Detecção de Anomalias em Transações Financeiras: Comparação entre Isolation Forest, LOF, KNN e Autoencoders sob Desbalanceamento Extremo

1st Marcos Didier Oliveira Neto
Centro de Informática – UFPE
Recife, Brasil
mdon@cin.ufpe.br

2nd João Pedro Bezerra de Melo Teixeira de Lima
Centro de Informática – UFPE
Recife, Brasil
jpbmtl@cin.ufpe.br

3rd Cesar Rio Tinto Cavalcanti
Centro de Informática – UFPE
Recife, Brasil
crtc@cin.ufpe.br

4th Guilherme de Oliveira Costa Campelo
Centro de Informática – UFPE
Recife, Brasil
gocc@cin.ufpe.br

***Resumo*—Fraudes em transações financeiras configuram um problema de elevada relevância prática, marcado por desbalanceamento extremo e custos assimétricos de erro. Neste trabalho, investigamos técnicas de detecção de anomalias aplicadas a transações com cartão de crédito, explorando abordagens não supervisionadas e semi-supervisionadas. Utilizamos o conjunto *Credit Card Fraud*, com mais de 280 mil transações reais, das quais aproximadamente 0,17% são fraudulentas. Avaliamos métodos de famílias distintas: modelos probabilísticos baseados em isolamento (Isolation Forest), métodos baseados em densidade (Local Outlier Factor) e modelos de *deep learning* baseados em reconstrução (Autoencoder e Variational Autoencoder). A comparação foi conduzida por métricas apropriadas a cenários de desbalanceamento extremo, com destaque para Average Precision (AP), Recall@500 e F2-score. Os resultados indicam que o VAE oferece melhor desempenho global, sugerindo maior efetividade na priorização e captura de padrões de fraude sob forte raridade de eventos.**

I. Introdução

Fraudes em transações financeiras representam um desafio central para instituições bancárias, operadoras de cartão e plataformas de pagamento digital. O aumento do volume de transações eletrônicas, aliado à sofisticação de estratégias maliciosas, torna inviável a inspeção manual e demanda sistemas automatizados capazes de operar em larga escala e com respostas rápidas. Em aplicações reais, falhas de detecção podem resultar em perdas financeiras diretas, custos operacionais, danos reputacionais e degradação da confiança do cliente.

Do ponto de vista de aprendizado de máquina, esse domínio é particularmente desafiador por dois fatores. O primeiro é o **desbalanceamento extremo**: a classe fraudulenta é rara, o que tende a induzir modelos a privilegiarem a classe majoritária se métricas e estratégias não forem cuidadosamente escolhidas. O segundo é a **assimetria de custos**: falsos negativos (fraudes não detectadas) são, em geral, mais críticos do que falsos positivos (transações legítimas sinalizadas), embora estes também causem impacto na experiência do usuário.

Nesse contexto, técnicas de **detecção de anomalias** são particularmente adequadas, pois permitem modelar o comportamento normal e identificar padrões atípicos, com menor dependência de rótulos. O objetivo deste projeto é compreender **quais modelos se relacionam melhor com o problema de fraude sob desbalanceamento extremo e por quê**, comparando famílias distintas e discutindo seus trade-offs à luz de métricas apropriadas e do custo de erros no mundo real.

II. Análise de Dados e Feature Engineering

A. Panorama do Conjunto de Dados

O conjunto de dados utilizado é o *Credit Card Fraud Dataset*, contendo 284.807 transações reais, das quais 492 são fraudulentas (aproximadamente 0,17%). O dataset dispõe de 28 atributos numéricos (V1 a V28), obtidos por uma transformação de **PCA** para preservação de privacidade. Além disso, inclui a variável *Time* (tempo decorrido desde a primeira transação) e *Amount* (valor monetário).

Não foram identificadas instâncias com valores nulos. Entretanto, observou-se a presença de algumas duplicatas, tratadas no pipeline. A Fig. 1 evidencia o desbalanceamento severo.

