, utilizada no treinamento do modelo, que é a Scikit-learn com seu módulo Isolation Forest.

Figura 6 – Leitura e preparação dos dados

```
# Define o nome do arquivo de entrada contendo os dados no formato JSON linha a linha
dados = "01-massa-de-dados.json"

# Abre o arquivo JSON em modo de leitura
with open(dados, "r") as f:
    # Lê cada linha do arquivo, remove linhas em branco e converte de JSON para dicionário Python
    linhas = [json.loads(linha) for linha in f if linha.strip()]

# Constrói um DataFrame (tabela de dados) a partir da lista de dicionários lidos do JSON
df = pd.DataFrame(linhas)

# Converte a coluna '@timestamp' para o formato de data/hora reconhecido pelo pandas
df["@timestamp"] = pd.to_datetime(df["@timestamp"])

# Cria nova coluna "hora" contendo apenas a hora (0 a 23) extraída do timestamp
df["hora"] = df["@timestamp"].dt.hour

# Cria nova coluna "minuto" contendo apenas os minutos (0 a 59) do timestamp
df["minuto"] = df["@timestamp"].dt.minute

# Cria nova coluna "dia_semana" com o número do dia da semana (0 = segunda, 6 = domingo)
df["dia_semana"] = df["@timestamp"].dt.dayofweek
```

Fonte: AUTOR, 2025

O trecho de código apresentado na Figura 6 – Leitura e preparação dos dados, tem como objetivo ler, processar e estruturar os dados provenientes do arquivo JSON, preparando-os para análises e posterior treinamento do modelo. Primeiramente, define-se o arquivo de entrada, que contém os dados no formato JSON, armazenados linha a linha. Em seguida, o script abre o arquivo em modo de leitura e percorre cada linha, removendo eventuais linhas em branco e convertendo o conteúdo JSON em dicionários Python.

A partir dessa lista de dicionários, é construído uma tabela de dados utilizando a biblioteca Pandas, que organiza os dados em formato tabular, facilitando o manuseio e a análise. A coluna @timestamp é convertida para um formato de data e hora reconhecido pela Pandas, permitindo a extração de informações temporais específicas. Depois, são criadas colunas: hora, que contém apenas a hora da ocorrência (0 a 23); minuto, que registra os minutos (0 a 59); e dia_semana, que indica o número correspondente ao dia da semana (0 para segunda-feira até 6 para domingo).

Essas transformações permitem que os dados fiquem estruturados de maneira adequada, tornando possível analisar padrões temporais e utilizar essas informações como variáveis de entrada no processo de treinamento do modelo.

Figura 7 – Codificação de variáveis categóricas

```
# Cria uma instância do codificador LabelEncoder
le_service = LabelEncoder()

# Codifica a coluna "service" (nomes de serviços) em números inteiros, e armazena em nova coluna
df["service_encoded"] = le_service.fit_transform(df["service"])
```

Fonte: AUTOR, 2025

O trecho de código apresentado na Figura 7 – Codificação de variáveis categóricas utiliza a classe LabelEncoder da biblioteca scikit-learn para converter valores categóricos em valores numéricos, permitindo que dados não numéricos possam ser processados pelo modelo de aprendizado de máquina. Nesse caso, a coluna *service*, que contém os nomes dos serviços Pix, é transformada em números inteiros correspondentes a cada categoria. O resultado é armazenado em uma nova coluna chamada service_encoded.

Essa codificação é essencial para que algoritmos de aprendizado de máquina, que geralmente trabalham apenas com dados numéricos, possam interpretar corretamente a variável categórica *service*, mantendo a correspondência entre cada serviço original e seu valor numérico equivalente.

Figura 8 – Treinamento do modelo de detecção de anomalia

```
# Define as colunas de entrada (features) para treinar o modelo
# As colunas representam hora, minuto, dia da semana, serviço codificado e duração
X = df[["hora", "minuto", "dia_semana", "service_encoded", "duracao"]]

# Cria uma instância do IsolationForest com 1% de contaminação (anomalias esperadas)
# e uma semente fixa (42) para garantir reprodutibilidade na geração de números aleatórios
modelo = IsolationForest(contamination=0.01, random_state=42)

# Treina o modelo com os dados de entrada X
modelo.fit(X)
```

Fonte: AUTOR, 2025

O trecho de código apresentado na Figura 8 – Treinamento do modelo de detecção de anomalia tem como objetivo preparar os dados de entrada e treinar o modelo de detecção de anomalias utilizando o algoritmo *Isolation Forest*. Primeiramente, são selecionadas as colunas

do DataFrame que servirão como variáveis de entrada (características) para o modelo, incluindo hora, minuto, dia da semana, serviço codificado e duração da operação.

Em seguida, é criada uma instância do Isolation Forest, configurada para considerar aproximadamente 1% dos registros como anomalias esperadas (contamination=0.01) e com uma semente fixa (random_state=42) para garantir que os resultados sejam reproduzíveis. A semente fixa controla a geração de números aleatórios, permitindo que o modelo produza sempre os mesmos resultados ao ser executado novamente. Com isso, o modelo é treinado com os dados de entrada, aprendendo a distinguir comportamentos normais de potenciais anomalias. Após essa etapa, ele estará pronto para ser aplicado em novos registros, permitindo identificar automaticamente eventos atípicos dentro do fluxo de logs do Pix.

Figura 9 – Salva o modelo treinado

```
# Salva o modelo treinado no arquivo "modelo_duracao.joblib"
dump(modelo, "modelo_duracao.joblib")

# Salva o codificador de labels no arquivo "label_encoder_service.joblib"
dump(le_service, "label_encoder_service.joblib")
```

Fonte: AUTOR, 2025

A etapa final conforme Figura 9 – Salva o modelo treinado consiste em salvar o modelo treinado e os artefatos necessários para utilização futura. Persistindo o modelo em um arquivo chamado "modelo_duracao.joblib", juntamente com o arquivo “label_encoder_service.joblib” com a codificação dos rótulos (*labels*) utilizada para transformar os nomes dos serviços em valores numéricos (LabelEncoder).

b) Aplicação do Modelo (Rosa – Etapa 2)

Com o modelo criado “modelo_duracao.joblib e label_encoder_service.joblib”, é possível utilizá-lo no fluxo atual de logs, enviando uma cópia dos logs para a aplicação com o modelo criado, para análise e identificação de anomalias, conforme visto na Figura 3 – Ambiente Big Data do Pix com introdução de IA.

Para isso, é necessário a criação de uma nova saída no logstash, e uma aplicação para receber as cópias dos logs e aplicar o modelo treinado. Segue detalhes das duas etapas.

Na Cópia dos Logs, o Logstash se conecta ao kafka e extrai a cópia dos logs para enviar via REST para a aplicação desenvolvida especificamente para aplicar o modelo de detecção de anomalias criado na etapa anterior, conforme configuração da Figura 10 – Configuração do logstash para extração e envio dos logs para a aplicação.

Figura 10 – Configuração do logstash para extração e envio dos logs para a aplicação

```
input {
  kafka {
    bootstrap_servers => "kafp2001:9092,kafp2002:9092,kafp2003:9092"
    codec => json
    topics => "mon-pix"
    consumer_threads => 3
    group_id => "ai-validation"
    auto_commit_interval_ms => "5000"
    client_id => 'ai-validation'
  }
}

filter {
  if ([service] and [operacao] == "CREDITO" and [duracao]) {
    mutate {
      remove_field => [
        "instOrigem", "instDestino", "short_message", "source_host",
        "id", "message", "full_message", "host",
        "@version", "tags", "type", "source", "gl2_*", "facility",
        "file", "stack", "environment", "valor", "descLevel", "method",
        "prioridade", "relativePath", "level", "line", "pos", "version",
        "agendado", "boolIntercredis", "idRequisicaoPagamento", "tipoQrcode"
      ]
    }
  } else {
    drop { }
  }
}

output {
  http {
    url => "http://aisp2001/validation"
    http_method => "post"
    format => "json"
    content_type => "application/json"
  }
}
```

Fonte: AUTOR, 2025

A configuração do Logstash apresentada conecta ao Kafka formado pelos brokers kafp2001:9092, kafp2002:9092 e kafp2003:9092, consumindo mensagens do tópico mon-Pix no formato JSON. O Logstash utiliza múltiplas threads de consumo para otimizar o processamento e o group_id identifica este consumidor dentro do cluster, para que ele continue lendo de onde parou. Durante a ingestão, apenas mensagens que possuam os campos service e duracao e cuja operação seja igual a "CREDITO" são processadas, enquanto os demais registros são descartados. Para as mensagens válidas, campos irrelevantes são removidos para reduzir o

volume de dados e simplificar o *dataset*. Após essa etapa de filtragem e sanitização, por fim, os registros são enviados para um *endpoint* HTTP via método POST, convertidos em JSON e com o tipo de conteúdo application/json, que se trata da aplicação que irá aplicar o modelo treinado.

Essa aplicação, utilizando o modelo treinado, avaliando individualmente cada registro de log, classificando-o como anômalo caso apresente características identificadas pelo modelo. A Figura 10 tem os detalhes do script utilizado, separado por etapas.

Figura 11 – Importação das bibliotecas necessárias

```
# Importa a classe Flask, o objeto request para acessar dados da requisição,
# e jsonify para retornar respostas em formato JSON
from flask import Flask, request, jsonify

# Importa a biblioteca pandas para manipulação de dados tabulares
import pandas as pd

# Importa o algoritmo de detecção de anomalias IsolationForest
from sklearn.ensemble import IsolationForest

# Importa o codificador de variáveis categóricas para inteiros
from sklearn.preprocessing import LabelEncoder

# Importa a função load da biblioteca joblib para carregar objetos salvos
from joblib import load

# Importa a biblioteca requests para envio de logs ao Graylog
import requests
```

Fonte: AUTOR, 2025

Esse bloco corresponde à importação das bibliotecas necessárias para a aplicação, conforme Figura 11 – Importação das bibliotecas necessárias. Ele traz bibliotecas do Flask para criar a API e tratar requisições HTTP, a biblioteca pandas para manipulação de dados tabulares, o IsolationForest para detecção de anomalias, o LabelEncoder para codificação de variáveis categóricas, a função load do joblib para carregar modelos treinados e a biblioteca requests para envio de logs ou alertas ao Graylog.

