PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

Érika Gonçalves de Assis

DeepSMOTE Tabular Optimize GAN: sobreamostragem de Dados Tabulares

Belo Horizonte

09 de maio de 2025

Érika Gonçalves de Assis

DeepSMOTE Tabular Optimize GAN: sobreamostragem de Dados Tabulares

Tese apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito para obtenção do título de Doutor em Informática.

Orientador: Profa. Dra. Cristiane Neri Nobre

Belo Horizonte

09 de maio de 2025

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Assis, Erika Gonçalves de
A848d DeepSMOTE tabular optimize GAN: sobreamostragem de dados tabulares / Erika Gonçalves de Assis. Belo Horizonte, 2025.
142 f. : il.

Orientadora: Cristiane Neri Nobre

Tese (Doutorado) - Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática

1. Algorítmos computacionais. 2. Aprendizado do computador. 3. Aprendizado Profundo. 4. Processamento de dados. 5. Estruturas de dados (Computação). 6. Otimização matemática. I. Nobre, Cristiane Neri. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática. III. Título.

CDU: 681.3.091

Ficha catalográfica elaborada por Fabiana Marques de Souza e Silva - CRB 6/2086

Érika Gonçalves de Assis

DeepSMOTE Tabular Optimize GAN: sobreamostragem de Dados Tabulares

Tese apresentada ao Programa de Pós-Graduação em Informática pela Pontifícia Universidade Católica de Minas Gerais, como requisito para o obtenção do título de Doutora em Informática.

Profa. Dra. Cristiane Neri Nobre – PUC Minas (Orientadora)

Prof Dr. Luiz Enrique Zárate Galvez – PUC Minas (Banca Examinadora)

Prof. Zenilton Kleber G. do Patrocínio Júnior – PUC Minas (Banca Examinadora)

Prof. Dr. Carlos Henrique da Silveira – UNIFEI (Banca Examinadora)

Prof. Dra. Anne Magaly de Paula Canuto – UFRN (Banca Examinadora)

Belo Horizonte, 09 Maio de 2025.

Às Marias, que serão ainda maiores

AGRADECIMENTOS

Este momento tão significativo da minha vida não seria possível sem o apoio e a dedicação de várias pessoas e instituições que estiveram ao meu lado, direta ou indiretamente, ao longo dessa jornada. Por isso, gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste trabalho.

Em primeiro lugar, quero agradecer à minha mãe, cujo amor incondicional e suporte constante foram o alicerce que me permitiu seguir em frente, mesmo nos momentos mais desafiadores. Sua força e encorajamento sempre me inspiraram a nunca desistir dos meus sonhos.

Ao meu irmão, por estar sempre ao meu lado, compartilhando palavras de motivação e ajudando a tornar os dias mais leves. Sua presença foi um porto seguro em meio às tempestades.

Às minhas queridas filhas, que, com tanto amor e compreensão, souberam lidar com a minha ausência em tantos momentos importantes. Vocês são minha maior motivação, e tudo o que faço é para que se sintam orgulhosas e inspiradas a seguir seus próprios caminhos. Obrigado por serem minha razão de lutar e perseverar.

Quero também expressar minha gratidão à minha fé e à minha espiritualidade, que se tornaram minha companhia constante e minha maior fonte de força. Não se trata de uma religião específica, mas de uma conexão profunda com algo maior que habita em mim e ao meu redor. Foi essa fé que me manteve firme nos momentos mais difíceis, que me deu esperança quando tudo parecia perdido e que me lembrou, todos os dias, de que eu não estava sozinha. Minha espiritualidade foi o abraço que me acolheu e a luz que me guiou, mesmo nas noites mais escuras.

A todos os amigos, colegas e professores que contribuíram de alguma forma para o desenvolvimento deste trabalho, meu sincero agradecimento. Cada conselho, cada palavra de incentivo e cada gesto de apoio foram fundamentais para que eu chegasse até aqui.

Gostaria de agradecer especialmente à DRH-UFMG, por meio do Programa de Incentivo à Qualificação em Nível de Pós-Graduação, pelo incentivo financeiro que foi fundamental para a realização desta etapa da minha formação. Esse apoio não apenas facilitou o acesso a recursos necessários, mas também reforçou minha motivação para seguir em frente.

À Diretoria de Tecnologia da Informação da UFMG, em especial ao Plano Anual de

Desenvolvimento (PLAD), pelo suporte e pelas oportunidades que me permitiram conciliar minha formação acadêmica com o desenvolvimento profissional.

Que este seja apenas o início de uma jornada repleta de conquistas e realizações.

Agradeço à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, códigos APQ-03104-24, APQ-05058-23 e APQ-03076-18) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, código 311573/2022-3) pelo apoio financeiro e pelas oportunidades concedidas, sem os quais este trabalho não seria possível.

"Finalmente, lembra-te que é melhor cumprir a própria tarefa, ainda que seja humilde e insignificante, do que querer fazer a tarefa de um outro, por mais nobre e excelente que seja. É melhor morrer no cumprimento do seu dever, do que viver negligenciando-o e querendo fazer o que a outros compete fazer"

Bhagavad Gita

RESUMO

O desequilíbrio de classes é um problema comum em muitas aplicações do mundo real, como na detecção de fraudes financeiras, diagnósticos médicos e manutenção preditiva, e exerce uma influência significativa no desempenho de algoritmos de aprendizado de máquina. Em cenários desequilibrados, a classe minoritária acaba sendo aprendida de forma insatisfatória, uma vez que os modelos tendem a favorecer a classe majoritária devido a sua maior representatividade nos dados de treinamento. Esse viés compromete a capacidade do modelo de generalizar corretamente para exemplos da classe de menor frequência, levando a métricas de desempenho insatisfatórias, como baixa precisão e Revocação. Para mitigar o problema do desbalanceamento de classes, diversas técnicas de balanceamento de dados têm sido propostas. Estas podem ser categorizadas em quatro grupos principais: *nível de dados*, *nível de algoritmo*, *híbridos* e *modelos generativos*. Os métodos em nível de dados atuam diretamente nos dados de entrada alterando o tamanho da amostra, e podem ser divididos em dois tipos: subamostragem (removem exemplos da classe majoritária) e sobreamostragem (criação de amostras sintéticas da classe minoritária. Já os métodos em nível de algoritmo, os ajuste são realizados nos algoritmos de aprendizado para dar maior peso às instâncias da classe minoritária durante o treinamento. Os métodos híbridos são a combinação de técnicas de nível de dados e de algoritmo para obter um melhor desempenho. Modelos generativos, como GANs (Generative Adversarial Networks) e VAEs (Autoencoders Variacionais), são utilizados para criar novos exemplos sintéticos da classe minoritária. Neste trabalho, focamos na aplicação de modelos generativos para o balanceamento de dados tabulares. Propomos o DeepSMOTE Tabular Optimize GAN (DSTO-GAN), uma adaptação do DeepSMOTE que incorpora uma GAN para gerar exemplos sintéticos mais realistas para dados tabulares. A complexidade dos dados tabulares, com atributos numéricos e categóricos, exige que os métodos de balanceamento preservem as relações entre os atributos. Os resultados da pesquisa demonstram que o DSTO-GAN, apresenta contribuições significativas no balanceamento de dados, especialmente em problemas binários e conjuntos categóricos, superando abordagens tradicionais na preservação da estrutura dos dados e no desempenho de classificadores. O DSTO-GAN apresentou melhor desempenho, particularmente em conjuntos de dados com dimensões moderadas (500-5.000 instâncias e 10-50 atributos), onde alcançou F1-score $\geq$ 0,95. O método mostrou-se especialmente eficaz em cenários com alto desbalanceamento (IR $\geq$ 30), com (F1-score = 1,00) em diversos casos, como evidenciado nos conjuntos*Absenteeism_at_work* (IR=66,3) e *Abalone* (IR=32,5). No entanto, sua eficácia diminui em cenários de desbalanceamento extremo ou dados multiclasse, além de demandar alto custo computacional, limitando

aplicações em tempo real. Os resultados indicam que abordagens híbridas e otimizações arquiteturais são promissoras para avanços futuros, reforçando a necessidade de uma seleção contextualizada das técnicas, adaptada às particularidades de cada problema.

Palavras-chave: Redes Adversárias Generativas, DCGAN, Deep SMOTE, sobreamostragem, dados tabulares.