B. Análise Exploratória Univariada

Nesta etapa, decidimos analisar a **distribuição das classes** para tornar explícita a magnitude do desbalanceamento e justificar escolhas metodológicas subsequentes (métricas, treinamento semi-supervisionado e definição de limiares). A Fig. 1 mostra a discrepância entre transações genuínas e fraudulentas, sendo um aspecto determinante para a seleção de métricas mais informativas do que acurácia.

Também analisamos a variável *Amount*. Observou-se que a mediana dos valores de transações não fraudulentas é aproximadamente 22, enquanto para transações fraudulentas é cerca de 9,25. Esse comportamento sugere que fraudes

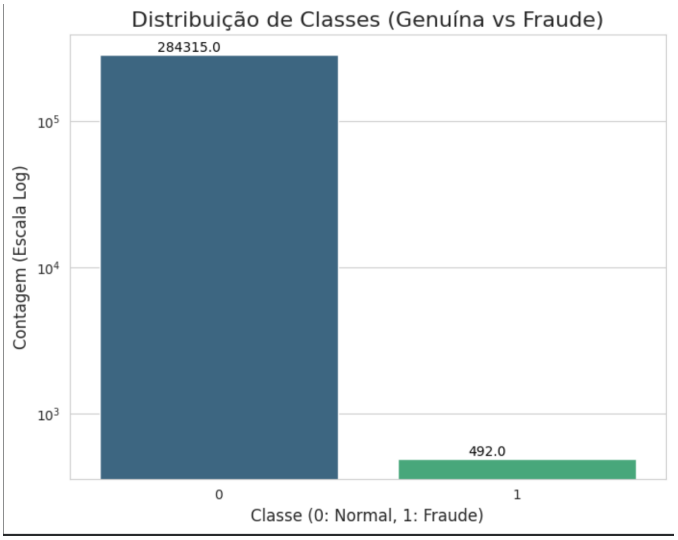

Figura 1. Distribuição das classes (escala logarítmica).

frequentemente operam com valores menores, possivelmente para reduzir suspeitas e evitar regras heurísticas simples.

C. Análise Bivariada

Para investigar como distribuições diferem entre classes em variáveis específicas, empregamos ***violin plots***, pois eles evidenciam simultaneamente densidade, assimetria e amplitude das distribuições, além de serem úteis quando há forte desbalanceamento.

Nessa análise, verificamos diferenças marcantes em componentes PCA específicas. Em particular, V14 e V17 apresentaram separação visual expressiva, com **ranges** e densidades distintos para as classes. A Fig. 2 exemplifica esse comportamento para V17.

Figura 2. Distribuição da variável V17 por classe.

Além disso, analisamos a relação conjunta entre V14 e V17 por meio de um gráfico de densidade (normais) sobreposto à dispersão das fraudes. A Fig. 3 evidencia um padrão importante: enquanto as instâncias normais se concentram em um **cluster** de alta densidade (região compacta), as instâncias fraudulentas aparecem **bem distribuídas** ao longo de uma faixa mais ampla no plano (V14, V17). Esse comportamento é coerente com a heterogeneidade das fraudes, que podem assumir múltiplos "modos" e ocupar regiões de baixa densidade, motivando modelos que exploram densidade local e/ou reconstrução não linear.