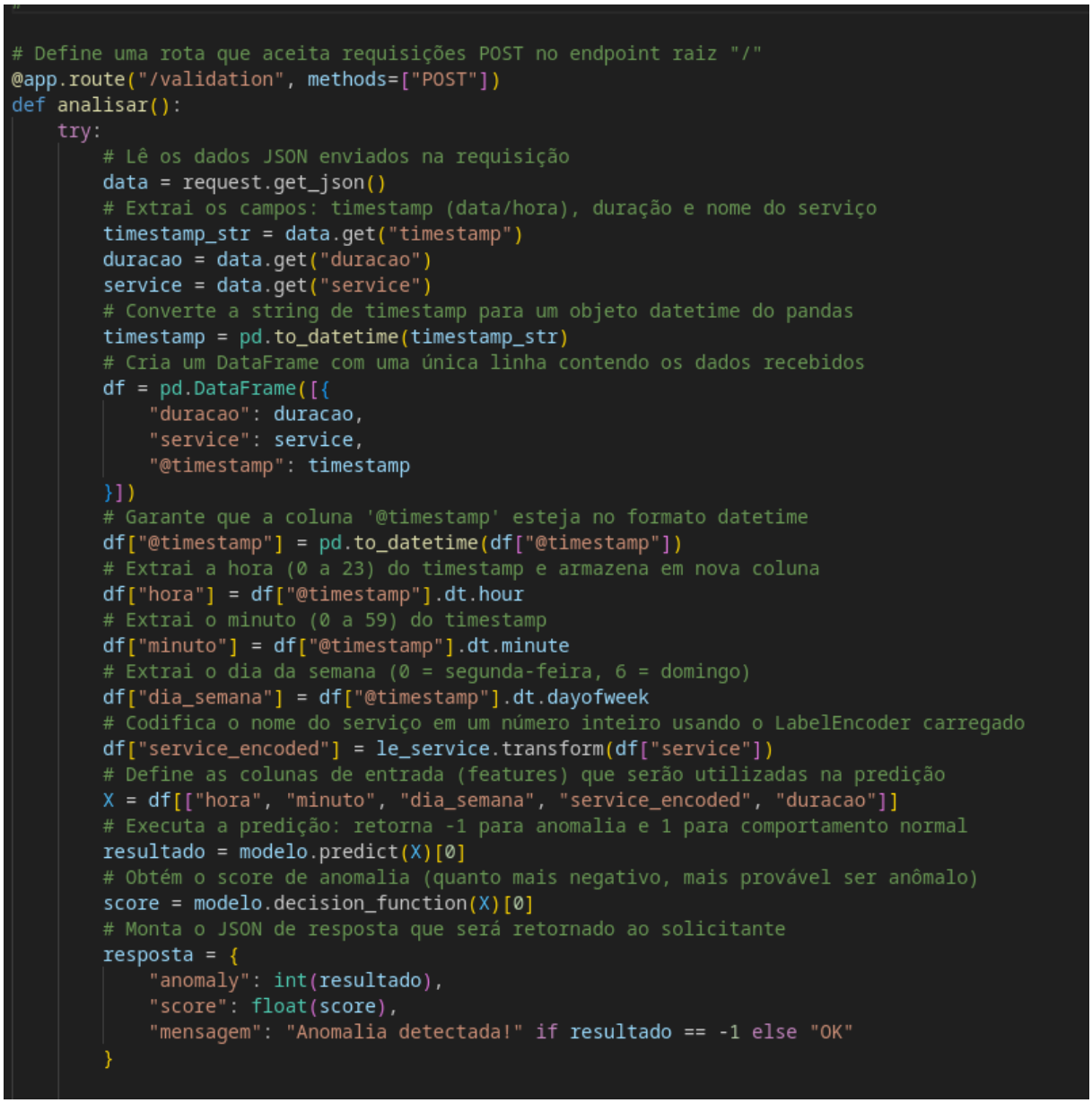
Figura 12 – Inicialização da aplicação Flask e carregamento dos modelos

```
# Cria uma instância da aplicação Flask
app = Flask(__name__)

# Carrega o modelo de detecção de anomalias previamente treinado
modelo = load("modelo_duracao.joblib")

# Carrega o codificador de serviços salvo (LabelEncoder)
le_service = load("label_encoder_service.joblib")

# Define o endpoint do Graylog para envio de logs no formato GELF
GRAYLOG_URL = "http://grap2001:12201/gelf"  # substitua pelo seu endpoint
```

Fonte: AUTOR, 2025

Esse trecho inicializa a aplicação e prepara os recursos necessários para análise de anomalias, de acordo com a Figura 12 – Inicialização da aplicação Flask e carregamento dos modelos. Primeiro, é criada uma instância do Flask, que permitirá expor a API para receber requisições. Em seguida, carrega-se o modelo de detecção de anomalias previamente treinado e armazenado no arquivo modelo_duracao.joblib juntamente com o LabelEncoder utilizado para codificar os nomes de serviços (le_service). Por fim, é definido o endpoint do Graylog, que será usado posteriormente para enviar logs ou alertas no formato GELF quando forem detectadas anomalias.

Figura 13 – Configuração inicial e análise de anomalias



```python
# Define uma rota que aceita requisições POST no endpoint raiz "/"
@app.route("/validation", methods=["POST"])
def analisar():
    try:
        # Lê os dados JSON enviados na requisição
        data = request.get_json()
        # Extrai os campos: timestamp (data/hora), duração e nome do serviço
        timestamp_str = data.get("timestamp")
        duracao = data.get("duracao")
        service = data.get("service")
        # Converte a string de timestamp para um objeto datetime do pandas
        timestamp = pd.to_datetime(timestamp_str)
        # Cria um DataFrame com uma única linha contendo os dados recebidos
        df = pd.DataFrame([{
            "duracao": duracao,
            "service": service,
            "@timestamp": timestamp
        }])
        # Garante que a coluna '@timestamp' esteja no formato datetime
        df["@timestamp"] = pd.to_datetime(df["@timestamp"])
        # Extrai a hora (0 a 23) do timestamp e armazena em nova coluna
        df["hora"] = df["@timestamp"].dt.hour
        # Extrai o minuto (0 a 59) do timestamp
        df["minuto"] = df["@timestamp"].dt.minute
        # Extrai o dia da semana (0 = segunda-feira, 6 = domingo)
        df["dia_semana"] = df["@timestamp"].dt.dayofweek
        # Codifica o nome do serviço em um número inteiro usando o LabelEncoder carregado
        df["service_encoded"] = le_service.transform(df["service"])
        # Define as colunas de entrada (features) que serão utilizadas na predição
        X = df[["hora", "minuto", "dia_semana", "service_encoded", "duracao"]]
        # Executa a predição: retorna -1 para anomalia e 1 para comportamento normal
        resultado = modelo.predict(X)[0]
        # Obtém o score de anomalia (quanto mais negativo, mais provável ser anômalo)
        score = modelo.decision_function(X)[0]
        # Monta o JSON de resposta que será retornado ao solicitante
        resposta = {
            "anomaly": int(resultado),
            "score": float(score),
            "mensagem": "Anomalia detectada!" if resultado == -1 else "OK"
        }
```

Fonte: AUTOR, 2025

Esse trecho define a rota /validation da aplicação Flask, responsável por receber requisições POST com dados de logs e executar a detecção de anomalias, representado pela Figura 13 – Configuração inicial e análise de anomalias.

Figura 14 – Envio de alertas para Graylog e tratamento de erros

```
    # Se o resultado indicar anomalia, envia os dados para o Graylog
    if resultado == -1:
        # Monta a mensagem no formato GELF
        gelf_msg = {
            "version": "1.1",  # versão do GELF
            "host": "pix-app",  # host que envia o log
            "short_message": "Anomalia detectada no fluxo Pix",  # mensagem resumida
            "timestamp": timestamp.timestamp(),  # timestamp Unix
            "level": 3,  # nível de erro
            "duracao": duracao,  # inclui duração da transação
            "service": service,  # serviço envolvido
            "score": float(score)  # score de anomalia
        }
        # Tenta enviar a mensagem ao Graylog, ignorando falhas para não interromper o serviço
        try:
            requests.post(GRAYLOG_URL, json=gelf_msg, timeout=2)
        except Exception as e:
            print(f"Falha ao enviar log para Graylog: {e}")

    # Retorna a resposta ao solicitante
    return jsonify(resposta)

except Exception as e:
    # Retorna erro 400 e mensagem descritiva em caso de falha na requisição ou predição
    return jsonify({"erro": str(e)}), 400
```

Fonte: AUTOR, 2025

Os resultados dessa análise são encaminhados para o Graylog, armazenados no Elasticsearch e, por fim, disponibilizados para visualização no Grafana, conforme Figura 14 – Envio de alertas para Graylog e tratamento de erros. Dessa forma, é possível criar alertas e gráficos que representam as anomalias detectadas em tempo real. Exemplo de detecção de anomalia: às 17h (horário de maior movimento do Pix nas sextas-feiras, o tempo de resposta do Pix aumenta para 800ms. O modelo treinado imediatamente detecta que esse tempo foge completamente do padrão esperado. Com base na anomalia detectada e nos dados coletados, o modelo gera um *alerta crítico* com qual serviço está sendo impactado e o tipo de operação. A Figura 15 – Inicialização da aplicação demonstra como iniciar o servidor Flask em modo debug se algum arquivo for executado diretamente.

Figura 15 – Inicialização da aplicação

```
# Inicia o servidor Flask em modo debug se este arquivo for executado diretamente
if __name__ == "__main__":
    app.run(debug=True)
```

Fonte: AUTOR, 2025

Em resumo do processo para criação do modelo de IA do Pix do Sicoob envolverá as fases: criação de servidores com o modelo treinado; interceptação dos logs oriundos dos serviços Pix; utilização do modelo para detectar anomalias; geração de alertas para o Graylog; monitoração por meio do Grafana e tomada de ações para evitar quedas do Pix; e atualização periódica do modelo.

A integração dessa etapa com o restante da arquitetura atual garante que os alertas de anomalia estejam disponíveis nas mesmas ferramentas já utilizadas pelas equipes técnicas, como o Graylog, Elasticsearch e o Grafana, sem a necessidade de plataformas adicionais e evitando a adoção de soluções paralelas, que implicariam na contratação de novas pessoas e no treinamento das equipes atuais para uma possível nova solução. Além disso, o envio dessas anomalias para o Elasticsearch, que já armazena os logs operacionais, permite cruzar as informações com outros indicadores do ambiente, enriquecendo a análise e contribuindo para decisões mais assertivas.

Com esse tipo de abordagem, torna-se possível antecipar falhas ou degradações no serviço antes mesmo que os usuários percebam, ampliando significativamente a capacidade de resposta da equipe técnica e também contribui para uma maior confiabilidade da arquitetura como um todo, reduzindo o MTBF e MTTR de incidentes relacionados ao Pix.

Além de permitir análises em tempo real e o acionamento de alertas preventivos, o modelo também poderá ser utilizado como base para processos automatizados de recuperação de serviços (*self-healing*). Nesses casos, a detecção de uma anomalia poderia servir como gatilho para uma ação corretiva imediata, como o reinício automático de um serviço ou a redistribuição de carga entre instâncias. Esse tipo de automação reduz o tempo de indisponibilidade e o esforço humano necessário para resolução de falhas operacionais.

3.5 Estratégia de montagem do *dataset* e desenvolvimento do modelo de IA

O processo começou pela montagem do *dataset* inicial em ambiente de desenvolvimento e testes. A escolha deste formato permitiu a criação de um arquivo com todos os dados necessários para o treinamento em um único ponto, evitando o aumento da complexidade no processo de treinamento, quando envolvendo arquivos pulverizados em vários locais. O *dataset* criado não poderia ser utilizado de forma bruta, pois apresentou ruídos, inconsistências e informações redundantes. Por esse motivo, a primeira etapa foi a limpeza dos dados. Essa limpeza se fez necessária porque qualquer dado contaminado impactaria a qualidade do modelo, gerando resultados distorcidos ou irrelevantes. Além disso, o formato

escolhido para representar os dados foi pensado para padronizar as informações e facilitar tanto o processamento quanto a interpretação futura. O objetivo era garantir que os dados estivessem organizados, consistentes e prontos para as próximas etapas.