ABSTRACT

Class imbalance is a common problem in many real-world applications, such as financial fraud detection, medical diagnosis, and predictive maintenance, and it significantly influences the performance of machine learning algorithms. In imbalanced scenarios, the minority class learns poorly since models tend to favor the majority class due to its more excellent representation in the training data. This would require the model to generalize correctly to lower-frequency class examples, leading to poor performance metrics, such as low precision and recall. Several data balancing techniques have been proposed to mitigate the problem of class imbalance. These can be categorized into four main groups: data-level, algorithm-level, hybrid, and generative models. Data-level methods act directly on the input data by changing the sample size. They can be divided into two types: undersampling (removing examples from the majority class) and oversampling (creating synthetic samples from the minority class). On the other hand, algorithm-level methods adjust the learning algorithms to provide further weight to instances from the minority class during training. Hybrid methods combine data-level and algorithm-level techniques to obtain better generative models. Generative models, such as GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders), create new synthetic examples from the minority class. This work focuses on applying generative models for balancing tabular data. We propose the DeepSMOTE Tabular Optimize GAN (DSTO-GAN), an adaptation of DeepSMOTE incorporating a DCGAN to generate more realistic examples for tabular data. The complexity of tabular data, with numerical and categorical data, requires that balancing methods preserve the relationships between attributes. DSTO-GAN meets this need by generating examples that reflect the original data distribution and contributing to a more effective balancing of classes. The research results demonstrate that DSTO-GAN presents significant contributions in data balancing, especially in binary problems and categorical sets, outperforming traditional approaches in preserving data structure and classifier performance. DSTO-GAN performed best, especially on moderate-dimensional datasets (500-5000 instances and 10-50 attributes), where it achieved an F1-score $\geq$ 0,95. The method was especially effective in highly imbalanced scenarios (IR $\geq$ 30), with (F1-score = 1,00) in several cases, as evidenced in the sets *Absenteeism_at_work* (IR=66,3) and *Abalone* (IR=32,5). However, its effectiveness decreases in scenarios of extreme imbalance or multiclass data, in addition to demanding high computational cost, limiting real-time applications. The results indicate that hybrid approaches and architectural optimizations are promising for future advances, reinforcing the need for a contextualized selection of techniques, adapted to the particularities of each

problem.

Keywords: Generative Adversary Networks, DCGAN, Deep SMOTE, Oversampling, Tabular Data

LISTA DE FIGURAS

LISTA DE TABELAS

ABREVIATURAS E SIGLAS

GAMO *Generative Adversarial Minority Oversampling*

SUMÁRIO

1 INTRODUÇÃO

A crescente disponibilidade de dados, impulsionada por tecnologias como a Internet das Coisas (IoT) (MRABET et al., 2020) e redes sociais (DING et al., 2023), tem proporcionado um ambiente fértil para o desenvolvimento de novas abordagens de aprendizado de máquina e mineração de dados. Essa vasta quantidade de informações tornou a análise e classificação de dados fundamentais para as transformações tecnológicas e comerciais, despertando o interesse de pesquisadores de diversas áreas (DING et al., 2023).

Além disso, o reconhecimento da importância estratégica dos dados nas organizações e na sociedade como um todo tem elevado a análise e a classificação de dados ao *status* de prioridade (VELARDE et al., 2024). Empresas estão cada vez mais dependendo de análises de dados para informar decisões de negócios, desde *marketing*, vendas até operações e desenvolvimento de produtos. Da mesma forma, na academia, os pesquisadores estão explorando dados para avançar em campos diversos, tais como medicina, ciências sociais e ciência da computação (JAIN et al., 2022).

A classificação é uma importante área de pesquisa em mineração de dados e aprendizado de máquina. A aprendizagem supervisionada, um método comum em tarefas de classificação, é fundamentada no treinamento de algoritmos com conjuntos de dados rotulados, visando construir uma função de mapeamento eficiente.

Como descrito na Equação 1.1, considerando um conjunto de dados D com m amostras, onde cada amostra é um par (x_i, y_i), tem-se que x_i possui um rótulo correspondente y_i. A equação estabelece que $x_i \in X$ e $y_i \in Y$, onde X representa as amostras de treinamento (observações) e Y os respectivos rótulos (ou classes) (DING et al., 2023).

$$D = \{(x_i, y_i) | x_i \in X,\ y_i \in Y,\ i = 1, ..., m\} \tag{1.1}$$

Portanto, na aprendizagem supervisionada, o modelo utiliza os dados para compreender a relação entre as observações X e os rótulos Y, permitindo que ele generalize e faça previsões de rótulos para novas amostras que não foram vistas durante o treinamento.

No entanto, sabe-se que a aplicação de métodos convencionais de aprendizado de máquina em tarefas do mundo real torna-se problemática quando há desequilíbrio na distribuição das classes. A desproporção entre o número de observações majoritárias e minoritárias influencia o processo de otimização em relação à função de perda, levando a um viés para a classe majoritária e degradação associada das capacidades preditivas para as classes minoritárias (KHAN; CHAUDHARI; CHANDRA, 2024). Isso pode resultar em modelos menos precisos e eficazes, especialmente para a detecção de eventos raros ou anômalos (KOZIARSKI; WOZNIAK; KRAWCZYKZ, 2020).

Em muitos cenários, é comum encontrar conjuntos de dados em que uma classe é significativamente mais predominante do que outras. Por exemplo, em problemas de detecção de fraudes, é provável que a maioria das transações seja legítima, enquanto apenas uma pequena fração será fraudulenta (ISLAM et al., 2023) (ORESKI, 2023). Da mesma forma, em um conjunto de dados médicos, a maioria dos pacientes pode ser saudável, enquanto a minoria tem uma doença (DUBEY et al., 2014). Na classificação de *spam*, a maioria dos *e-mails* recebidos pode ser legítima, com apenas uma pequena porção sendo *spam* (MULLICK; DATTA; DAS, 2019).

Nestes contextos, a classe minoritária tende a ser sub-representada durante o treinamento. Isso resulta em um modelo que não aprende adequadamente as características da classe minoritária, prejudicando a capacidade de prever instâncias pertencentes a essa classe (MULLICK; DATTA; DAS, 2019). Por exemplo, se uma classe representa apenas 2% do total de instâncias, um classificador poderia alcançar uma acurácia de 98% simplesmente atribuindo todas as instâncias à classe majoritária. Contudo, o referido classificador falharia completamente em discernir a classe minoritária, a qual frequentemente é de interesse primordial (MULLICK; DATTA; DAS, 2019).

Neste sentido, o balanceamento de dados é uma técnica fundamental para mitigar esse problema, garantindo que os modelos de aprendizado de máquina sejam capazes de aprender representações mais equitativas das diferentes classes. Tecnicamente, qualquer conjunto de dados que apresenta uma discrepância na distribuição entre as classes pode ser considerado desbalanceado. No entanto, na comunidade científica, o entendimento comum é que dados desbalanceados referem-se a conjuntos de dados que demonstram desequilíbrios consideráveis e, em certos casos, extremos. Especificamente, essa forma de disparidade é denominada desequilíbrio entre classes; desequilíbrios na ordem de 100:1, 1.000:1 e 10.000:1 não são incomuns, nos quais uma classe prevalece significativamente sobre a outra (HE; GARCIA, 2009).

Para o tratamento de desbalanceamento de classes, um conjunto diversificado de técnicas, abrangendo desde manipulações nos dados, como subamostragem e sobreamostragem, até adaptações nos algoritmos de aprendizado, têm sido utilizadas. Além disso, métodos híbridos, que combinam múltiplas abordagens, e a geração de dados sintéticos por meio de modelos generativos, como as Redes Adversariais Generativas (GANs), oferecem soluções inovadoras para esse problema.

Para uma melhor compreensão das abordagens de balanceamento de classes em problemas de aprendizado de máquina, é essencial destacar que os métodos atuais podem ser classificados em quatro categorias principais, conforme proposto por (KRAWCZYK, 2016) e (FERNÁNDEZ, 2018): métodos em *nível de dados*, em *nível de algoritmo*, *híbridos* e *generativos*. Essa categorização organiza as estratégias de acordo com o estágio em que

atuam no processo de modelagem e a forma como abordam o desbalanceamento de classes.

Os métodos em *nível de dados* alteram o tamanho da amostra antes do treinamento do modelo e podem ser divididos em sobreamostragem e subamostragem. Na sobreamostragem há a criação das amostras sintéticas da classe minoritária. Destaca-se os seguintes algoritmos: SMOTE (*Synthetic Minority Over-sampling Technique*) (CHAWLA et al., 2002), ADASYN (*Adaptive Synthetic Sampling*) (Haibo He et al., 2008) e *Random Oversampling* (DOUZAS; BACAO, 2018). Na subamostragem removem-se exemplos da classe majoritária até que as classes estejam balanceadas, por exemplo os algoritmos RUS *Random Undersampling* (KUBAT; MATWIN et al., 1997), *NearMiss* (MANI; ZHANG, 2003) e *Tomek Links* (TOMEK, 1976).

Em *nível de algoritmo*, os métodos modificam os algoritmos de aprendizado de máquina para lidar melhor com conjuntos de dados desequilibrados sem alterar diretamente os dados, como, por exemplo, os algoritmos SVM (SCHöLKOPF et al., 2001) e o *Cost-Sensitive Learning* que atribuem pesos diferentes às classes durante o treinamento, dando mais importância à classe minoritária (LOEZER et al., 2020).