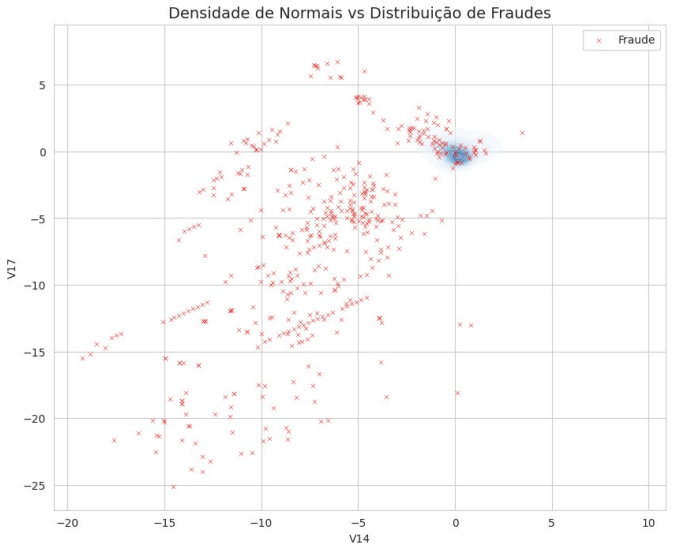

Figura 3. Densidade de transações normais (V14 vs V17) e dispersão das fraudes.

D. Análise Multivariada

Para uma visão global no espaço de atributos, utilizamos t-SNE após o pré-processamento (com amostragem de 20k transações normais e todas as fraudes disponíveis, conforme pipeline). O t-SNE foi empregado **exclusivamente para fins exploratórios**. A Fig. 4 sugere a presença de agrupamentos e regiões em que fraudes se concentram, motivando métodos baseados em densidade e modelos capazes de capturar representações não lineares.

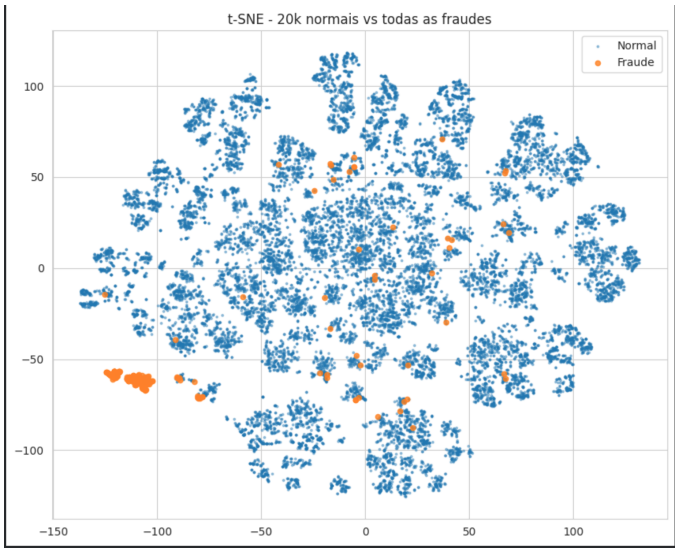

Figura 4. Visualização t-SNE de transações normais e fraudulentas.

E. Pré-processamento

Apesar de o dataset apresentar boa qualidade inicial (sem nulos), seguimos um pipeline sistemático:

- Remoção de duplicatas;
- Divisão dos dados em treino (70%), validação (15%) e teste (15%);
- Padronização por *StandardScaler*;
- *Feature selection* para estabilizar métodos sensíveis à dimensionalidade.

Adotamos a estratégia de **não incluir fraudes no conjunto de treino**. A justificativa principal é que modelos baseados em reconstrução (Autoencoders) **não devem ser treinados com padrões fraudulentos**, pois o objetivo é aprender uma representação do comportamento normal e atribuir maior erro de reconstrução a padrões anômalos.

IF e LOF também foram treinados apenas com dados normais porque escolhemos tratar a fraude como *novelty* (novidade) em relação ao comportamento legítimo. Em particular, no LOF utilizamos **novelty detection** (`novelty=True`), em que o modelo é ajustado no conjunto normal e aplicado a novas amostras (validação/teste). Assim, embora o termo *outlier* seja comum em anomalias, a intenção prática foi aproximar o cenário de fraude: **fraudes como novidades** frente ao padrão normal.

F. Feature Engineering (Seleção de Atributos)

Para **feature selection**, utilizamos *feature importance* baseada em Random Forest, pois o método captura relações não lineares, lida bem com interações entre variáveis e tende a ser robusto a outliers. A Fig. 5 resume as importâncias obtidas.