A descoberta de contaminações no conjunto de dados não ocorreu de imediato. Durante a exploração inicial, percebeu-se que existiam registros repetidos, inconsistentes e até fora de contexto. Essa identificação do problema foi realizada através de análises estatísticas simples, como verificar manualmente tempos acima da média, e verificações manuais, que mostraram a necessidade de aplicar regras de pré-processamento antes de qualquer tentativa de modelagem. Assim, a limpeza e a transformação do *dataset* não foram apenas passos formais, mas escolhas conscientes para aumentar a confiabilidade de todo o processo.

Na sequência, surgiram as etapas de treinamento e aplicação do modelo dentro do fluxo já existente. A inclusão dessas etapas foi motivada pela necessidade de transformar o fluxo de big data existente, em algo preditivo e acionável. Treinar o modelo significou ensinar a máquina a reconhecer padrões a partir de exemplos históricos, enquanto a aplicação permitiu validar esses aprendizados em uma situação real.

O motivo para seguir esse caminho, em vez de optar por soluções de mercado já prontas, esteve relacionado a dois fatores principais que são adequação e controle. Inicialmente, foi pensada a utilização de uma solução pronta de mercado como o Dynatrace Davis, mas uma solução pronta funciona muitas vezes como “caixa preta”, entregando resultados rápidos, mas limitando quanto a flexibilidade e a transparência do processo de treinamento e aplicabilidade. Nesse trabalho, era essencial compreender cada decisão tomada pelo modelo e ajustar de forma precisa os parâmetros em consonância com as particularidades do contexto. Desenvolver internamente essas etapas possibilitou um maior controle sobre o fluxo e assegurou que as decisões fossem fundamentadas no comportamento real dos dados, e não em generalizações impostas por plataformas externas. Além disso, a personalização da solução trouxe vantagens como a escalabilidade do processo e a possibilidade de expandir a solução para novos cenários sem depender de licenças, restrições técnicas ou custos adicionais das ferramentas de mercado.

Após optar pelo desenvolvimento de um modelo local, foi iniciada a análise de soluções alternativas, sendo escolhido o Scikit-learn, pois trata-se de uma das bibliotecas mais consolidadas e reconhecidas no ecossistema Python para aprendizado de máquina, sendo amplamente utilizada tanto em projetos acadêmicos quanto em aplicações de mercado. Sua principal característica é a simplicidade de uso juntamente com a robustez. Ele oferece uma ampla lista de algoritmos de Aprendizagem de Máquinas já prontos, como bibliotecas para

modelos lineares, regressão e classificação, árvores de decisão etc. Isso elimina a necessidade de reimplementar algoritmos complexos, economizando tempo e reduzindo riscos de erro.

Outro fator importante é a padronização da API, que torna o fluxo de testes muito mais ágil. Durante o desenvolvimento é possível trocar de um modelo para outro com poucas linhas de código, mantendo mesmo padrão de chamadas para treino, validação e predição. Por este motivo, o Scikit-learn se integra de forma nativa com outras ferramentas fundamentais no projeto, como NumPy e pandas, o que facilita o pré-processamento, a manipulação de dados e a análise dos resultados.

A escolha do Scikit-learn também se justifica pela vasta documentação e comunidade ativa, que oferecem suporte, exemplos práticos e boas práticas de forma pública, contribuindo para os trabalhos de desenvolvimento e aprendizagem deste trabalho, já que eventuais problemas ou dúvidas podem ser rapidamente resolvidos. Quando comparado a alternativas mais pesadas e complexas, como TensorFlow ou PyTorch, o Scikit-learn é extremamente eficiente para prototipagem e implementação de modelos tradicionais, principalmente quando não há necessidade de redes neurais profundas ou processamento em larga escala com GPUs. No contexto de um projeto que busca equilíbrio entre simplicidade, confiabilidade e desempenho, o Scikit-learn se mostra a escolha natural.

Adicionalmente, a adoção do Scikit-Learn em contextos reais reforçou sua robustez. Por exemplo, o LinkedIn implementou uma versão distribuída do Scikit-learn com Isolation Forest em Spark/Scala para lidar com grandes quantidades de dados e abusos na plataforma, como contas falsas e raspagem automatizada. Outros casos incluem detecção de fraudes em transações financeiras e monitoramento de servidores, onde o método identifica padrões fora do normal e evita falhas antes que afetem a experiência do usuário.

No caso da etapa de treinamento, a escolha do Python como linguagem de desenvolvimento está diretamente ligada à sua simplicidade e ao vasto ecossistema de bibliotecas voltadas para ciência de dados e aprendizado de máquina sendo a principal para este trabalho a biblioteca Scikit-learn já mencionada. O Python também é utilizado massivamente por grandes empresas, sendo considerada praticamente a linguagem padrão para ciência de dados. Ele permite que o desenvolvimento seja mais rápido, que os protótipos se tornem funcionais em menos tempo e que a integração entre diferentes etapas, desde a manipulação dos dados até a exposição do modelo, seja feita de forma fluida. Outras linguagens poderiam ser utilizadas, como R ou Java, mas dificilmente alcançariam a mesma combinação de curva de aprendizado suave, ampla comunidade de suporte e ferramentas integradas.

Dentro desse ecossistema, o NumPy utilizado nesse projeto, se destaca como a base matemática que sustenta operações vetoriais e matriciais de alta performance. Ele é a base de muitas outras bibliotecas, e sua utilização garante eficiência no tratamento de dados numéricos. Poderíamos utilizar estruturas mais simples como listas nativas da linguagem, mas isso comprometeria a performance e inviabilizaria operações complexas, pois seria necessário um trabalho de desenvolvimento extensivo para igualar o desempenho do NumPy.

Referente ao Pandas, indispensável para manipulação e análise de dados tabulares, oferece estruturas como DataFrames, que permitem organizar e transformar grandes volumes de informação de forma e eficiente. Sem o Pandas, essas mesmas operações exigiriam código muito mais verboso e propenso a erros, o que poderia impactar diretamente o desenvolvimento deste trabalho, pois trata-se de um modelo treinado com uma massa de dados razoável, e na eventual expansão deste trabalho para outros componentes da empresa, a falta do Pandas poderia criar um gargalo já que hoje movimentamos cerca de 4 terabytes de logs por dia.

No desenho da solução, a separação entre o fluxo de treinamento e o fluxo de execução do modelo é fundamental. Isso ocorre porque o treinamento exige recursos mais intensivos, maior volume de dados e ajustes constantes de parâmetros, enquanto a execução deve ser leve, ágil e escalável para responder a requisições em tempo real. Unir os dois processos em um único fluxo criaria sobrecarga, dificultaria a manutenção e reduziria a eficiência operacional.

Finalmente, a escolha pelo Flask como camada de exposição do modelo se justifica pela sua leveza e pela facilidade de integração em cenários de APIs REST. Ele é minimalista o suficiente para não adicionar complexidade desnecessária, mas ao mesmo tempo robusto para atender às necessidades de servir previsões de forma confiável. Alternativas, como frameworks mais completos, poderiam ser utilizadas, mas acabariam trazendo peso e complexidade sem agregar valor real no contexto específico deste projeto, que demanda justamente simplicidade e agilidade na execução do modelo em produção.

3.6 Validação do modelo proposto – medição do sucesso

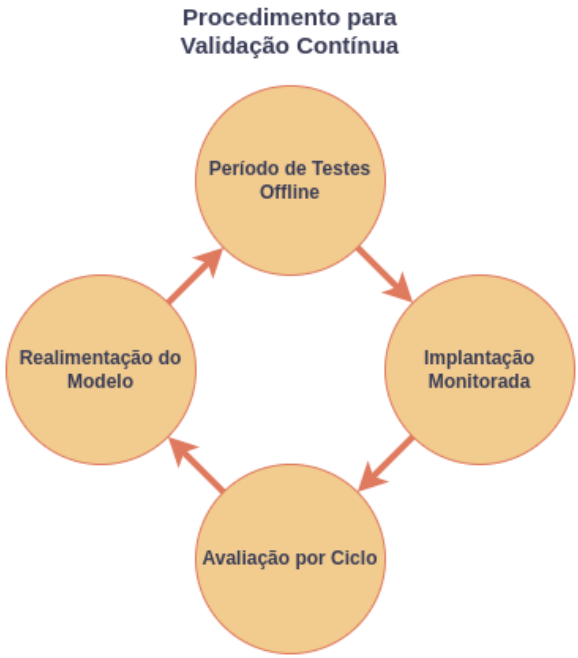
A validação do modelo visa avaliar a eficácia da detecção de anomalias que possam indicar falhas iminentes ou degradação no serviço Pix. Essa avaliação será feita com base em: métricas estatísticas tradicionais de ML, as KPIs que afetam a resiliência cibernética e monitoramento em tempo real integrado a painéis de monitoração.

No contexto do Sicoob e do Pix, a validação deverá considerar 3 indicadores operacionais. O primeiro indicador é composto do MTBF e MTTR, onde o MTBF mede o

tempo médio entre falhas no ambiente Pix e o MTTR mede o tempo médio de recuperação após uma falha nesse ambiente. O impacto do modelo é medido observando se há aumento do MTBF e redução do MTTR após implantação do sistema de detecção. O segundo indicador é o SLA de Disponibilidade para avaliar se há redução no tempo de indisponibilidade. Exemplo: redução de incidentes com tempos de resposta acima de 500ms. O terceiro e último indicador é a Taxa de Alertas Úteis versus Falsos Positivos, para diferenciar os alertas úteis, que são aqueles que resultaram em ação corretiva e falsos positivos, que são os alertas que não indicaram falhas reais.

Para que o modelo possa ser colocado em produção com confiança, é proposto um procedimento de validação contínua, conforme Figura 16 – Procedimento de validação contínua.

Figura 16 – Procedimento de validação contínua



Fonte: AUTOR, 2025.

O Período de Testes Offline é para avaliar o modelo com dados históricos rotulados, cujo objetivo é medir a acurácia e precisão do modelo sem impactar o ambiente de produção, ajustando parâmetros e identificando padrões recorrentes de anomalias. Depois, a Implantação Monitorada serve para colocar o modelo em produção com paralelismo ao sistema atual. Nesta etapa, o modelo roda em tempo real, mas seus resultados não afetam diretamente os alertas ou decisões operacionais. Os logs continuam sendo processados pela abordagem tradicional,

permitindo comparar os diagnósticos e refinar o modelo antes da implementação oficial em produção. Na sequência, a Avaliação por Ciclo, que prevê revisar semanalmente métricas de alerta, desempenho e KPIs. Essa avaliação cíclica envolve a análise conjunta da equipe técnica e de negócio, com foco em ajustar limiares, identificar falsos positivos/negativos e garantir que o modelo esteja agregando valor na detecção precoce de incidentes ou comportamentos atípicos. E, por fim, a Realimentação do Modelo que é o retreinamento periódico com novos dados e feedback dos times técnicos. A cada ciclo de avaliação, os dados mais recentes e os feedbacks nos times são incorporados ao pipeline de treinamento. Isso permite que o modelo evolua com o ambiente, mantendo sua eficácia diante de mudanças nos padrões de comportamento dos sistemas monitorados.