Os *métodos híbridos* integram técnicas de balanceamento em nível de dados e de algoritmo, visando superar as limitações individuais de cada abordagem e melhorar o desempenho. Esses métodos podem combinar estratégias de sobreamostragem e subamostragem, como no caso do *SMOTE-ENN* (CHAWLA et al., 2002) (Wilson, 1972), ou do *SMOTE Tomek Links* (PRATI; BATISTA; MONARD, 2004). Além disso, os métodos híbridos também podem unir técnicas de balanceamento em nível de dados com ajustes em nível de algoritmo, como por exemplo: *SMOTE +Cost-Sensitive Learning* (ALMHAITHAWI; JAFAR; ALJNIDI, 2020), que combina a geração de amostras sintéticas com a atribuição de custos diferenciados para erros de classificação, ou pelo *SMOTEBoost* (CHAWLA et al., 2003), que integra SMOTE ao algoritmo *AdaBoost* para priorizar a classificação correta da classe minoritária. Outro exemplo é o *RUSBoost* (SEIFFERT et al., 2010), que alia RUS ao AdaBoost, reduzindo o viés em direção à classe majoritária enquanto mantém a eficácia do *boosting*.

Os modelos *generativos* distinguem-se das técnicas tradicionais de sobreamostragem por sua capacidade de gerar novos exemplos de dados sintéticos de forma mais sofisticada, aprendendo diretamente a distribuição dos dados. Enquanto métodos convencionais, como o SMOTE, criam amostras sintéticas por meio de interpolações lineares entre instâncias existentes, os modelos generativos utilizam abordagens baseadas em aprendizado profundo, como Redes Adversariais Gerativas (GANs) (GOODFELLOW et al., 2014) e *Autoencoders* Variacionais (VAEs) (KINGMA; WELLING, 2014), para produzir amostras que capturam a complexidade e a variabilidade dos dados reais. Essas técnicas não apenas equilibram as classes, mas também preservam a estrutura intrínseca dos dados, resultando em conjuntos de dados sintéticos mais realistas e diversificados.

A escolha da técnica de balanceamento de dados depende de vários fatores, incluindo o tipo de problema, a natureza do desbalanceamento e os recursos computacionais disponíveis. O tipo de problema está relacionado ao domínio de aplicação e aos objetivos específicos da tarefa de classificação, como detecção de fraudes, diagnóstico médico ou previsão de falhas, nos quais o impacto de falsos positivos e falsos negativos pode variar significativamente. A natureza do desbalanceamento, por sua vez, envolve a proporção entre as classes, a distribuição dos dados no espaço de características e a presença de ruídos ou sobreposição entre as classes, aspectos que influenciam a eficácia das técnicas de balanceamento. Diante dessas variáveis, é fundamental testar diferentes abordagens e avaliar seus resultados por meio de métricas adequadas.

1.1 Motivação

A motivação desta pesquisa surgiu da necessidade de enfrentar o desafio do aprendizado de máquina em conjuntos de dados desbalanceados, um problema recorrente e amplamente estudado devido às suas implicações práticas em diversos domínios, como financeira (ZHU et al., 2023; ORESKI, 2023), redes de computadores (XU et al., 2020; CHEN; CHIANG; HUANG, 2022), IoT (HABIBI; CHEMMAKHA; LAZAAR, 2023; PARFENOV et al., 2023), sistemas de recomendação (SHAFQAT; BYUN, 2022), segurança da informação (MACI et al., 2023), indústria (YUAN et al., 2023), engenharia (SUN; WANG; CHU, 2023), medicina (SOLEIMANI et al., 2023; RODRIGUEZ-ALMEIDA et al., 2023), logística (MOHAMMADPOUR; KHEDMATI; ZADA, 2023), dentre outros. Especificamente, este estudo foca na manipulação de dados tabulares, que se tornaram cada vez mais relevantes no cenário moderno, à medida que as organizações estão implementando aprendizado de máquina em dados relacionais para automatizar e melhorar processos antes realizados manualmente (XU; VEERAMACHANENI, 2018). Segundo Wang et al. (2024), os dados tabulares são amplamente utilizados em ambientes empresariais, sendo o formato mais comum nos negócios e o segundo mais utilizado no meio acadêmico, perdendo apenas para dados textuais, que são predominantes em áreas como processamento de linguagem natural (NLP) e mineração de textos. Esses dados desempenham um papel crucial em setores diversos como sistemas de detecção de intrusões (DONG et al., 2021), sistemas de recomendação (SHAFQAT; BYUN, 2022), falha de motores de indução (HEJAZI; PACKIANATHER; LIU, 2023), detecção de fraudes (ZHANG; ZHANG, 2017), diagnósticos médicos (SUN et al., 2024), empréstimos bancários (UDDIN et al., 2023), classificação de padrões (XU et al., 2020), dentre muitas outras aplicações práticas.

1.2 Problema

O problema central desta tese é o desequilíbrio de classes em conjuntos de dados tabulares, um obstáculo significativo que prejudica a eficácia dos modelos de aprendizado de máquina (LIU et al., 2023). Esse desbalanceamento é particularmente desafiador em cenários onde a classe de interesse, frequentemente a classe minoritária, está significativamente sub-representada, resultando em um desempenho insatisfatório dos algoritmos de classificação (XU; VEERAMACHANENI, 2018).

A maioria dos algoritmos de aprendizado de máquina parte da premissa de que os dados de treinamento são equilibrados. Isso faz com que os modelos gerados tendam a generalizar bem para a maioria dos exemplos, mas enfrentam dificuldades em classificar corretamente os exemplos pertencentes à classe minoritária. Esse viés para a classe majoritária faz com que os algoritmos falhem ao alcançar um desempenho ideal quando aplicados a conjuntos de dados desbalanceados, especialmente em contextos onde a identificação da classe minoritária é crítica (XU et al., 2019a).

Os dados tabulares, amplamente utilizados em setores como negócios, saúde e finanças, apresentam desafios específicos devido à sua natureza heterogênea. Diferente de dados como imagens ou textos, onde as correlações são mais evidentes e homogêneas, os dados tabulares costumam incluir uma combinação de variáveis contínuas e categóricas, muitas vezes com distribuições complexas e assimétricas. Isso dificulta a coleta, codificação, síntese e avaliação desses dados de forma padronizada, além de tornar mais difícil o balanceamento de classes minoritárias (KHAN; CHAUDHARI; CHANDRA, 2024).

Assim, ao contrário de conjuntos de dados de imagens, onde os valores de *pixels* são frequentemente modelados — embora nem sempre de forma ideal — por uma distribuição gaussiana, os recursos contínuos presentes em dados tabulares muitas vezes não seguem essa distribuição padrão. Em vez disso, apresentam distribuições mais complexas, como distribuições multimodais ou caudas longas Essa variação cria dificuldades adicionais para algoritmos de aprendizado de máquina, que frequentemente são ajustados para lidar com dados mais homogêneos e previsíveis, como imagens (XU et al., 2019a). Outro obstáculo é a modelagem da distribuição de probabilidade em dados tabulares. A geração de dados sintéticos que sejam representativos e realistas é um desafio devido à coexistência de colunas discretas e contínuas (XU et al., 2019a).

Portanto, o desafio reside em encontrar estratégias eficazes que abordem o desbalanceamento de classes em dados tabulares, sem comprometer a integridade e a complexidade intrínseca desses dados. A capacidade de gerar dados sintéticos realistas e bem balanceados, que preservem as correlações e estruturas originais, é fundamental para melhorar o desempenho dos algoritmos de aprendizado de máquina em cenários com dados desbalanceados (XU et al., 2019a).

Ao longo dos anos, diversas técnicas de balanceamento de dados foram desenvolvidas para mitigar essas limitações (KHAN; CHAUDHARI; CHANDRA, 2024). Mais recentemente, novas abordagens vêm ganhando destaque, incluindo o uso de modelos generativos, uma classe de algoritmos de aprendizado de máquina projetados para gerar novos dados que preservam a distribuição de um conjunto existente. Diferentemente dos modelos discriminativos, que aprendem a distinguir entre classes, os modelos generativos capturam as características intrínsecas dos dados, permitindo a criação de amostras realistas. Entre os modelos generativos mais reconhecidos, destacam-se os *Autoencoders* Variacionais (KINGMA; WELLING, 2014) e as Redes Adversárias Generativas (GOODFELLOW, 2017).

Um VAE possui dois componentes principais: um codificador e um decodificador, que são treinados conjuntamente para aprender a distribuição dos dados reais e representá-los em um espaço latente de menor dimensão. O codificador do VAE gera parâmetros de uma distribuição predefinida no espaço latente para cada entrada. O VAE então impõe uma restrição a essa distribuição latente, forçando-a a ser uma distribuição normal, gerando pontos sintéticos que seguem a distribuição dos dados reais (KHADKA et al., 2023).