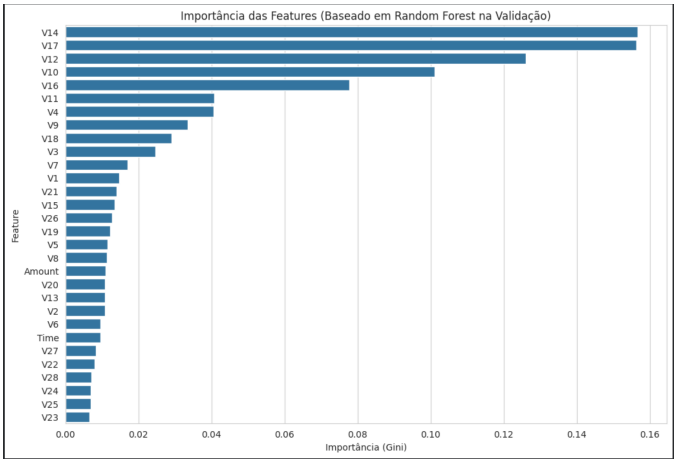

Figura 5. Importância das variáveis (Random Forest na validação).

Com base no gráfico, definimos um critério simples e interpretável: **remover variáveis com importância inferior a 1%**. Esse corte resultou na remoção de **8 variáveis** e manteve as componentes mais informativas (com destaque para V14, V17 e V12), reduzindo dimensionalidade e ruído. Essa decisão foi especialmente relevante porque utilizamos LOF, método sensível à **maldição da dimensionalidade**: em alta dimensão, distâncias e densidades locais perdem poder discriminativo.

III. Modelagem

Selecionamos modelos de três categorias: (i) probabilísticos (Isolation Forest), (ii) baseados em densidade (LOF) e (iii) *deep learning* (Autoencoder e VAE). Para todos os modelos, realizamos uma busca por hiperparâmetros por meio de um **Grid Search limitado**, equilibrando custo computacional e desempenho. O **critério de seleção** do grid foi a **maior Average Precision (AP) no conjunto de validação**.

Definição de limiar (threshold). Independentemente do modelo, os thresholds finais foram escolhidos de forma consistente **maximizando o F2-score no conjunto de validação**, refletindo a preferência por maior recall em um contexto em que falsos negativos são mais custosos.

A. Isolation Forest

O Isolation Forest (iForest) isola observações por meio de partições aleatórias: anomalias tendem a ser isoladas com menos divisões, produzindo escores mais altos. Escolhemos iForest em detrimento de GMM por exigir menos suposições sobre distribuição e por escalar bem em alta dimensionalidade.

O Grid Search do Isolation Forest foi executado com o seguinte espaço de busca:

```
param_grid_if = {
    'n_estimators': [200, 500],
    'max_samples': [128, 256, 512],
    'max_features': [0.6, 0.8, 1.0],
    'bootstrap': [False, True],
}
```

A Fig. 6 apresenta a matriz de confusão (teste) do iForest com feature selection.

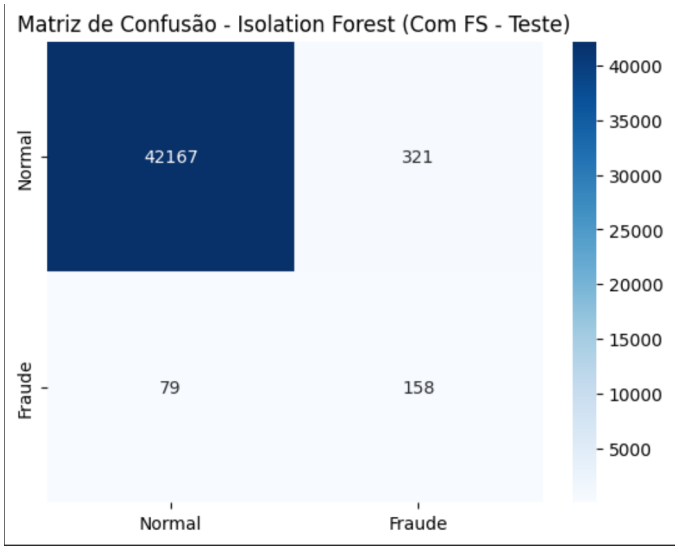

Figura 6. Matriz de confusão do Isolation Forest (teste).