Essa abordagem de validação contínua permite que o modelo evolua junto com o ambiente, mantendo sua eficácia. A implementação em produção só ocorrerá após a conclusão dessas etapas, garantindo que os alertas de anomalia sejam gerados, o que aumenta a confiança na solução. A detecção de anomalias permitirá que a equipe técnica seja mais proativa, antecipando falhas antes que os usuários as percebam, o que contribui para uma maior confiabilidade e resiliência cibernética da solução Pix.

Com base nessas etapas, a implementação do modelo pode ser considerada confiável, mas o processo contínuo de validação e refinamento é crucial para manter a eficácia da solução a longo prazo.

4 Resultados

Este capítulo detalha a avaliação experimental do modelo de detecção de anomalias proposto. O objetivo é demonstrar a eficácia do modelo em identificar comportamentos atípicos nas transações do Pix e, assim, validar a sua aplicação como uma ferramenta proativa para aumentar a resiliência cibernética do ambiente do Sicoob. A análise está dividida em três partes que são a preparação do ambiente, a aplicação do modelo e uma análise aprofundada dos resultados obtidos.

Antes de iniciar a explicação das partes envolvidas, explicamos que a avaliação é o processo de aplicar o modelo treinado em um *dataset* diferente do *dataset* usado no treinamento, sendo possível verificar visualmente a qualidade do resultado e identificar possíveis erros na detecção de anomalias. O modelo pode ser entendido como uma espécie de "ferramenta de aprendizado", criada a partir dos dados de treinamento, que aprende padrões e tenta usá-los para analisar novos dados. O modelo mencionado foi criado através do processo já descrito neste trabalho, sendo este capítulo focado na análise e apresentação dos resultados obtidos após o treinamento.

A primeira parte do processo é a preparação do ambiente de experimentação, onde foi necessária a extração de um novo *dataset*, para assegurar que a avaliação fosse realizada com informações recentes de transações Pix do ambiente de testes do Sicoob. Para criação de um novo *dataset* foram extraídos novos logs do PIX durante um período de 15 minutos utilizando o Logstash. A partir dessa extração, seguindo o mesmo padrão de entrada, filtragem e saída descritos anteriormente, foi possível gerar um novo *dataset*, contendo os mesmos campos usados no desenvolvimento e treinamento de dados, mas agora com informações recentes, e atendendo aos requisitos do modelo.

Importante ressaltar que o Logstash utilizado neste procedimento é o mesmo existente no fluxo atual de logs do PIX, ou seja, o fluxo permanece inalterado, sendo criada apenas uma cópia dos dados existentes para o *dataset*, objetivando a menor modificação possível no ambiente. Para essa ação, foi necessária a criação de um novo arquivo de configuração no Logstash, que realiza as etapas de entrada, filtragem e saída.

A Figura 17 – Exemplo de arquivo de configuração, apresenta o exemplo do arquivo de configuração utilizado no Logstash para a geração do dataset, que será empregado neste capítulo para avaliação dos resultados do modelo.

Figura 17 – Exemplo de arquivo de configuração

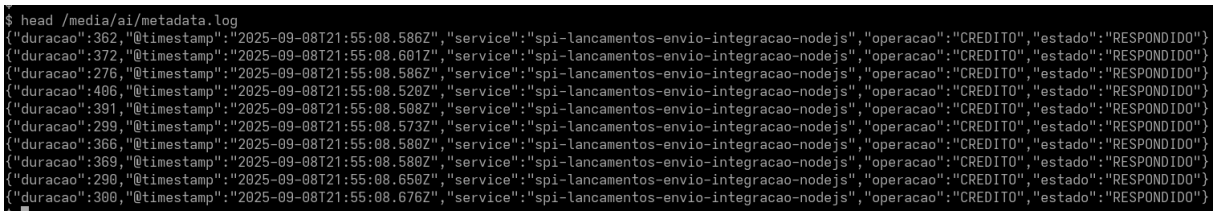
AI-PIX > Scripts > avaliacao.conf

```
input {
  kafka {
    bootstrap_servers => "kafp2001:9092,kafp2002:9092,kafp2003:9092"
    codec => json
    topics => "mon-pix"
    consumer_threads => 3
    group_id => "ai-training"
    auto_commit_interval_ms => "5000"
    client_id => 'ai-training'
  }
}


filter {
  if ([service] and [operacao] == "CREDITO" and [duracao]) {
    mutate {
      remove_field => [
        "instOrigem", "instDestino", "short_message", "source_host",
        "id", "message", "full_message", "host",
        "@version", "tags", "type", "source", "gl2_*", "facility",
        "file", "stack", "environment", "valor", "descLevel", "method",
        "prioridade", "relativePath", "level", "line", "pos", "version",
        "agendado", "boolIntercredis", "idRequisicaoPagamento", "tipoQrcode"
      ]
    }
  } else {
    drop { }
  }
}

output {
  file {
    path => "/media/ai/metadata.log"
    codec => json_lines
  }
}
```

Fonte: AUTOR, 2025.

Podemos ver que o Logstash se conecta ao Kafka, acessando especificamente a fila responsável pela monitoração do PIX. A partir dessa conexão, os dados recebidos passaram por um processo de filtragem, onde os campos considerados desnecessários foram removidos no passo do “filter”. Esse processo garante que apenas as informações realmente úteis sejam mantidas, facilitando a análise e reduzindo o volume de dados a serem processados. Por fim, o resultado desse fluxo é armazenado localmente no *dataset* “metadata.log”, que reúne os registros já tratados e prontos para a aplicação do modelo, conforme Figura 18 – Registros tratados.

Figura 18 – Registros tratados



Fonte: AUTOR, 2025.

No *dataset* em questão, é possível visualizar todos os campos necessários para a avaliação do modelo. Entre eles estão: “duracao”, que indica o tempo total da operação; “estado”, representando o status da execução; “operação”, que identifica o tipo de operação realizada; “service”, correspondente ao serviço em questão; e “timestamp”, que registra a data e a hora da execução. A figura 19 – Formato do dataset mostra com mais detalhes o formato do dataset.

Figura 19 – Formato do dataset

Variável	Descrição	Tipo de Dado
Duracao	Tempo total de execução da operação em milissegundos (ms).	Long: ou seja, numérico até 64 bits.
@timestamp	Data e hora da execução.	Timestamp: armazena uma combinação de data e hora que pode alcançar precisão de frações de segundos.
service	Nome do micro serviço que recebe a requisição.	String: representa uma sequência de caracteres, incluindo letras, números, símbolos e espaços.
operacao	Tipo da operação, como crédito ou débito.	String
Estado	Estado da Operação, como por exemplo RESPONDIDO.	String

Fonte: AUTOR, 2025.

Objetivando apenas o mapeamento dos resultados, o metadado criado foi copiado para um servidor diferente do utilizado pelo Logstash, onde serão realizados os trabalhos deste tópico.

Com o novo *dataset* produzido, passamos para segunda parte da experimentação onde aplicamos o modelo que foi treinado utilizando os mesmos scripts em Python, com Scikit-learn e Isolation Forest. A diferença é que, nesta etapa, os resultados são salvos em um arquivo formato CSV, que servirá como base para a avaliação do resultado do modelo. Os dados não serão enviados para o Graylog, pois nesta etapa, o Graylog apenas armazena e mostra os dados, não permitindo a composição dos resultados de forma a comprovar a qualidade dos resultados, sendo melhor tratado localmente, utilizando a estação de trabalho para compor e analisar os dados.

A partir do arquivo gerado, é comparado os resultados obtidos com os dados atualizados, analisado possíveis padrões e identificado qualquer discrepância em relação ao comportamento esperado, garantindo uma avaliação precisa e consistente do modelo.

A figura 20 – Script simplificado para ler *dataset* mostra o script simplificado, utilizado para ler o dataset, aplicar o modelo treinado com scikit-learn com Isolation Forest e salvar o resultado em um novo CSV com nome “resultados_com_anomalia.csv”.

Figura 20 – Script simplificado para ler o dataset

AI-PIX > Scripts > 02-verificar-qualidade-do-modelo.py > ...

```python
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import LabelEncoder
from joblib import load
from datetime import datetime
import json

with open("metadata.log", "r") as f:
    linhas = [json.loads(linha) for linha in f if linha.strip()]

df = pd.DataFrame(linhas)
df["@timestamp"] = pd.to_datetime(df["@timestamp"])
df["hora"] = df["@timestamp"].dt.hour
df["minuto"] = df["@timestamp"].dt.minute
df["dia_semana"] = df["@timestamp"].dt.dayofweek

modelo = load("modelo_duracao.joblib")
le_service = load("label_encoder_service.joblib")

df["service_encoded"] = le_service.transform(df["service"])
X = df[["hora", "minuto", "dia_semana", "service_encoded", "duracao"]]
df["anomaly"] = modelo.predict(X)

df.to_csv("resultados_com_anomalia.csv", index=False)
```

Fonte: AUTOR, 2025.

Destaque para as linhas 1 a 6, onde são importadas as bibliotecas utilizadas, em especial Scikit-learn e Isolation Forest. Na linha 8, ocorre a importação do *dataset* criado. Já nas linhas 17 e 18, o modelo previamente treinado é carregado. Na linha 22, o modelo é aplicado aos dados, e, por fim, na linha 24, o resultado é salvo em um arquivo CSV, consolidando os dados gerados pela aplicação do modelo.

Editando o arquivo “resultados_com_anomalia.csv”, conforme apresentado na Figura 21 – Principais campos gerados pelo modelo, é possível visualizar os principais campos gerados pelo modelo, sendo que o campo anomaly é o foco da análise, indicando a detecção de anomalias com o valor -1 e o comportamento considerado normal com o valor 1.