As GANs, propostas Goodfellow et al. (2014), são compostas por dois componentes principais: um gerador e um discriminador. O gerador cria amostras sintéticas, enquanto o discriminador avalia essas amostras em comparação com os dados reais, resultando em uma competição entre os dois modelos. Esse processo de treinamento adversarial permite que as GANs produzam amostras que, gradualmente, se tornam indistinguíveis das amostras reais, proporcionando uma poderosa ferramenta para gerar dados sintéticos em diferentes domínios, como imagens, texto e dados tabulares (GOODFELLOW, 2017).

As GANs têm se mostrado promissoras para a sobreamostragem de dados, inclusive em cenários com dados tabulares. Soluções adaptadas, como as *Conditional GANs* (CGANs) (MIRZA; OSINDERO, 2014) e variantes específicas para dados tabulares, como CTGAN (MAHINNEZHAD et al., 2024) e TableGAN (HU et al., 2021), têm sido exploradas para atender a essas particularidades. O uso de GANs em dados tabulares visa criar um conjunto de dados balanceado que preserve as propriedades estatísticas dos atributos originais e melhore o desempenho dos algoritmos de aprendizado em cenários de classes desbalanceadas.

Assim, a utilização de modelos generativos para sobreamostragem de dados tabulares não só amplia as possibilidades de geração de dados sintéticos, como também oferece uma abordagem robusta para lidar com desbalanceamento de classes em aplicações práticas.

Assim, as questões de pesquisa propostas para este trabalho buscam explorar e aprimorar o uso de modelos generativos para o balanceamento de conjuntos de dados tabulares, considerando suas particularidades e desafios.

QP1: Quais modificações e adaptações são permitidas em modelos generativos para melhorar o balanceamento de conjuntos de dados tabulares, levando em consideração suas características heterogêneas e complexas?

QP2: Como o desempenho de métodos generativos se compara com outras abordagens de balanceamento em termos de eficácia no balanceamento de classes e na acurácia de modelos de classificação?

QP3: Quais são as configurações ótimas de hiperparâmetros e condições de aplicação (tamanho do conjunto de dados, dimensionalidade e nível de desbalanceamento) que maximizam o desempenho, a estabilidade e a eficiência computacional de modelos generativos, em cenários práticos de classificação desbalanceada?

QP4: Em quais tipos de conjuntos de dados (numéricos ou categóricos) os métodos baseados em modelos generativos apresentam um desempenho superior em relação a outras técnicas de balanceamento?

Essas questões orientam o estudo na busca de melhorias e inovações que possibilitem uma solução mais robusta e eficiente para o problema do desbalanceamento em dados tabulares.

1.3 Hipóteses

A adaptação de algoritmos baseados em modelos generativos para o tratamento de dados tabulares heterogêneos e complexos, especialmente em cenários de desbalanceamento de classes, demonstra potencial para superar abordagens tradicionais de balanceamento.

A partir da hipótese geral apresentada, foram propostas as seguintes hipóteses especificas:

H1: É possível adaptar algoritmos baseados em modelos generativos para conjuntos de dados tabulares, preservando sua eficácia em contextos de dados heterogêneos e complexos.

H2: Métodos de balanceamento de dados baseados em modelos generativos podem superar outras abordagens de balanceamento em termos de eficácia no balanceamento de classes e precisão de modelos de classificação.

H3: Métodos de balanceamento baseados em modelos generativos tendem a apresentar melhores resultados em conjuntos de dados predominantemente numéricos quando comparados a outras abordagens de balanceamento.

H4: O índice de desbalanceamento (IR) pode ter uma influência direta no desempenho de algoritmos baseados modelos generativos. A hipótese é que quanto maior o desbalanceamento, melhor o desempenho desses métodos em ajustar a distribuição das classes.

Essas hipóteses de pesquisa norteiam as etapas experimentais da pesquisa, buscando validar a eficácia do uso de modelos generativos para sobreamostragem de dados tabulares em diferentes contextos de desbalanceamento de classes.

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo deste trabalho é desenvolver um método eficaz de balanceamento para conjuntos de dados tabulares, fundamentado na aplicação de modelos generativos, especificamente Redes Adversárias Generativas e Autoencoders Variacionais.

A proposta central é avaliar a eficácia dessas técnicas na mitigação do problema de classes desequilibradas, uma limitação comum em diversas aplicações do mundo real. Busca-se desenvolver uma solução que leve em consideração as particularidades desse tipo de base de dados, melhorando o desempenho dos classificadores, reduzindo o viés em favor da classe majoritária e contribuindo para a construção de modelos preditivos mais precisos em cenários desbalanceados.

1.4.2 Objetivos Específicos

Para alcançar os objetivos gerais, foram estabelecidos os seguintes objetivos específicos:

a) Investigar o comportamento de métodos de balanceamento baseados em modelos generativos em função das particularidades dos conjuntos de dados, tais como dimensionalidade, número de classes, grau de desbalanceamento e natureza dos dados (numéricos ou categóricos).
b) Realizar uma análise comparativa sistemática dos métodos de sobreamostragem com modelos generativos em relação a outros métodos de balanceamento, garantindo uma análise abrangente do desempenho relativo da abordagem proposta.

Com isso, busca-se investigar o comportamento dos algoritmos baseados em modelos generativos como método de balanceamento para dados tabulares, através da comparação desses algoritmos com métodos de balanceamento convencionais.

1.5 Justificativa

Para abordar o desafio da sobreamostragem em dados tabulares, especialmente em cenários de desequilíbrio de classes, os modelos generativos híbridos que combinam VAEs e GANs representam uma solução inovadora.

As GANs, embora poderosas, exigem grandes volumes de dados e apresentam desafios em seu treinamento. O gerador, por vezes, mapeia diferentes pontos de entrada para uma

única saída, concentrando amostras em regiões limitadas do espaço de características. O discriminador, por sua vez, ao reconhecer esse padrão, ajusta-se, o que pode desencadear um ciclo de ajustes que prejudica a convergência do modelo e a eficácia na geração de dados diversos (GULRAJANI et al., 2017).

Os VAEs se mostram complementares, pois aprendem uma representação latente contínua do espaço de dados, permitindo a geração estável de novas amostras a partir dessa representação. O processo de otimização das VAEs, que foca na reconstrução dos dados e na aproximação da distribuição latente a uma gaussiana, evita os problemas de instabilidade e colapso de modo que afligem as GANs. No entanto, a qualidade das amostras geradas por VAEs pode ser inferior à das GANs, pois o objetivo principal das VAEs é aprender uma distribuição de dados, e não necessariamente gerar instâncias indistinguíveis dos dados reais (KHADKA et al., 2023).

A união dessas duas arquiteturas em um modelo generativo híbrido capitaliza as forças de cada uma enquanto mitiga suas fraquezas individuais. A VAE pode ser utilizada para aprender uma representação latente robusta e estável dos dados, fornecendo um espaço onde a geração de novas amostras pode ser mais controlada e diversificada. Subsequentemente, as GANs podem ser empregadas para refinar a qualidade das amostras geradas a partir desse espaço latente. Isso significa que o gerador da GAN não precisaria aprender a estrutura de dados complexa do zero, mas sim aprimorar a qualidade das amostras já representadas em um espaço latente bem-comportado pela VAE. Esse arranjo híbrido promove um treinamento mais estável para a GAN e resulta na geração de dados sintéticos de alta qualidade e com maior diversidade, cruciais para aprimorar a capacidade de discriminação em cenários de desequilíbrio de classes.

1.6 Contribuições

As contribuições desta tese são centradas na investigação e exploração do uso de modelos generativos para o balanceamento de classes em dados tabulares desequilibrados. Com isso, a pesquisa pretende oferecer contribuições relevantes tanto para o entendimento quanto para o aprimoramento de técnicas de balanceamento em dados complexos, indo além de uma abordagem meramente técnica.

A validação experimental em diversos cenários de dados desbalanceados agrega uma contribuição prática significativa, fornecendo evidências concretas de sua aplicabilidade e potencial de uso em diferentes domínios. Isso oferece uma avaliação abrangente da eficácia do uso de modelos generativos em relação a outras estratégias de balanceamento, contribuindo para a melhor compreensão de suas vantagens e limitações.

Desta forma, esta tese propõe uma nova abordagem para o balanceamento de classes em

dados tabulares desequilibrados, utilizando modelos generativos adversariais e *Variational Autoencoders*. Ao gerar novos exemplos sintéticos da classe minoritária, buscamos diminuir o viés dos algoritmos de aprendizado de máquina e melhorar a capacidade de generalização dos modelos. Nossos experimentos em diversos conjuntos de dados reais demonstram a eficácia da técnica proposta, especialmente em cenários com alto desbalanceamento e complexidade de dados.