B. Local Outlier Factor (LOF)

O LOF compara a densidade local de uma instância com a densidade de seus vizinhos. Optamos por LOF por sua capacidade de capturar **anomalias locais**. Usamos `novelty=True` para treinar no conjunto normal e aplicar em novas amostras

(validação/teste), alinhando a detecção ao cenário de **fraude como novidade** frente ao padrão normal.

O Grid Search do LOF foi executado com o seguinte espaço de busca:

```
param_grid_lof = {
   'n_neighbors': [20, 50, 100],
   'metric': ['euclidean', 'minkowski'],
   'contamination': ['auto']
}
```

A Fig. 7 mostra a matriz de confusão (teste) do LOF.

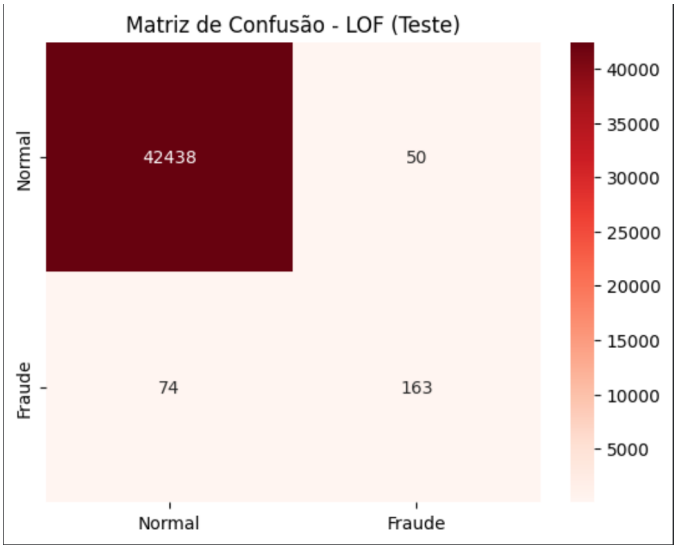

Figura 7. Matriz de confusão do LOF (teste).

C. Autoencoder

Autoencoders são redes neurais treinadas para reconstruir a entrada. Tipicamente, a arquitetura é composta por um **encoder** que comprime a entrada em uma representação latente e um **decoder** que reconstrói a entrada a partir dessa representação. Em detecção de anomalias, o treinamento é realizado apenas com exemplos normais, fazendo com que o modelo aprenda o "manifold" do comportamento legítimo. Assim, amostras normais tendem a apresentar **baixo erro de reconstrução**, enquanto fraudes (com padrões distintos) tendem a gerar **maior erro**, permitindo ranqueá-las e classificá-las via threshold.

O Grid Search do Autoencoder variou o tamanho do gargalo (*bottleneck*), isto é, a dimensão do espaço latente, conforme:

```
bottleneck_grid = [4, 8, 10, 16]
```

A Fig. 8 apresenta a matriz de confusão (teste) do Autoencoder.

D. Beta Variational Autoencoder (VAE)

O Variational Autoencoder (VAE) estende o autoencoder clássico ao introduzir uma formulação **probabilística** do espaço latente. Em vez de mapear a entrada para um ponto determinístico, o encoder aprende uma distribuição aproximada $q_\theta(z|x)$ (por exemplo, média e variância), enquanto o decoder