Figura 21 – Principais campos gerados pelo modelo

	A	B	C	D	E	F	G	H	I	J	K
1	estado	tipoQrcode	@timestamp	operacao	service	duracao	hora	minuto	dia_semana	service_encoded	anomaly
2	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	656	21	11	6	1	1
3	FINALIZADO_SUCESSO		2025-09-08 21:55:08.601000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1164	21	11	6	1	-1
4	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	475	21	11	6	1	1
5	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.520000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	400	21	11	6	1	1
6	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.508000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1210	21	11	6	1	-1
7	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	620	21	11	6	1	1
8	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1394	21	11	6	1	-1
9	RESPONDIDO		2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	501	21	11	6	0	-1
10	RESPONDIDO		2025-09-08 21:55:08.650000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	628	21	11	6	0	-1
11	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.676000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	867	21	11	6	1	-1
12	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.539000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	847	21	11	6	1	-1
13	FINALIZADO_SUCESSO	ESTATICO	2025-09-08 21:55:08.568000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1059	21	11	6	1	-1
14	RESPONDIDO		2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	794	21	11	6	0	-1
15	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.615000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	483	21	11	6	1	1
16	FINALIZADO_SUCESSO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1031	21	11	6	1	-1
17	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.655000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1074	21	11	6	1	-1
18	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.715000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	532	21	11	6	1	1
19	LIQUIDADO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	494	21	11	6	1	1
20	RESPONDIDO		2025-09-08 21:55:08.672000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	592	21	11	6	0	-1
21	RESPONDIDO		2025-09-08 21:55:08.714000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	583	21	11	6	0	-1
22	RESPONDIDO		2025-09-08 21:55:08.762000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	611	21	11	6	0	-1
23	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.701000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	690	21	11	6	1	1
24	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1278	21	11	6	1	-1
25	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.751000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1293	21	11	6	1	-1
26	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.789000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1233	21	11	6	1	-1
27	RESPONDIDO		2025-09-08 21:55:08.739000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1186	21	11	6	0	-1
28	RESPONDIDO		2025-09-08 21:55:08.876000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1030	21	11	6	0	-1
29	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1301	21	11	6	1	-1
30	RESPONDIDO		2025-09-08 21:55:08.760000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	479	21	11	6	0	-1
31	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.877000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	441	21	11	6	1	1
32	LIQUIDADO		2025-09-08 21:55:08.871000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1094	21	11	6	1	-1
33	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.894000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	378	21	11	6	1	1
34	LIQUIDADO		2025-09-08 21:55:09.008000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	520	21	11	6	1	1
35	LIQUIDADO		2025-09-08 21:55:08.961000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1061	21	11	6	1	-1
36	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.963000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	629	21	11	6	1	1
37	RESPONDIDO		2025-09-08 21:55:08.990000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	909	21	11	6	0	-1
38	LIQUIDADO		2025-09-08 21:55:08.782000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	403	21	11	6	1	1
39	RESPONDIDO		2025-09-08 21:55:09.013000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	460	21	11	6	0	-1
40	LIQUIDADO	DINAMICO	2025-09-08 21:55:09.053000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	435	21	11	6	1	1
41	LIQUIDADO	ESTATICO	2025-09-08 21:55:09.047000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	565	21	11	6	1	1
42	RESPONDIDO		2025-09-08 21:55:08.937000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	709	21	11	6	0	-1
43	LIQUIDADO		2025-09-08 21:55:08.811000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	999	21	11	6	1	-1

Fonte: AUTOR, 2025.

A Figura 22 – Detalhes campos do arquivo demonstra melhor os campos do arquivo, trazendo a descrição, detalhando formatos e armazenamento desses campos.

Figura 22 – Detalhes campos do arquivo

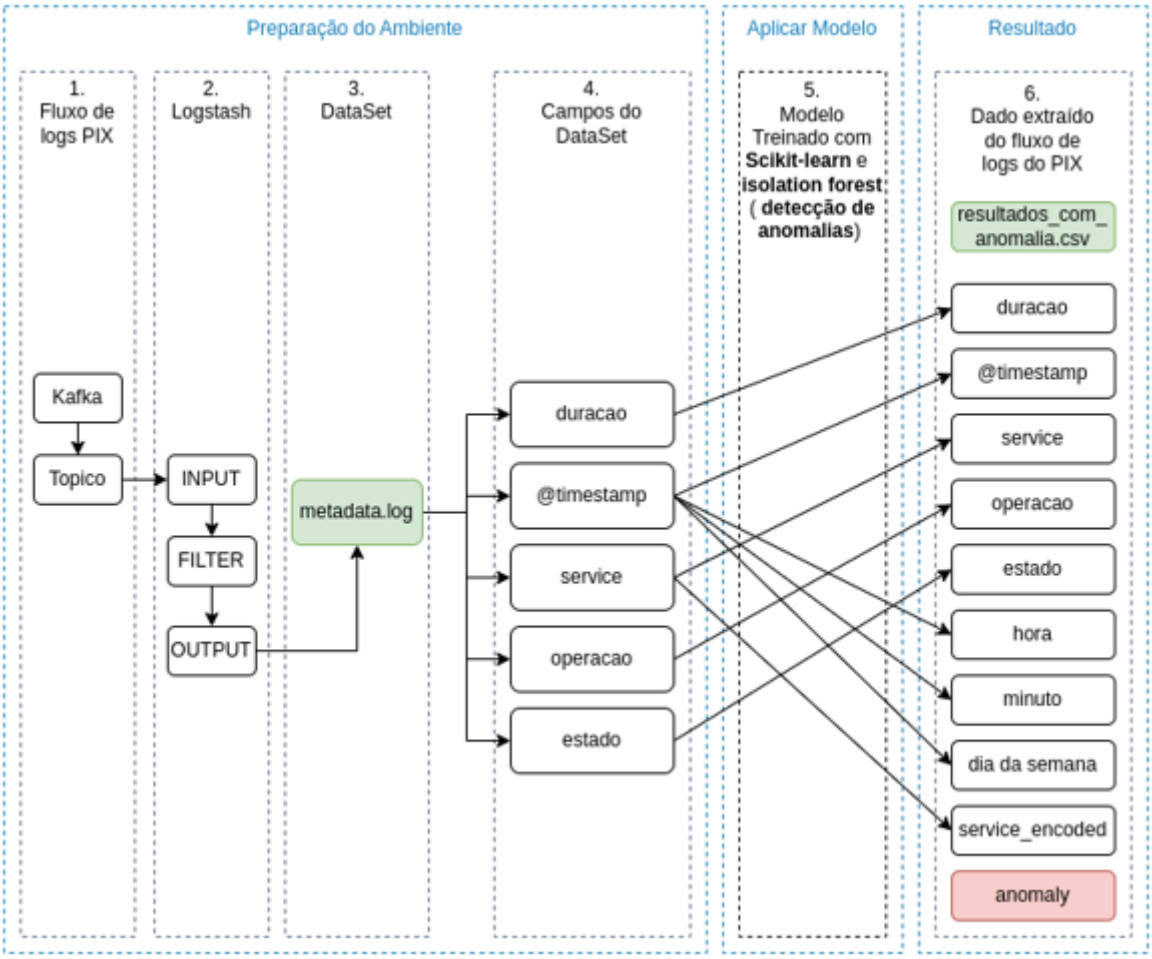
Variável	Descrição	Tipo de Dado
Duracao	Tempo total de execução da operação em milissegundos (ms).	Long: ou seja, numérico até 64 bits.
@timestamp	Data e hora da execução.	Timestamp: armazena uma combinação de data e hora que pode alcançar precisão de frações de segundos.
Service	Nome do micro serviço que recebe a requisição.	String: representa uma sequência de caracteres, incluindo letras, números, símbolos e espaços.
operacao	Tipo da operação, como crédito ou débito.	String.

Estado	Estado da Operação, como por exemplo RESPONDIDO.	String.
Hora	Horário em que o micro serviço respondeu à requisição.	Int: Número inteiro, positivo ou negativo.
minuto	Minuto em que o micro serviço respondeu à requisição.	Int: Número inteiro, positivo ou negativo.
dia da semana	Dia da semana em formato numérico.	Int: Número inteiro, positivo ou negativo, neste caso, os valores variam de 1 a 7, sendo o primeiro correspondente ao domingo, ou seja, o primeiro dia da semana, e o último correspondente ao sábado, como o último dia da semana.
service_encoded	Campo serviço codificado em um formato numérico, utilizado pelo Scikit-learn	Long: ou seja, numérico até 64 bits, que neste caso, representa o campo service no formato número, utilizado na conversão do Scikit-learn.
anomaly	campo que indica se um dado foi classificado como normal ou fora do padrão esperado.	Int: Número inteiro, positivo ou negativo, neste caso, representando 1 para normal e -1 para anomalia.

Fonte: AUTOR, 2025.

Para descrever melhor todo o processo aplicado, a Figura 23 – Implementação e etapas demonstra a implementação utilizada na análise dos resultados, contemplando as três etapas mencionadas: preparação do ambiente, aplicação do modelo e análise dos resultados.

Figura 23 – Implementação e etapas



Fonte: AUTOR, 2025.

Com o ambiente estruturado, foi possível identificar falhas e realizar as correções. Um exemplo claro ocorreu na primeira carga de análise, quando o resultado apresentou uma quantidade significativa de erros. Durante a investigação, constatou-se que a massa de dados utilizada no treinamento possuía um nível de contaminação acima do previsto, ou seja, continha uma proporção maior do que o esperado de serviços com tempo de resposta fora do normal. Essa condição comprometeu o processo, fazendo com que o modelo fosse treinado com parte dos dados incorretos.

Para solucionar esse problema, a margem de contaminação no processo de treinamento foi ajustada, alterando o parâmetro contamination para 1%. Essa alteração resultou em uma melhora significativa na acurácia do modelo. A partir desse ajuste, o modelo passou a identificar de forma mais confiável os serviços com tempo de resposta fora do normal,

reduzindo a ocorrência de falsos positivos e aumentando a acurácia dos resultados obtidos. A Figura 24 – Ajuste da margem da contaminação mostra o ajuste da margem de contaminação.

Figura 24 – Ajuste da margem da contaminação

```
# Cria uma instância do IsolationForest com 1% de contaminação (anomalias esperadas)
# e uma semente fixa (42) para garantir reprodutibilidade na geração de números aleatórios
modelo = IsolationForest(contamination=0.01, random_state=42)
```

Fonte: AUTOR, 2025.

Foi ainda identificado que o tamanho do modelo treinado era maior que a massa de dados utilizada no treino, situação que inviabiliza a utilização de uma IA para este tipo de trabalho. Após uma verificação mais aprofundada, constatou-se que não seria necessário realizar o treinamento do modelo utilizando as informações de segundos no campo timestamp. Em outras palavras, a análise poderia ser agrupada por hora e minuto. Para isso, foi necessária apenas a criação de três novos campos a partir do timestamp: hora, minuto e dia da semana. A Figura 25 – Criação de campos mostra essa criação.

Figura 25 – Criação de campos

```
38  # Converte a coluna '@timestamp' para o formato de data/hora reconhecido pelo pandas
39  df["@timestamp"] = pd.to_datetime(df["@timestamp"])
40
41  # Cria nova coluna "hora" contendo apenas a hora (0 a 23) extraída do timestamp
42  df["hora"] = df["@timestamp"].dt.hour
43
44  # Cria nova coluna "minuto" contendo apenas os minutos (0 a 59) do timestamp
45  df["minuto"] = df["@timestamp"].dt.minute
46
47  # Cria nova coluna "dia_semana" com o número do dia da semana (0 = segunda, 6 = domingo)
48  df["dia_semana"] = df["@timestamp"].dt.dayofweek
```

Fonte: AUTOR, 2025.