Desenvolvemos e publicamos no *PyPI* a biblioteca *open-source* DSTO-GAN para balanceamento de dados `https://pypi.org/project/dsto-gan/`. A disponibilização do código e documentação detalhada visa impulsionar a pesquisa e o desenvolvimento de soluções mais precisas para classificação em dados desbalanceados.

1.7 Organização da tese

Esta tese está organizada de maneira sistemática e detalhada, com cada capítulo dedicado a um aspecto específico da pesquisa, visando proporcionar uma compreensão clara e abrangente do estudo realizado.

Assim, no Capítulo 2 são introduzidos os conceitos e teorias essenciais para a investigação sobre o balanceamento de dados tabulares. São apresentados os fundamentos teóricos que sustentam a pesquisa, incluindo definições, e princípios necessários para o entendimento do problema em questão.

O Capítulo 3 é dedicado à revisão e catalogação dos estudos correlatos à pesquisa. Nele, são analisados trabalhos anteriores que abordam temas semelhantes ou complementares ao balanceamento de dados tabulares. A revisão inclui uma discussão sobre as metodologias, técnicas e resultados obtidos por outros pesquisadores, permitindo identificar as contribuições existentes e as limitações que ainda precisam ser superadas. Essa análise crítica serve para posicionar o presente estudo dentro do cenário científico atual.

O Capítulo 4 apresenta o algoritmo DeepSMOTE Tabular Optimize GAN, desenvolvido como parte central desta pesquisa. A descrição inclui os princípios teóricos que embasam o algoritmo, sua arquitetura, funcionamento e as inovações propostas em relação às técnicas existentes. São detalhados os processos de otimização e integração com redes generativas adversariais (GANs), bem como as vantagens e aplicabilidades do método proposto para o balanceamento de dados tabulares.

O Capítulo 5 descreve, de forma detalhada, os materiais, métodos e procedimentos utilizados ao longo da pesquisa. Inclui a explicação das ferramentas, algoritmos e técnicas empregadas, bem como o planejamento experimental, a preparação dos dados e as etapas metodológicas adotadas para assegurar a validade e confiabilidade dos resultados.

O Capítulo 6 expõe os resultados obtidos a partir dos experimentos conduzidos com

o algoritmo proposto. Os dados são analisados de forma crítica, com discussões que relacionam os achados aos objetivos da pesquisa e às expectativas teóricas. São exploradas as implicações práticas dos resultados, suas limitações e como eles contribuem para o avanço do conhecimento na área. Gráficos, tabelas e outras formas de visualização de dados são utilizados para facilitar a compreensão dos resultados.

O Capítulo 7 apresenta um estudo de caso detalhado da aplicação do método DSTO-GAN ao conjunto de dados reais denominado RESP-Microcefalia.

Finalmente, o Capítulo 8 sintetiza as principais conclusões da pesquisa, destacando as contribuições científicas e práticas do estudo. São apresentadas reflexões sobre os desafios enfrentados e as lições aprendidas ao longo do processo. Além disso, são sugeridas direções para trabalhos futuros, indicando possíveis aprimoramentos do algoritmo desenvolvido, novas abordagens para o balanceamento de dados tabulares e outras áreas de investigação que podem ser exploradas com base nos resultados obtidos.

2 REFERENCIAL TEÓRICO

Este Capítulo explora a evolução do balanceamento de dados ao longo dos anos, destacando as principais categorias de algoritmos: métodos de *nível de dados*, *nível de algoritmo*, *híbridos* e *generativos*. Cada seção é dedicada a explicar e exemplificar técnicas representativas dessas categorias, oferecendo um embasamento teórico para o entendimento das abordagens existentes.

2.1 Evolução do Balanceamento de Dados

Nos anos 1960 e 1970, muitos algoritmos de aprendizado de máquina foram projetados para trabalhar com conjuntos de dados balanceados, assumindo uma distribuição uniforme entre as classes (TAREKEGN; GIACOBINI; MICHALAK, 2021).

No entanto, a aplicabilidade desses algoritmos em dados reais, onde é comum encontrar situações em que uma classe possui uma quantidade substancialmente maior de amostras em comparação com outras, expôs uma limitação significativa quanto ao viés para as classes majoritárias. Como resultado, esses algoritmos geralmente apresentavam baixa precisão na detecção de eventos menos comuns, o que impactava diretamente a qualidade dos modelos e suas aplicações (KHAN; CHAUDHARI; CHANDRA, 2024).

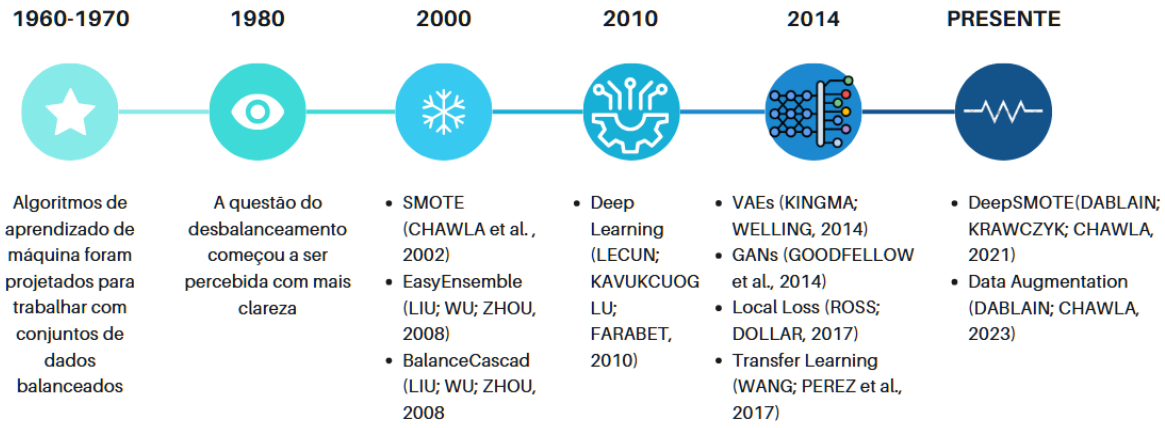
A questão do desbalanceamento começou a ser percebida com mais clareza nos anos 1980, à medida que modelos supervisionados passaram a ser aplicados em problemas de classificação onde as classes minoritárias tinham baixa representatividade (REZVANI; WANG, 2023). Isso se tornou um desafio evidente em problemas de reconhecimento de padrões, onde era comum que dados coletados apresentassem uma distribuição desigual de classes. Foi também nessa época que a pesquisa em redes neurais artificiais começou a destacar como esses modelos tendiam a priorizar as classes majoritárias, ignorando ou mal interpretando as instâncias de classes minoritárias (JAPKOWICZ et al., 2000).

Chamando a atenção para a necessidade de adaptações em algoritmos que pudessem corrigir esses problemas, Chawla et al. (2002) introduziram a técnica SMOTE (*Synthetic Minority Over-sampling Technique*) como uma forma de realizar *oversampling* em dados desbalanceados. Esse método se tornou um marco no desenvolvimento de técnicas de balanceamento, permitindo gerar novas instâncias sintéticas da classe minoritária para equilibrar a distribuição dos dados sem perder variabilidade, algo que era um problema nas técnicas de duplicação de dados (*oversampling aleatório*) da época.

Na década de 2000, houve uma crescente integração de técnicas de balanceamento com métodos de aprendizado *ensemble*. Essas abordagens combinavam subamostragem ou sobreamostragem com múltiplos modelos para melhorar a robustez e a generalização.

Métodos como *EasyEnsemble* (LIU; WU; ZHOU, 2008) e *BalanceCascade* (LIU; WU; ZHOU, 2008) foram propostos para criar conjuntos de modelos treinados em subconjuntos balanceados dos dados. A Figura 1 apresenta a linha cronológica dos métodos de balanceamento.

Figura 1: Linha cronológica das abordagens para balanceamento de dados



Fonte: Elaborada pela autora

Em meados de 2010, com a popularização do *Deep Learning* (LECUN; KAVUKCUOGLU; FARABET, 2010), o desafio do desbalanceamento de classes ganhou maior relevância, já que modelos profundos dependem de grandes quantidades de dados bem distribuídos para alcançar um bom desempenho e generalização. Essa dependência incentivou a criação de técnicas mais sofisticadas para lidar com conjuntos de dados desbalanceados. Entre as abordagens mais atuais, destaca-se o uso de aprendizado adversarial (GOODFELLOW et al., 2014), no qual redes neurais generativas (GANs) são utilizadas para criar dados sintéticos que complementam as classes minoritárias, ajudando a equilibrar a distribuição dos dados. Técnicas baseadas em aprendizado profundo não se destacou apenas na importância do balanceamento de dados, mas também forneceu ferramentas avançadas para lidar com esse problema de forma mais eficiente, veja Figura 1.