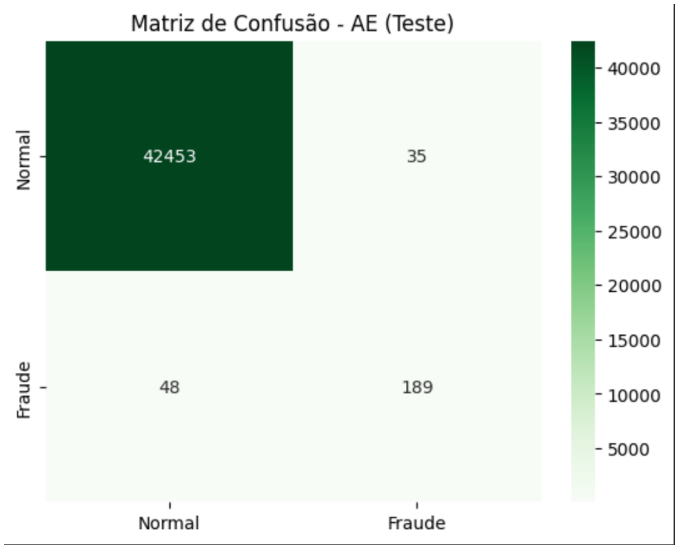

Figura 8. Matriz de confusão do Autoencoder (teste).

reconstrói a entrada a partir de amostras de z. O treinamento maximiza a ELBO, combinando fidelidade de reconstrução e regularização do latente via divergência KL.

No **Beta-VAE**, um fator β controla a força da regularização. Valores $\beta > 1$ impõem maior estruturação do espaço latente, favorecendo generalização. Em detecção de anomalias, esse tipo de regularização tende a produzir representações latentes mais "compactas" para o comportamento normal, aumentando a discrepância (pior reconstrução) quando o padrão é anômalo.

Para o VAE, realizamos o ajuste em duas etapas. Primeiro, um grid inicial mais amplo:

```
latent_dims = [4, 8, 16]
betas = [0.1, 0.5, 1.0]
```

A partir dessa etapa, observamos que **o espaço latente de dimensão 8** foi consistentemente o melhor hiperparâmetro. Assim, realizamos um **segundo grid afunilado**, mantendo $latent\_dim = 8$ e testando novos valores de β para refinar a regularização:

```
betas = [1.2, 1.5, 2.0]
```

A Fig. 9 apresenta a matriz de confusão (teste) do VAE.

IV. Análise de Resultados

Antes da comparação final, destacamos as métricas centrais: **Average Precision (AP)**, **Recall@k** e **F2-score**.

A. Average Precision (AP)

A AP corresponde à área sob a curva *Precision-Recall* (PR), sendo especialmente adequada em cenários de **desbalanceamento extremo**. Em problemas como detecção de fraude, a AUC-ROC pode se manter alta mesmo quando o desempenho na classe minoritária é limitado. A curva PR e a AP focam diretamente no trade-off entre **precisão** e **revocação** na classe positiva, razão pela qual AP é amplamente utilizada em detecção de fraudes.

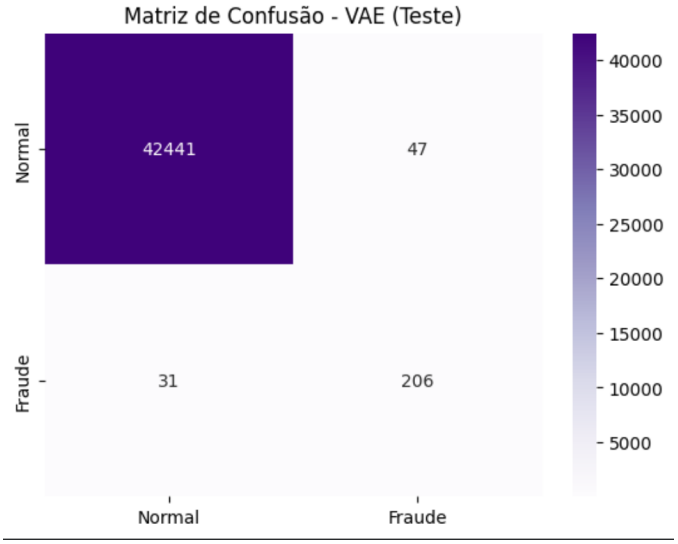

Figura 9. Matriz de confusão do VAE (teste).