Com essa segregação, foi possível realizar um novo treinamento do modelo, no qual houve uma redução de 97% no tamanho em relação ao primeiro teste. Na Figura 26 – Execução do modelo é possível visualizar a execução do modelo, considerando os campos de treino sem o campo timestamp.

Figura 26 – Execução do modelo

```
63
64  # Define as colunas de entrada (features) para treinar o modelo
65  # As colunas representam hora, minuto, dia da semana, serviço codificado e duração
66  X = df[["hora", "minuto", "dia_semana", "service_encoded", "duracao"]]
67
68  # Cria uma instância do IsolationForest com 1% de contaminação (anomalias esperadas)
69  # e uma semente fixa (42) para garantir reprodutibilidade na geração de números aleatórios
70  modelo = IsolationForest(contamination=0.01, random_state=42)
71
72  # Treina o modelo com os dados de entrada X
73  modelo.fit(X)
```

Fonte: AUTOR, 2025.

Outro ponto identificado foi a incapacidade do Isolation Forest de interpretar campos não numéricos. Para resolver essa limitação, foi necessário aplicar um codificador de labels (LabelEncoder) no campo serviço. O Isolation Forest, assim como a maioria dos algoritmos de aprendizado de máquina do Scikit-learn, só consegue trabalhar com números e não entende valores em texto ou categorias diretamente. Por este motivo, o LabelEncoder atua como uma espécie de tradutor, convertendo cada valor em texto para um número. Os termos “CREDITO”, “DEBITO” e “TRANSFERENCIA” passaram a ser representados numericamente, permitindo que o algoritmo utilize essas informações em seus cálculos.

É importante destacar que essa conversão não significa que exista uma relação direta entre os valores, mas apenas que os dados ficam em um formato adequado para que a implementação do Isolation Forest os processe junto com os demais campos numéricos. A Figura 27 – Trecho do script mostra trecho do script com a conversão de dados.

Figura 27 – Trecho do script

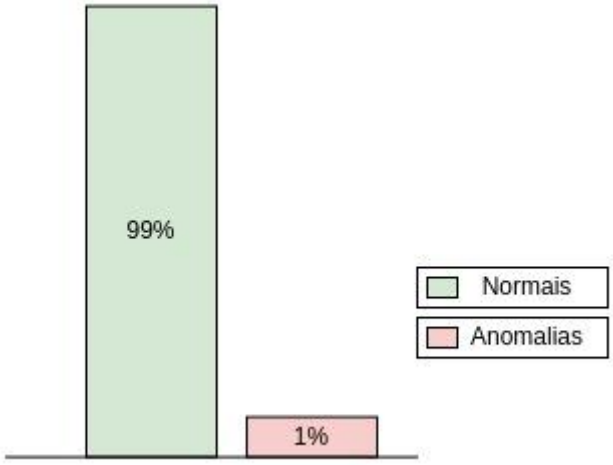
```
53
54  # Cria uma instância do codificador LabelEncoder
55  le_service = LabelEncoder()
56
57  # Codifica a coluna "service" (nomes de serviços) em números inteiros, e armazena em nova coluna
58  df["service_encoded"] = le_service.fit_transform(df["service"])
59
```

Fonte: AUTOR, 2025.

Finalizados todos os ajustes identificados durante os testes, passamos a verificar e analisar os resultados obtidos no ambiente. Nesta etapa, são avaliados o desempenho do modelo, a precisão na detecção de anomalias e a consistência dos dados processados, permitindo identificar eventuais pontos de melhoria e confirmar a eficácia das correções

realizadas. A Figura 28 – Distribuição das transações demonstra a distribuição das transações normais versus anomalias.

Figura 28 – Distribuição das transações



Fonte: AUTOR, 2025.

De imediato, um fato de destaque na avaliação é a comparação do resultado do modelo com o tempo de resposta da aplicação, detalhado no campo duracao. Este campo representa o tempo, em milissegundos, que a aplicação leva para processar cada transação. Ressaltamos ainda as informações sobre outros campos como tipo_qrcode, que identifica se a transação foi feita por um QR Code estático ou dinâmico, que pode influenciar o comportamento do sistema. Já a operacao detalha o tipo da transação, como "CREDITO" ou "DEBITO". O campo service descreve o serviço específico que foi acionado, como exemplo de teste "spi-lancamentos-integracao-nodejs". A análise desse dado é fundamental para verificar se as anomalias estão concentradas em um serviço em particular, permitindo avaliação cada vez mais aprofundadas dos componentes tecnológicos da solução Pix. Por fim, o service_encoded é a versão numérica do campo service, utilizada para facilitar o processamento pelo modelo.

Para apresentar os resultados de forma mais objetiva, vamos destacar a eficácia do modelo em diferentes cenários, com o intuito de demonstrar como o modelo se comporta e sua efetividade em situações variadas, validando a sua aplicação em um ambiente de testes bem similar ao ambiente de produção do Sicoob. Focamos então na análise do comportamento do modelo de detecção de anomalias dentro dos 82 micros serviços do PIX, de forma a verificar o comportamento do modelo quanto a capacidade de identificar e classificar diferentes tipos de micros serviços.

Apesar de o modelo ser aplicado em todos os micros serviços do PIX, é importante mencionar que nem todos possuem os dados necessários para sua identificação. Por esse motivo, nem todos terão seus tempos classificados.

A Figura 29 – Avalição do micro serviço serve como referência para avaliação do micro serviço selecionado. A observação principal na tabela é a correlação entre a duração da operação e o resultado da detecção de anomalia. As linhas destacadas mostram que, quando o tempo de resposta (duracao) da aplicação se eleva, o modelo classifica a transação como uma anomalia (indicada pelo valor -1 no campo anomaly).

Figura 29 – Avaliação do micro serviço

	A	B	C	D	E	F	G	H	I	J	K
1	estado	tipoQrcode	@timestamp	operacao	service	duracao	hora	minuto	dia_semana	service_encoded	anomaly
2	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	656	21	11	6	1	1
3	FINALIZADO_SUCESSO		2025-09-08 21:55:08.601000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1164	21	11	6	1	-1
4	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	475	21	11	6	1	1
5	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.520000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	400	21	11	6	1	1
6	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.508000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1210	21	11	6	1	-1
7	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	620	21	11	6	1	1
8	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1394	21	11	6	1	-1
9	RESPONDIDO		2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	501	21	11	6	0	-1
10	RESPONDIDO		2025-09-08 21:55:08.650000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	628	21	11	6	0	-1
11	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.676000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	867	21	11	6	1	-1
12	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.539000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	847	21	11	6	1	-1
13	FINALIZADO_SUCESSO	ESTATICO	2025-09-08 21:55:08.568000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1059	21	11	6	1	-1
14	RESPONDIDO		2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	794	21	11	6	0	-1
15	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.615000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	483	21	11	6	1	1
16	FINALIZADO_SUCESSO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1031	21	11	6	1	-1
17	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.655000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1074	21	11	6	1	-1
18	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.715000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	532	21	11	6	1	1
19	LIQUIDADO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	494	21	11	6	1	1
20	RESPONDIDO		2025-09-08 21:55:08.672000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	592	21	11	6	0	-1
21	RESPONDIDO		2025-09-08 21:55:08.714000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	583	21	11	6	0	-1
22	RESPONDIDO		2025-09-08 21:55:08.762000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	611	21	11	6	0	-1
23	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.701000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	690	21	11	6	1	1
24	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1278	21	11	6	1	-1
25	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.751000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1293	21	11	6	1	-1
26	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.789000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1233	21	11	6	1	-1
27	RESPONDIDO		2025-09-08 21:55:08.739000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1186	21	11	6	0	-1
28	RESPONDIDO		2025-09-08 21:55:08.876000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1030	21	11	6	0	-1
29	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1301	21	11	6	1	-1
30	RESPONDIDO		2025-09-08 21:55:08.760000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	479	21	11	6	0	-1
31	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.877000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	441	21	11	6	1	1
32	LIQUIDADO		2025-09-08 21:55:08.871000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1094	21	11	6	1	-1
33	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.894000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	378	21	11	6	1	1
34	LIQUIDADO		2025-09-08 21:55:09.008000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	520	21	11	6	1	1
35	LIQUIDADO		2025-09-08 21:55:08.961000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1061	21	11	6	1	-1
36	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.963000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	629	21	11	6	1	1
37	RESPONDIDO		2025-09-08 21:55:08.990000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	909	21	11	6	0	-1

Fonte: AUTOR, 2025.

Essa consistência demonstra a precisão e a eficácia do modelo em identificar comportamentos atípicos dentro de um mesmo serviço. Ele consegue diferenciar um fluxo de operação normal de um comportamento de performance degradada, mesmo que ambas as transações ocorram no mesmo micro serviço. Isso confirma a capacidade do modelo de capturar e sinalizar desvios importantes que podem indicar problemas no sistema.

Expandindo a análise, é importante avaliar o comportamento do modelo de detecção de anomalias em diferentes micros serviços. O objetivo é verificar se o modelo é capaz de identificar a diferença de comportamentos entre cada serviço, em vez de aplicar uma regra genérica de anomalia baseada apenas em um único fator, como o tempo de resposta. A Figura 30 – Capacidade do modelo ilustra essa capacidade do modelo.

Figura 30 – Capacidade do modelo

	A	B	C	D	E	F	G	H	I	J	K
1	estado	tipoQrcode	@timestamp	operacao	service	duracao	hora	minuto	dia_semana	service_encoded	anomaly
2	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	656	21	11	6	1	1
3	FINALIZADO_SUCESSO		2025-09-08 21:55:08.601000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1164	21	11	6	1	-1
4	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.586000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	475	21	11	6	1	1
5	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.520000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	400	21	11	6	1	1
6	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.508000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1210	21	11	6	1	-1
7	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	620	21	11	6	1	1
8	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1394	21	11	6	1	-1
9	RESPONDIDO		2025-09-08 21:55:08.580000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	501	21	11	6	0	-1
10	RESPONDIDO		2025-09-08 21:55:08.650000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	628	21	11	6	0	-1
11	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.676000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	867	21	11	6	1	-1
12	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.539000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	847	21	11	6	1	-1
13	FINALIZADO_SUCESSO	ESTATICO	2025-09-08 21:55:08.568000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1059	21	11	6	1	-1
14	RESPONDIDO		2025-09-08 21:55:08.573000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	794	21	11	6	0	-1
15	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.615000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	483	21	11	6	1	1
16	FINALIZADO_SUCESSO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1031	21	11	6	1	-1
17	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.655000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1074	21	11	6	1	-1
18	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.715000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	532	21	11	6	1	1
19	LIQUIDADO		2025-09-08 21:55:08.621000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	494	21	11	6	1	1
20	RESPONDIDO		2025-09-08 21:55:08.672000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	592	21	11	6	0	-1
21	RESPONDIDO		2025-09-08 21:55:08.714000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	583	21	11	6	0	-1
22	RESPONDIDO		2025-09-08 21:55:08.762000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	611	21	11	6	0	-1
23	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.701000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	690	21	11	6	1	1
24	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1278	21	11	6	1	-1
25	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.751000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1293	21	11	6	1	-1
26	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.789000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1233	21	11	6	1	-1
27	RESPONDIDO		2025-09-08 21:55:08.739000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1186	21	11	6	0	-1
28	RESPONDIDO		2025-09-08 21:55:08.876000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	1030	21	11	6	0	-1
29	FINALIZADO_SUCESSO	DINAMICO	2025-09-08 21:55:08.705000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1301	21	11	6	1	-1
30	RESPONDIDO		2025-09-08 21:55:08.760000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	479	21	11	6	0	-1
31	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.877000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	441	21	11	6	1	1
32	LIQUIDADO		2025-09-08 21:55:08.871000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1094	21	11	6	1	-1
33	LIQUIDADO	DINAMICO	2025-09-08 21:55:08.894000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	378	21	11	6	1	1
34	LIQUIDADO		2025-09-08 21:55:09.008000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	520	21	11	6	1	1
35	LIQUIDADO		2025-09-08 21:55:08.961000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	1061	21	11	6	1	-1
36	LIQUIDADO	ESTATICO	2025-09-08 21:55:08.963000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	629	21	11	6	1	1
37	RESPONDIDO		2025-09-08 21:55:08.990000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	909	21	11	6	0	-1
38	LIQUIDADO		2025-09-08 21:55:08.782000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	403	21	11	6	1	1
39	RESPONDIDO		2025-09-08 21:55:09.013000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	460	21	11	6	0	-1
40	LIQUIDADO	DINAMICO	2025-09-08 21:55:09.053000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	435	21	11	6	1	1
41	LIQUIDADO	ESTATICO	2025-09-08 21:55:09.047000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	565	21	11	6	1	1
42	RESPONDIDO		2025-09-08 21:55:08.937000+00:00	CREDITO	spi-lancamentos-envio-integracao-nodejs	709	21	11	6	0	-1
43	LIQUIDADO		2025-09-08 21:55:08.811000+00:00	CREDITO	spi-lancamentos-integracao-nodejs	999	21	11	6	1	-1