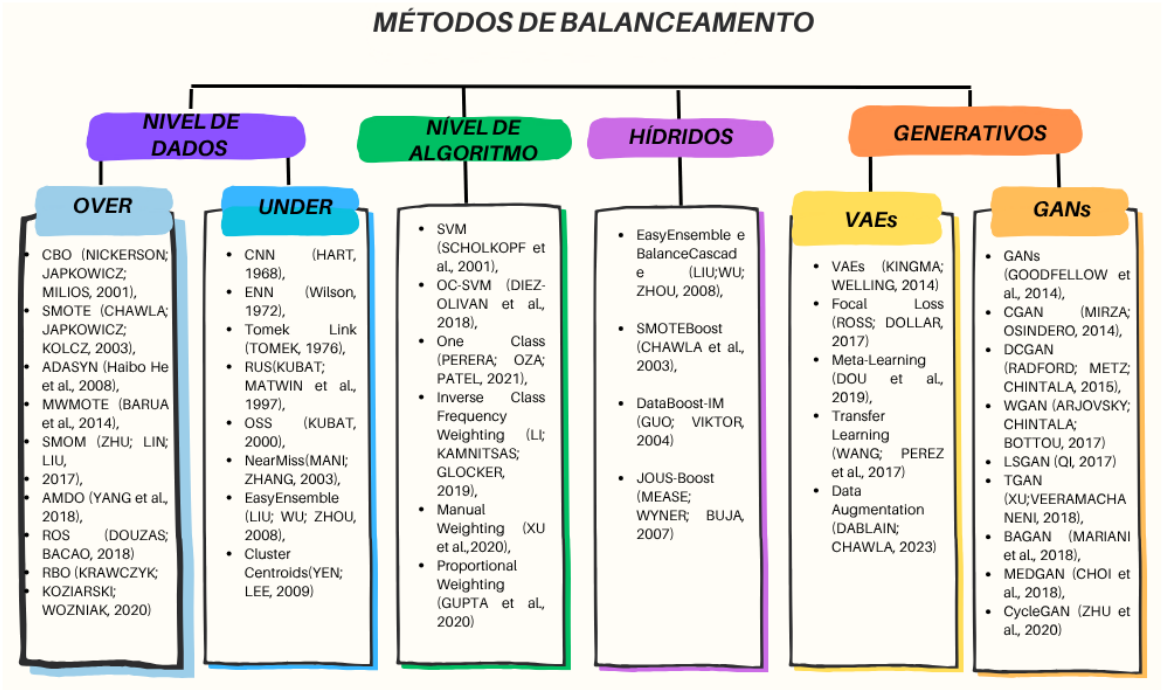
Na década atual, novas técnicas de *Loss Functions* adaptativas, como a *Focal Loss* (ROSS; DOLLÁR, 2017), foram desenvolvidas para tratar da atenção desigual em redes neurais, técnicas de *Meta-Learning* para adaptar modelos em tempo real (DOU et al., 2019), e a combinação de *Transfer Learning* com balanceamento (WANG; PEREZ et al., 2017), permitindo que modelos generalizem melhor para classes minoritárias em novos contextos (GHOSH et al., 2024). Além disso, o *Data Augmentation* ganhou destaque, gerando variações das instâncias da classe minoritária para enriquecer o treinamento (DABLAIN; CHAWLA, 2023).

Assim, o balanceamento de classes, antes considerado um problema secundário, evoluiu

para uma área de pesquisa estratégica em *machine learning*. A crescente utilização de algoritmos de aprendizado em áreas sensíveis, como saúde e detecção de fraudes, tornou evidente a necessidade de modelos imparciais e confiáveis. A história do balanceamento de classes nos mostra como a comunidade científica passou de técnicas simples de reamostragem para abordagens mais sofisticadas, como o uso de modelos generativos. Ao compreender essa evolução, podemos identificar as lacunas existentes e direcionar futuras pesquisas para o desenvolvimento de técnicas mais robustas e eficientes para lidar com a complexidade dos dados reais.

Atualmente, as abordagens de balanceamento de dados podem ser organizadas em quatro categorias principais (JOHNSON; KHOSHGOFTAAR, 2019), (JAFARIGOL; TRAFALIS, 2023),(WANG et al., 2024): métodos no *nível de dados*, que atuam diretamente no conjunto de dados para equilibrar as classes; métodos no *nível de algoritmo*, que adaptam os modelos de aprendizado de máquina para lidar com desbalanceamentos; *métodos híbridos*, que combinam técnicas de diferentes níveis; e *métodos baseados em modelos generativos*, como GANs, que geram dados sintéticos para complementar classes minoritárias. Essa divisão, apresentada na Figura 2, tem sido amplamente adotada na literatura para sistematizar e analisar as estratégias de balanceamento.

Figura 2: Visão geral sobre as categorias de algoritmos de balanceamento de dados: nível de dados, nível de algoritmo, híbridos e generativos



Fonte: Elaborada pela autora

Conforme evidenciado em (KOZIARSKI; WOZNIAK; KRAWCZYKZ, 2020), o

desbalanceamento de classes é apenas um dos desafios enfrentados em problemas de aprendizado de máquina. A presença de *outliers*, ruído nos dados e distribuições complexas pode amplificar os efeitos do desbalanceamento. Além disso, a natureza multi-classe de muitos problemas, como discutido em (ZHU; LIN; LIU, 2017) e (KRAWCZYK; KOZIARSKI; WOZNIAK, 2020), introduz novas complexidades, pois as relações entre as classes são mais difíceis de modelar.

A escolha da técnica de balanceamento mais adequada depende de diversos fatores, incluindo o tamanho do conjunto de dados, a distribuição das classes, o tipo de algoritmo de aprendizado utilizado e os objetivos da análise. Estes últimos referem-se aos resultados ou metas que se deseja alcançar com o modelo de aprendizado de máquina, como a maximização da precisão global, a redução de falsos positivos ou falsos negativos, ou a otimização de métricas específicas, como Revocação, Precisão ou *F1-score*. Por exemplo, em problemas de detecção de fraudes, o objetivo pode ser priorizar a identificação correta de casos positivos (alta sensibilidade), mesmo que isso implique um aumento de falsos positivos. De acordo com Koziarski, Wozniak e Krawczykz (2020), é fundamental entender as características específicas do problema em questão, incluindo seus objetivos, para selecionar a abordagem de balanceamento mais eficaz e alinhada às necessidades da aplicação.

Nas seções subsequentes, é apresentada uma análise detalhada de cada categoria de método de balanceamento, explorando seus fundamentos teóricos, procedimentos operacionais e aplicações práticas. Essa abordagem permitirá uma compreensão abrangente das técnicas empregadas, bem como uma avaliação crítica de suas vantagens, limitações e contextos de utilização.

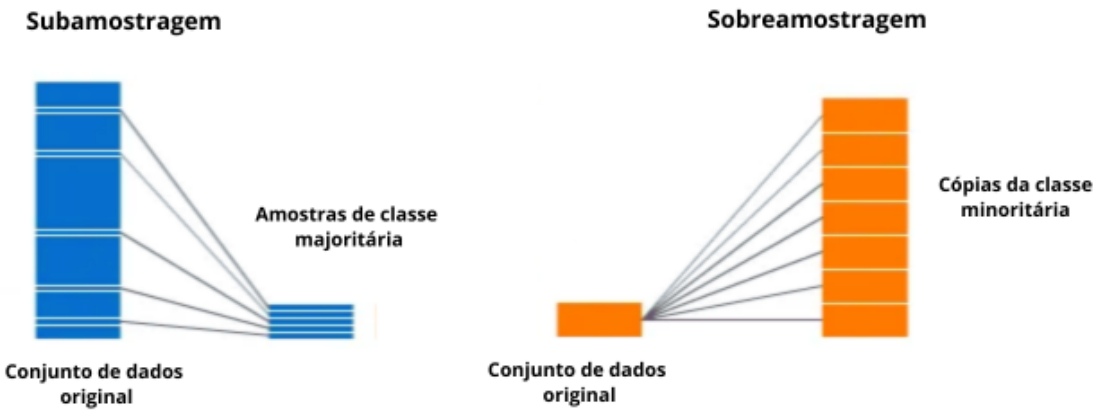
2.2 Modelos de Balanceamento em Nível de Dados

As estratégias de nível de dados lidam diretamente com a amostragem dos dados, modulando a distribuição dos dados de treinamento, com o intuito de otimizar a precisão dos modelos resultantes (NGUYEN et al., 2021).

Estes métodos compreendem a sobreamostragem da classe minoritária, a subamostragem da classe majoritária ou a aplicação de uma combinação dessas técnicas. Enquanto a subamostragem implica na remoção de elementos pertencentes à classe majoritária, a sobreamostragem visa aumentar a representatividade da classe minoritária, incorporando instâncias adicionais (NGUYEN et al., 2021), veja Figura 3.