B. *Recall@k*

O Recall@k mede a proporção de fraudes recuperadas entre as k transações mais suspeitas (ordenadas pelo escore do modelo). Essa métrica é importante operacionalmente: em muitos cenários existe um **orçamento de investigação**. Assim, Recall@k avalia diretamente a capacidade do modelo de **priorizar verdadeiros positivos** no topo do ranqueamento.

C. *F2-score*

O Fβ-score generaliza o F1 ao permitir controlar a importância relativa entre precisão e recall:

$$F_\beta = (1+\beta^2) \cdot \frac{(\text{Precisão} \cdot \text{Recall})}{(\beta^2 \cdot \text{Precisão}) + \text{Recall}}.$$

Ao escolher $\beta = 2$, o **F2-score** atribui maior peso ao **recall** do que à precisão. Adotamos F2 por refletir a priorização típica em fraude: frequentemente, é preferível **barrar uma transação legítima** do que **deixar passar uma fraude**.

D. *Tabela comparativa e discussão*

A Tabela I resume os resultados finais.

Tabela I
RESULTADOS COMPARATIVOS DOS MODELOS.

Modelo	Recall@500	AUC-ROC	AP	F2 (Fraude)
VAE	0.881857	0.966563	0.815623	0.857619
Autoencoder	0.835443	0.954991	0.761377	0.806314
LOF	0.831224	0.957766	0.607732	0.701981
Isolation Forest (FS)	0.691983	0.960584	0.429066	0.553609

Os resultados mostram que o **VAE** obteve o melhor desempenho global, liderando em AP e F2, além de apresentar o maior Recall@500. Isso sugere que o modelo ranqueia melhor as transações suspeitas, recuperando mais fraudes no topo da lista, e mantém alto foco em recall (via F2), o que é desejável no contexto do problema.

O **Autoencoder** apresenta desempenho muito competitivo, com AP elevada e F2 alto, reforçando que abordagens de reconstrução são fortes candidatas em cenários semi-supervisionados.

O **LOF** tem Recall@500 semelhante ao Autoencoder, mas AP e F2 inferiores aos modelos neurais, indicando que, apesar de capturar anomalias locais, sua priorização global (precision-recall ao longo do ranqueamento) é mais limitada neste caso.

Por fim, o **Isolation Forest (FS)** mantém AUC-ROC elevada, porém apresenta AP e F2 inferiores, sugerindo menor efetividade quando a avaliação enfatiza precisão-revocação na classe minoritária.

V. CONCLUSÃO

Este trabalho investigou detecção de anomalias em transações financeiras sob desbalanceamento extremo, comparando modelos probabilísticos (Isolation Forest), baseados em densidade (LOF) e redes neurais de reconstrução (Autoencoder e VAE). A análise exploratória evidenciou que, apesar de as features estarem em PCA, algumas variáveis (notadamente V14, V17 e V12) exibem padrões significativamente distintos entre classes. Em especial, a análise bivariada mostrou que transações normais formam um **cluster** denso, enquanto fraudes se distribuem amplamente no espaço (V14, V17), reforçando a heterogeneidade do comportamento fraudulento e a necessidade de métodos capazes de capturar padrões raros e dispersos.

Metodologicamente, destacam-se: (i) o treinamento semi-supervisionado, restringindo o treino à classe normal, (ii) a seleção de atributos por Random Forest com corte de 1% de importância (removendo 8 variáveis), reduzindo dimensionalidade e ruído, e (iii) a padronização da escolha de thresholds por **maximização do F2-score**. No conjunto de métricas, AP foi crucial por focar no trade-off precisão-revocação; Recall@500 aproximou o cenário de priorização real; e F2 explicitou a preferência por recuperar fraudes mesmo ao custo de alguns falsos positivos.