Fonte: AUTOR, 2025.

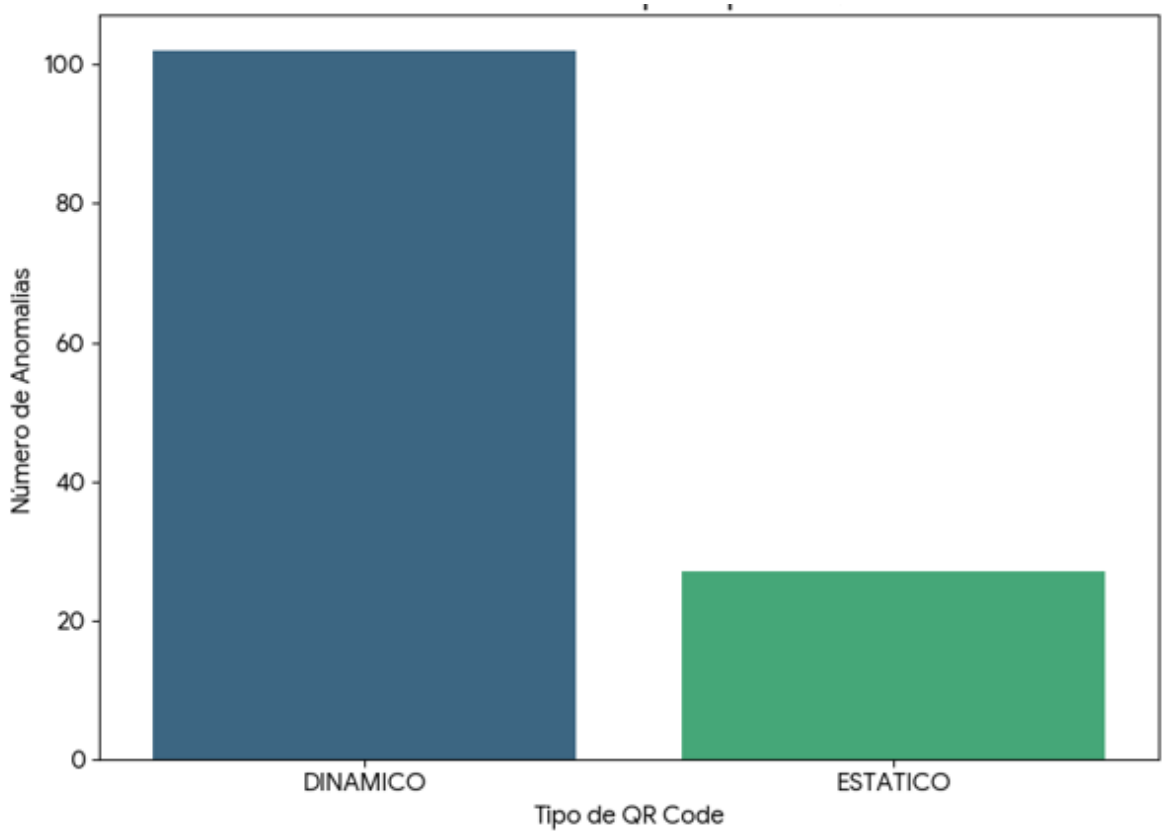
Observamos que o micro serviço spi-lancamentos-integracao-nodejs, destacado em verde, teve uma transação classificada como normal (anomaly = 1), mesmo com um tempo de resposta relativamente alto, de 620ms, em comparação com outros micros serviços.

Em contraste, o micro serviço spi-lancamentos-envio-integracao-nodejs, destacado em vermelho, teve uma transação classificada como anomalia (anomaly = -1) com um tempo de resposta de apenas 479ms, significativamente menor que o tempo da transação normal no outro micro serviço.

Esses resultados demonstram a qualidade do modelo. Em vez de usar um limite fixo para a duração, ele aprendeu o comportamento normal de cada micro serviço individualmente. Para o spi-lancamentos-envio-integracao-nodejs, um tempo de 479ms pode ser anormalmente alto, indicando uma performance degradada para aquele serviço em específico, enquanto 620ms pode ser considerado um tempo de resposta esperado para o spi-lancamentos-integracao-nodejs. Isso prova que o modelo é capaz de identificar anomalias relativas ao contexto de cada micro serviço, validando sua eficácia e flexibilidade em um ambiente de arquitetura distribuída.

Avançando nas análises dos resultados, verificamos a relação entre o Tipo de Transação e a Anomalia, por meio da análise do campo tipo_qrcode, conforme Figura 31 – Anomalias detectadas por QR Code.

Figura 31 – Anomalias detectadas por QR Code.

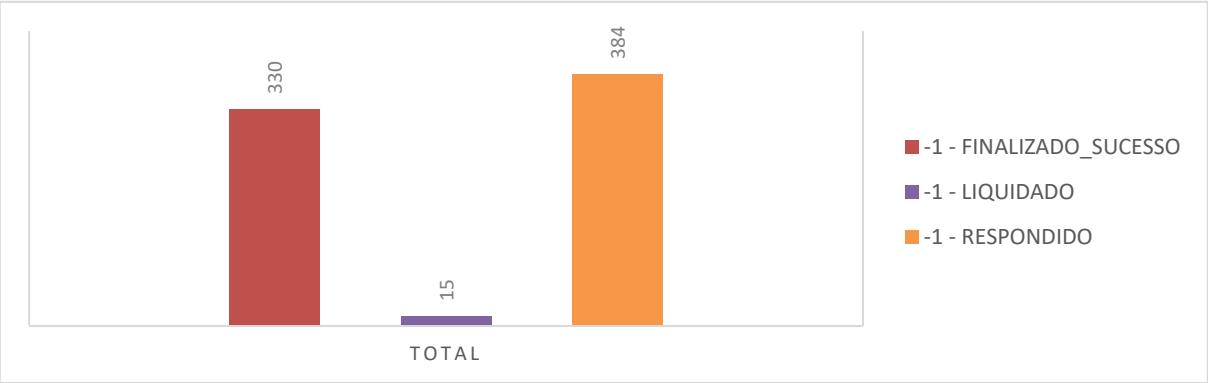


Fonte: AUTOR, 2025.

É possível verificar que o modelo identifica mais anomalias em transações com QR Codes dinâmicos do que em QR Codes estáticos. Isso ocorre porque as transações dinâmicas envolvem mais etapas e interações com a aplicação, tendo assim mais pontos potenciais para atrasos ou falhas. Fazer essa comparação permite entender se certos tipos de transações são mais suscetíveis a comportamentos anormais, oferecendo novas percepções para a otimização do ambiente tecnológico e dos processos do Sicoob.

Na Figura 32 – Tipos de Operações, é possível verificar que operações com o estado “FINALIZADO_SUCESSO” possuem mais anomalias que outros tipos de operação, reforçando a utilidade do modelo para identificar problemas de desempenho, mesmo em operações aparentemente bem-sucedidas.

Figura 32 – Tipos de Operações



Fonte: AUTOR, 2025.

5 Conclusões

O presente trabalho teve como objetivo principal realizar um estudo, utilizando técnicas de Big Data e Inteligência Artificial , para analisar as informações contidas nos milhões de logs coletados nas transações financeiras realizadas com Pix e armazenados em ambiente de Big Data do Sicoob, visando criar um modelo que auxiliasse na detecção e classificação de anomalias do ambiente tecnológico, tornando-o mais resiliente e, assim, evitando falhas que resultassem em paradas parciais ou totais dos serviços Pix.

Conforme apresentado nos Capítulos 3 e 4, o objetivo principal foi alcançado. A metodologia adotada estabeleceu um fluxo de trabalho claro e eficiente, que se integrou à arquitetura de observabilidade e Big Data já existente no Sicoob. A estratégia de coleta de dados, utilizando o Logstash para espelhamento e filtragem de logs do Apache Kafka, foi fundamental para a montagem de um dataset limpo e específico para o treinamento, garantindo a qualidade e a relevância das características (tempo de duração, serviço, hora, minuto e dia da semana). Essa abordagem demonstrou a capacidade de transformar dados não estruturados de logs em informações estruturadas e acionáveis, confirmando a necessidade do Sicoob em processar volumes massivos de dados (Big Data) para extrair percepções.

A implementação do modelo não supervisionado Isolation Forest e o ajuste fino de seus parâmetros, como a taxa de contaminação para 1%, provaram ser a escolha adequada. Os experimentos detalhados no Capítulo 4 validaram a capacidade do modelo de aprender o comportamento normal de cada um dos diferentes micros serviços do Pix de forma contextualizada. Ao classificar corretamente transações como anômalas, mesmo em cenários de sucesso transacional (FINALIZADO_SUCESSO), o modelo comprovou sua eficácia na detecção de degradação de desempenho, um precursor comum de falhas completas e indisponibilidades, alinhando-se diretamente ao propósito de aumentar a resiliência cibernética e evitar prejuízos.

Em suma, a criação e a validação do modelo de IA, integrado ao fluxo de logs do Pix, demonstraram que a metodologia proposta é viável e eficaz para a geração de percepções preditivas, transformando a gestão de falhas de uma postura reativa para uma postura proativa e preventiva, como previsto nos desafios da organização. A análise aprofundada dos resultados permitiu uma conclusão executiva de confiança quanto à maturidade do modelo de detecção de anomalias para sua implantação em ambiente de produção tecnológica do Sicoob.