Dada a ausência de uma garantia de que a distribuição original dos dados de treinamento seja ideal para a construção eficaz de classificadores, os métodos de amostragem desempenham um papel crucial na alteração da distribuição dos dados. Esta modificação

Figura 3: Diferenças entre as abordagens de balanceamento de dados em nível de dados com subamostragem e sobreamostragem



Fonte: Elaborada pela autora

visa viabilizar a geração de classificadores mais robustos e generalizáveis (DUBEY et al., 2014).

A subamostragem de dados, ou *undersampling*, é uma técnica que reduz o número de amostras da classe majoritária para equilibrá-la com a classe minoritária. A ideia é diminuir a desproporção entre as classes, facilitando o treinamento de modelos que são menos enviesados em favor da classe majoritária (BATISTA; PRATI; MONARD, 2004b).

Os principais algoritmos de subamostragem são: CNN (*Condensed Nearest Neighbor Rule*) (HART, 1968), ENN (*Edited Nearest Neighbor*) (Wilson, 1972), Tomek Link (TOMEK, 1976), RUS (*Random Under-Sampling*) (KUBAT; MATWIN et al., 1997), OSS (*One-Sided-Selection*) (KUBAT, 2000), NearMiss (MANI; ZHANG, 2003), EasyEnsemble (LIU; WU; ZHOU, 2008), *Cluster Centroids* (YEN; LEE, 2009) e NCL (*Neighborhood Cleaning Rule*)(LAURIKKALA, 2001).

Como a subamostragem diminui o número de amostras da classe majoritária, o conjunto de dados resultante é menor. Isso reduz o tempo de treinamento dos algoritmos de aprendizado de máquina, pois há menos dados a serem processados. Essa vantagem é especialmente relevante para grandes volumes de dados, onde o processamento pode ser um gargalo (JEDRZEJOWICZ; JEDRZEJOWICZ, 2021).

Ao remover amostras da classe majoritária, a subamostragem ajuda a criar um conjunto de dados mais balanceado. Isso melhora a capacidade do modelo em detectar padrões da classe minoritária, resultando em um melhor desempenho em termos de Precisão, Revocação e *F1-score* para a classe minoritária (NGUYEN et al., 2021).

A subamostragem, especialmente na forma aleatória, é simples de implementar e não requer parâmetros complexos. Isso a torna uma técnica prática e rápida de ser aplicada, sem a necessidade de gerar novas amostras ou realizar pré-processamentos elaborados (SUN et al., 2024).

No entanto, a maior desvantagem da subamostragem é a possibilidade de perda de informações valiosas da classe majoritária. Como as amostras são removidas aleatoriamente ou de forma seletiva, há o risco de eliminar dados que poderiam ser importantes para o modelo, especialmente se essas amostras representarem padrões ou características raras dentro da classe majoritária. Isso pode levar a um desempenho inferior do modelo para a classe majoritária (WEN et al., 2023).

Ao reduzir o número de amostras, a subamostragem também pode reduzir a variabilidade dos dados na classe majoritária. De acordo com Jedrzejowicz e Jedrzejowicz (2021), menos variabilidade pode levar o modelo a capturar menos nuances da classe majoritária, comprometendo sua capacidade de generalizar para novos dados.

Em certos casos, a subamostragem pode resultar em um subconjunto da classe majoritária que não representa adequadamente a distribuição original dessa classe. Isso pode ocorrer se a subamostragem não for feita de maneira representativa ou se algumas áreas do espaço de características forem eliminadas de forma desproporcional (WEN et al., 2023).

Quando o conjunto de dados original já é pequeno, a subamostragem pode piorar a situação, removendo muitas amostras e deixando poucos dados disponíveis para o treinamento. Isso pode resultar em um modelo fraco, com baixa capacidade de generalização, especialmente quando as amostras restantes da classe majoritária não são suficientes para representar adequadamente a classe (SUN et al., 2024).

Em alguns casos, a subamostragem extrema pode fazer com que o modelo se torne enviesado em favor da classe minoritária. Segundo Nguyen et al. (2021), embora o objetivo seja balancear o conjunto de dados, remover amostras da classe majoritária em excesso pode levar a um modelo que tem dificuldade em prever corretamente a classe majoritária.

Por outro lado, a sobreamostragem de dados, ou *oversampling*, diferentemente da subamostragem, não elimina amostras. Isso é vantajoso, pois garante que informações importantes da classe majoritária não sejam perdidas, mantendo a integridade do conjunto de dados (KRAWCZYK; KOZIARSKI; WOZNIAK, 2020), veja Figura 3.

A sobreamostragem aumenta a representatividade da classe minoritária criando novas amostras, com o objetivo de equilibrar a distribuição entre as classes e melhorar a performance dos algoritmos de aprendizado de máquina. Entre os principais algoritmos de sobreamostragem temos: AMDO (*An Over-Sampling Technique For Multi-Class Imbalanced Problems*) (YANG et al., 2018), CBO (*Cluster-based Oversampling*)

(NICKERSON; JAPKOWICZ; MILIOS, 2001), ADASYN (*Adaptive Synthetic Sampling Approach For Imbalanced Learning*) (Haibo He et al., 2008), MC-CCR (*Multi-Class Combined Cleaning and Resampling*) (KOZIARSKI; WOZNIAK; KRAWCZYKZ, 2020), MWMOTE (*Majority Weighted Minority Oversampling Technique*)(BARUA et al., 2014), ROS (*Random Oversampling*) (DOUZAS; BACAO, 2018) e RBO (*Radial Based Oversampling For Multiclass Imbalanced Data Classification*) (KRAWCZYK; KOZIARSKI; WOZNIAK, 2020), SMOM (*Synthetic Minority Oversampling Technique For Multiclasse Imbalance Problems*) (ZHU; LIN; LIU, 2017) e SMOTE (*Synthetic Minority Over-sampling TEchnique*) (CHAWLA; JAPKOWICZ; KOLCZ, 2003).

Ao aumentar o número de amostras da classe minoritária, a sobreamostragem ajuda os algoritmos de aprendizado a identificar melhor os padrões dessa classe (FAN et al., 2021). Isso leva a um desempenho mais equilibrado, reduzindo o viés em favor da classe majoritária e melhorando métricas de avaliação que são essenciais para a detecção de classes minoritárias (FERNANDEZ et al., 2018). Essas métricas incluem medidas como Revocação, que avalia a capacidade do modelo em identificar corretamente os exemplos da classe minoritária, ou a métrica *F1-score*, que combina Precisão e Revocação em uma única métrica, sendo útil quando há um desequilíbrio significativo entre as classes; e AUC-ROC (Área sob a Curva ROC), que mede a capacidade do modelo de distinguir entre as classes. Em problemas onde a classe minoritária é crítica, como em diagnósticos médicos ou detecção de fraudes, otimizar essas métricas é fundamental, pois elas refletem a eficácia do modelo em detectar casos raros ou de maior importância, mesmo que isso implique uma menor precisão global.

Um dos principais problemas associados à sobreamostragem, especialmente em métodos que replicam amostras existentes, é o risco de *overfitting*. Como o algoritmo é exposto repetidamente às mesmas instâncias ou amostras sintéticas muito semelhantes, ele pode se ajustar excessivamente a esses dados, prejudicando a sua capacidade de generalizar para novos dados de teste. Isso é comum quando se usa replicação simples ou quando os exemplos sintéticos estão muito próximos das amostras originais (FAN et al., 2021).

Em alguns cenários, a sobreamostragem pode inadvertidamente gerar amostras sintéticas da classe minoritária que se sobrepõem às regiões da classe majoritária. Isso pode confundir os algoritmos de classificação ao reduzir a clareza da fronteira entre as classes, levando a uma degradação da performance global, especialmente em casos onde a fronteira entre as classes já é tênue (NDICHU et al., 2022).

Além disso, como a sobreamostragem aumenta o tamanho do conjunto de dados, especialmente em casos com classes muito desbalanceadas, isso pode levar a um aumento no tempo de treinamento dos modelos. Schultz et al. (2024) afirmam que, dependendo do tamanho do conjunto de dados, o aumento de complexidade computacional pode ser

significativo.

2.2.1 Métodos de Sobreamostragem para Problemas de Classificação

A sobreamostragem é uma técnica que abrange diversas abordagens, incluindo a replicação de exemplos existentes, conhecida como sobreamostragem com substituições, e a geração de dados sintéticos (JOHNSON; KHOSHGOFTAAR, 2019). A sobreamostragem com substituições consiste em duplicar aleatoriamente exemplos da classe minoritária, permitindo que o modelo seja exposto a esses dados com maior frequência durante o treinamento (FERNÁNDEZ, 2018). No entanto, essa abordagem pode levar ao sobreajuste, especialmente quando os exemplos replicados são idênticos, fazendo com que o modelo memorize os dados em vez de generalizar padrões (JOHNSON; KHOSHGOFTAAR, 2019). Por isso, a criação de dados sintéticos emerge como uma estratégia particularmente vantajosa em contextos onde a replicação de exemplos não é viável quando se deseja evitar o sobreajuste. Nas próximas seções, serão apresentados os principais métodos de sobreamostragem considerados neste trabalho, destacando suas características, vantagens e limitações.