Os resultados indicaram que o **VAE** foi o modelo mais efetivo, sugerindo maior capacidade de capturar padrões complexos do comportamento normal e separar fraudes como eventos raros. O **Autoencoder** também apresentou desempenho robusto, evidenciando a força de modelos de reconstrução quando a classe positiva é rara. O **LOF** mostrou desempenho intermediário e reforçou a utilidade do modo *novelty detection* para tratar fraudes como novidades frente ao padrão normal. O **Isolation Forest**, embora eficiente e com boa AUC-ROC, foi menos competitivo em AP/F2, destacando a importância de métricas alinhadas ao objetivo real em desbalanceamento extremo.

a) Trabalhos Futuros: Como continuidade deste trabalho, algumas extensões se mostram relevantes. Uma direção natural é a investigação de modelos baseados em *Generative Adversarial Networks* (GANs) para detecção de anomalias, em especial abordagens como AnoGAN ou variantes condicionais, que podem aprender distribuições complexas do comportamento normal e fornecer escores de anomalia a partir

da qualidade da reconstrução. Outra possibilidade consiste na construção de *ensembles* combinando modelos baseados em densidade e reconstrução, explorando a complementaridade observada entre as abordagens. Além disso, a avaliação dos modelos sob métricas de custo explícito, incorporando diferentes penalidades para falsos positivos e falsos negativos, pode aproximar ainda mais a análise do cenário operacional real. Por fim, a adaptação dos modelos a cenários dinâmicos, considerando *concept drift* e aprendizado contínuo, representa um passo importante para aplicações em ambientes de produção.

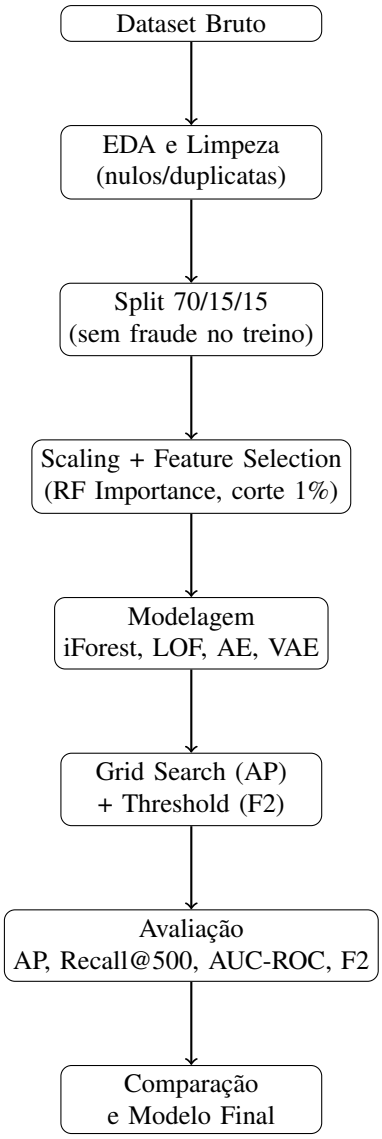

Figura 10. Arquitetura do pipeline do projeto de detecção de anomalias.

REFERÊNCIAS

[1] C. C. Aggarwal, *Outlier Analysis*. Springer, 2017.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng e J. Sander, "LOF: Identifying Density-Based Local Outliers," *SIGMOD*, 2000.

[3] F. T. Liu, K. M. Ting e Z.-H. Zhou, "Isolation Forest," *ICDM*, 2008.

[4] D. P. Kingma e M. Welling, "Auto-Encoding Variational Bayes," *ICLR*, 2014.

[5] I. Higgins et al., "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *ICLR*, 2017.

[6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *JMLR*, 2011.

[7] T. Saito e M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, 2015.

[8] A. Dal Pozzolo, O. Caelen, R. A. Johnson e G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," *IEEE Symposium Series on Computational Intelligence*, 2015.