O modelo, além de ser tecnicamente robusto, demonstrou duas qualidades operacionais cruciais. A primeira foi a sensibilidade contextual. Nessa primeira qualidade, o modelo não se baseou em um limite fixo de duração (tempo de resposta) para definir anomalia. Pelo contrário, ele aprendeu o padrão de tempo de resposta individual de cada micro serviço (tabela do Capítulo 4), classificando um tempo de 479ms como anômalo para um serviço, mas considerando 620ms como normal para outro. Essa precisão contextual minimiza drasticamente a ocorrência de falsos positivos (alertas desnecessários), um fator decisivo para a confiança das equipes de Operação e para a estabilidade do ambiente de produção. A segunda qualidade foi a identificação de precursores de falha. Nessa segunda qualidade, a capacidade de detectar anomalias em transações FINALIZADO_SUCESSO (Figura 32) é uma contribuição de alto valor. O modelo pode sinalizar uma degradação sutil de desempenho (aumento de latência) que precede uma falha total, permitindo que as equipes técnicas realizem ações preventivas, como o aumento de capacidade (escalabilidade horizontal) ou reiniciar os componentes, antes que o tempo de inatividade afete o cliente final.

Dada a validação bem-sucedida, especialmente em cenários dinâmicos (Figura 31), o modelo está pronto para uma implantação monitorada no ambiente de produção, conforme o procedimento de Validação Contínua (Figura 16). A integração via micro serviço Flask, enviando alertas padronizados via protocolo GELF para o Graylog e Grafana, assegura que a solução se encaixe sem disrupção nas ferramentas de monitoramento já utilizadas pelo Sicoob. A aplicação consistente e o uso dos alertas gerados têm o potencial de aumentar o MTBF e reduzir o MTTR, impactando positivamente o SLA de Disponibilidade da plataforma Pix, conforme as metas de resiliência cibernética estabelecidas pela organização.

Destacamos que esse trabalho contribuiu cientificamente ao demonstrar a viabilidade de aplicar algoritmos não supervisionados de Aprendizado de Máquina em fluxos de Big Data, dos logs transacionais em tempo real, para a melhoria da resiliência cibernética em um produto financeiro crítico como o Pix. A principal contribuição prática é a transformação de um ambiente de observabilidade reativo em um ambiente preditivo.

Finalmente, para trabalhos futuros, sugerem-se algumas linhas de pesquisa correlacionadas. A primeira delas é detecção de anomalias em múltiplas dimensões, expandindo o dataset para incluir outras características, como uso de CPU/Memória dos micros serviços, latência da rede (*ping*) e volume transacional. Um modelo treinado com essas múltiplas dimensões poderia identificar anomalias mais complexas (ex: baixa latência com pico de CPU indicando um loop de processamento anômalo), aumentando a acurácia diagnóstica. A segunda

linha é o desenvolvimento de autorreparo (*self-healing*) automatizado, utilizando os alertas gerados pelo modelo de IA como gatilho para ações de remediação automatizada. Por exemplo, a detecção de uma anomalia em um micro serviço pode acionar automaticamente um script para isolar a instância com falha e escalar uma nova instância, promovendo um autorreparo na infraestrutura. A terceira linha é a comparação e otimização de algoritmos, realizando estudos comparativos entre o Isolation Forest e outros algoritmos de detecção de anomalias de *streaming*, como o One-Class SVM ou Autoencoders, otimizados para detecção em tempo real. Por fim, a análise preditiva de valores atípicos das transações financeiras, aplicando o modelo para monitorar valores e volumes financeiros das transações de Pix, separando o contexto de resiliência cibernética para detectar fraudes ou comportamentos de lavagem de dinheiro que se manifestam fora do padrão de uso do Pix pelos cooperados do Sicoob.

Referências

BALANÇO SISBR 2024. *Balanço Sisbr 2024.* Disponível em: https://www.sicoob.com.br/group/balancosisbr-2024.
Acesso em 15 de março de 2025.

BANCO CENTRAL DO BRASIL. *Pix.* Disponível em https://www.bcb.gov.br/estabilidadefinanceira/Pix.
Acesso em 16 de fevereiro de 2025.

BANCO CENTRAL DO BRASIL. *Cooperativas de crédito.* Disponível em: https://www.bcb.gov.br/estabilidadefinanceira/cooperativacredito.
Acesso em 09 de fevereiro de 2025.

BANCO CENTRAL DO BRASIL. *Pix.* Disponível em https://www.bcb.gov.br/estabilidadefinanceira/estatisticasPix.
Acesso em 26 de fevereiro de 2025.

BELACEL, N. and RICHARD,R and RANGAVAJJALA, D.P. and ADHADUK, R. *Online Anomaly Detection for Streaming Data Implemented on Top of Kafka, Scikit-Multiflow and River.* 2021.
https://link.springer.com/chapter/10.1007/978-3-030-89912-7_63.
Acesso em 27 de fevereiro de 2025.

BHANAGE, A.D. and PAWAR, A.V. *Bibliometric survey of IT Infrastructure Management to Avoid Failure Conditions*. 2020.
https://www.emerald.com/insight/content/doi/10.1108/idd-06-2020-0060/full/html.
Acesso em 27 de fevereiro de 2025.

BHANAGE, A.D. and PAWAR, A.V. and KOTECHA, K and ABRAHAM, A. *Failure Detection Using Semantic Analysis and Attention-Based Classifier Model for IT Infrastructure Log Data.* India: 2023.
https://core.ac.uk/download/pdf/402561511.pdf.
Acesso em 27 de fevereiro de 2025.

BHANAGE, A.D. and PAWAR, A.V. and VISHAL, A and KOTECHA, K. *Review and Analysis of Failure Detection and Prevention Techniques in IT Infrastructure Monitoring.* Nebraska: 2021.
https://core.ac.uk/download/pdf/402561511.pdf.
Acesso em 27 de fevereiro de 2025.

BRAGA, Fabiane and GOMES, Elisabeth. *Inteligência Competitiva em Tempos de Big Data*. São Paulo: 2017.

ÉPOCA NEGÓCIOS. *Pix deve crescer 35% ao ano e superar cartões nas vendas online até 2027.* Disponível em https://epocanegocios.globo.com/economia/noticia/2025/01/Pix-deve-crescer-35percent-ao-ano-e-superar-cartoes-nas-vendas-online-ate-2027-aponta-relatorio.ghtml.
Acesso em 10 de março de 2025.

ESTADÃO. *Pesquisa: PIX fora do ar: o que causou o problema que atingiu milhões de brasileiros?* Disponível em https://einvestidor.estadao.com.br/educacao-financeira/Pix-fora-do-ar-o-que-aconteceu/.
Acesso em 10 de março de 2025.

FORBES. *Pesquisa: Brasil torna-se segundo maior mercado de pagamentos instantâneos do mundo.* Disponível em https://forbes.com.br/forbes-money/2024/04/pesquisa-brasil-torna-se-segundo-maior-mercado-de-pagamentos-instantaneos-do-mundo/.
Acesso em 10 de março de 2025.

GOLDENCLOUD.TECH. *Inteligência Artificial e big data: entenda como a relação dessas tecnologias pode ser benéfica para você.* Disponível em https://blog.dbacorp.com.br/2021/08/11/inteligencia-artificial-e-big-data/
Acesso em 09 de fevereiro de 2025.

GOOGLE CLOUD. *Pesquisa: O que é Big Data?* Disponível em https://cloud.google.com/learn/what-is-big-data?hl=pt-BR
Acesso em 06 de fevereiro de 2025.

INFRASTRUCTURE OBSERVABILITY. *Pesquisa: O que é Observabilidade em infraestrutura de TI?* Disponível em https://www.dynatrace.com/platform/infrastructure-observability/
Acesso em 30 de março de 2025.

KOSINSK, Daniel. *A digitalização dos meios de pagamento: O Pix e as Central Bank Digital Currences em perspectiva comparada*. Rio de Janeiro: TEC – Textos de Economia, 2021.

MARTIN, C and LANGENDOERFER, P and ZARRIN, P and DÍAZ, M and BARTOLOMÉ, R. *Kafka-ML: Connecting the data stream with ML/AI frameworks. India: 2022*.
https://www.sciencedirect.com/science/article/pii/S0167739X21002995.
Acesso em 02 de março de 2025.

MEHTA, S. and KOTHURI, P. and GARCIA, D.L. *Anomaly Detection for Network Connection Logs.* Ithaca: 2018.
https://arxiv.org/abs/1812.01941.
Acesso em 09 de fevereiro de 2025.

MEINEN, Ênio and PORT, Márcio. *O cooperativismo de crédito ontem, hoje e amanhã*. Brasília: Confebras, 2021.

MTTR e MTBF. *Pesquisa: MTTR e MTBF, o que são e quais suas diferenças.* Disponível em https://www.opservices.com.br/mttr-e-mtbf/#:~:text=O%20que%20s%C3%A3o%20MTBF%20e,MTBF%20e%20diminuir%20o%20MTTR.
Acesso em 30 de março de 2025.

NUBANK. *Pix: tudo que você precisa saber sobre o meio de pagamentos*. Disponível em https://blog.nubank.com.br/Pix-tudo-sobre/.
Acesso em 10 de março de 2025.

ORGANIZAÇÃO DAS COOPERATIVAS BRASILEIRAS (OCB). *Impactos do Cooperativismo de Crédito no Brasil: Um Estudo da Fipe*. Brasília, 2024.

O'SULLIVAN, Conor. *Pesquisa: Isolation Forest Guide: Explanation and Python Implementation.* Eua: 2024. Disponível em https://www.datacamp.com/tutorial/isolation-forest
Acesso em 06 de maio de 2025.

REZENDE, Solange. *Sistemas inteligentes: Fundamentos e Aplicações*. São Paulo: 1994.

SCIKIT-LEARN. *Pesquisa: Using Scikit-Learn – Machine Learning in Python?* Disponível em https://scikit-learn.org/stable/index.html.
Acesso em 07 de abril de 2025.

SICOOB. *Site oficial.* Disponível em: https://www.sicoob.com.br. Acesso em 26 de fevereiro de 2025.

SILVA, João. *O desenvolvimento de competências-chave em cibernética por profissionais que não são de TI: uma análise dos possíveis impactos sobre a resiliência cibernética das organizações*. Rio de Janeiro: Repositório Institucional da ESG, 2023. https://repositorio.esg.br/handle/123456789/1905.
Acesso em 16 de fevereiro de 2025.

SOFTPLAN. *Pesquisa: Observabilidade: O que é, vantagens, pilares e qual a diferença para o monitoramento!* Disponível em https://www.softplan.com.br/tech-writers/o-que-e-observabilidade
Acesso em 30 de março de 2025.

TECHCHANNEL. *The Cost of Enterprise Downtime.* Disponível em https://techchannel.com/data-security/the-cost-of-enterprise-downtime/.
Acesso em 10 de março de 2025.