2.2.1.1 SMOTE

Chawla et al. (2002) propuseram uma abordagem que consiste em gerar casos sintéticos para a classe de interesse a partir dos casos já existentes. Os novos casos são gerados na vizinhança de cada caso da classe minoritária com o intuito de se crescer o espaço de decisão desta classe (região do R^n) e aumentar o poder de generalização dos classificadores obtidos.

De acordo com Chawla et al. (2002), a abordagem do SMOTE (*Synthetic Minority Over-sampling TEchnique*) é fundamentada em conceitos estatísticos e de interpolação, proporcionando uma maneira robusta e eficaz de lidar com conjuntos de dados desbalanceados. A formalização do Algoritmo 1 é apresentada a seguir:

Inicialmente o *require* do Algoritmo 1 define os parâmetros de entrada. X: Conjunto de dados original (*features*). y: Rótulos das classes (*labels*). k: Número de vizinhos mais próximos que serão considerados para gerar amostras sintéticas. N: Número de amostras sintéticas que serão geradas para cada exemplo da classe minoritária.

A (**linha 1**) do Algoritmo 1 apresenta um *For* que itera sobre cada exemplo x_i que pertence à classe minoritária (a classe com menos exemplos no conjunto de dados). O objetivo é gerar amostras sintéticas para equilibrar a distribuição das classes.

Para cada exemplo x_i da classe minoritária, o algoritmo encontra os k vizinhos mais próximos (usando uma métrica de distância, como a distância euclidiana). Esses vizinhos

Algoritmo 1: SMOTE

1: **Entrada:** Conjunto de dados X, rótulos das classes y, número de vizinhos k, N número de amostras sintéticas
2: **Saída:** x_i e x_j Conjunto de dados balanceado
3: **for** cada exemplo x_i da classe minoritária **do**
4: Encontrar os k vizinhos mais próximos de x_i
5: **for** $j = 1$ até N **do**
6: Gerar um número aleatório ρ entre 0 e 1
7: Calcular $E = x_i + \rho \times (x_j - x_i)$
8: Adicionar s ao conjunto de dados
9: **end for**
10: **end for**
11: **return** conjunto de dados balanceado

também devem pertencer à classe minoritária (**linha 2**).

Para cada exemplo x_i, o algoritmo gera N amostras sintéticas (**linha 4**). O valor de N é definido pelo usuário e controla quantas novas amostras serão criadas para equilibrar as classes.

É gerado um número aleatório ρ no intervalo [0,1] (**linha 5**). Esse número será usado para criar uma nova amostra sintética ao longo da linha que conecta x_i a um de seus vizinhos (CHAWLA et al., 2002).

Depois é criada uma nova amostra sintética usando interpolação linear entre x_i e um de seus vizinhos x_j; onde x_i é exemplo original da classe minoritária, x_j é um dos k vizinhos mais próximos de x_i, ρ é um número aleatório que controla a posição da nova amostra ao longo da linha entre x_i e x_j. A Equação $E = x_i + \rho \times (x_j - x_i)$ garante que a nova amostra E esteja no espaço de características entre x_i e x_j, ou seja, ao longo da linha que conecta essas duas amostras (**linha 5**). Embora o SMOTE busque evitar a invasão do espaço de decisão da classe negativa, há situações em que a geração de amostras sintéticas pode ultrapassar esse limite, especialmente quando as classes estão muito próximas ou sobrepostas no espaço de características.

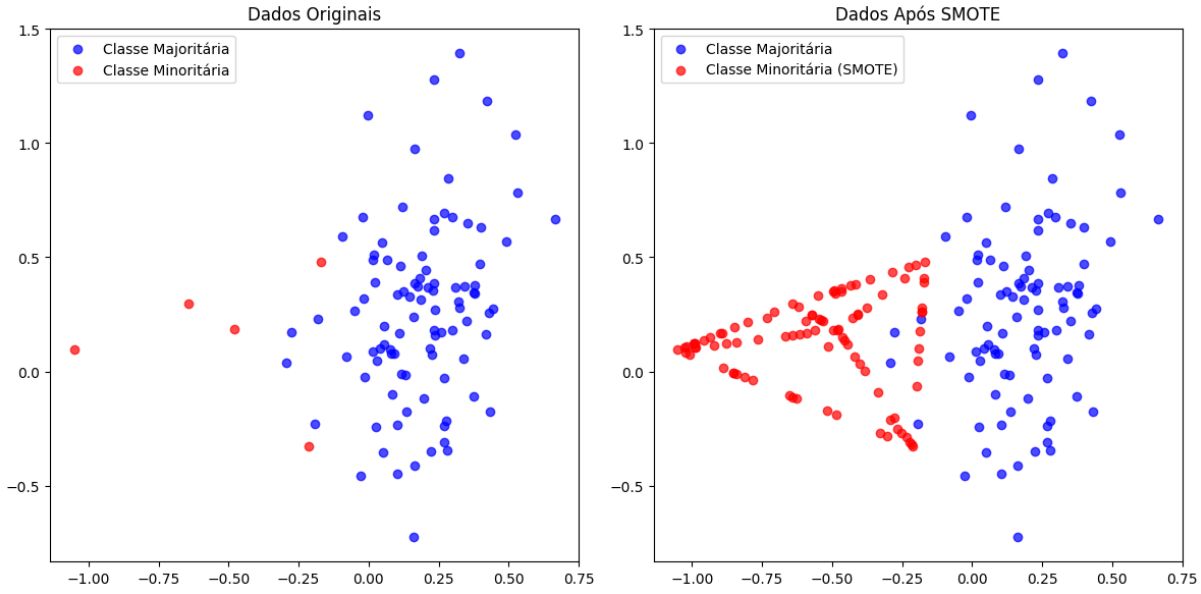
A amostra sintética E é adicionada ao conjunto de dados original, (**linha 6**). Isso aumenta o número de exemplos da classe minoritária, ajudando a equilibrar a distribuição das classes.

Por fim, o algoritmo retorna o conjunto de dados original com as novas amostras sintéticas adicionadas. Agora, o conjunto de dados está mais balanceado, com mais exemplos da classe minoritária (**linha 9**) (CHAWLA et al., 2002).

Desta forma, o método SMOTE, essencialmente, cria novos exemplos ao longo de linhas que conectam os exemplos existentes da classe minoritária e seus vizinhos mais próximos.

Isso permite que a técnica capture a distribuição local dos dados e gere exemplos sintéticos que são semelhantes aos exemplos originais, mas não necessariamente idênticos. A representação gráfica do algoritmo SMOTE é apresentada na Figura 4.

Figura 4: Visualização do processo de sobreamostragem sintética (SMOTE)



Fonte: Elaborada pela autora

O SMOTE pode gerar casos positivos que invadem o espaço de decisão da classe negativa. Essa característica, denominada sobreposição de classes, tende a degenerar o desempenho de classificadores obtidos a partir de tais dados (BATISTA; PRATI; MONARD, 2004a). A sobreposição de classes também pode ser uma propriedade natural dos dados que é aguçada com a utilização do SMOTE. Esse problema do relacionamento entre desbalanceamento e sobreposição de classes tem recebido atenção de autores que utilizam os métodos Tomek links (TOMEK, 1976) e ENN (Wilson, 1972) para a limpeza de dados após a aplicação do SMOTE. .

2.2.2 Modelos de Subamostragem para Problemas de Classificação

A subamostragem elimina exemplos da classe majoritária. Os exemplos que serão suprimidos podem ser selecionados aleatoriamente (subamostragem aleatória) ou com base em alguma informação a priori (subamostragem informativa) (SUN et al., 2024).

No caso de subamostragem aleatória, a principal dificuldade é a possibilidade de perda de informação causada pela eliminação de exemplos representativos da classe majoritária. A subamostragem informativa tenta solucionar esse problema eliminando uma fração menos representativa como: exemplos redundantes, ruidosos e/ou próximos à fronteira de separação entre as classes (*borderlines*) (SUN et al., 2024). Contudo, a escolha de

critérios adequados para selecionar esses exemplos não é uma tarefa trivial. A maioria dos métodos informativos usa o algoritmo KNN (*K-Nearest Neighbour*) para guiar o processo de subamostragem. Nas próximas seções, os principais métodos de subamostragem são apresentados.

2.2.2.1 RUS

O RUS (*Random Under-Sampling*) é uma técnica clássica de balanceamento de conjuntos de dados desequilibrados. Apesar de sua autoria não ser atribuída a um único pesquisador específico, o RUS emergiu como uma abordagem intuitiva e simples no contexto de métodos de pré-processamento de dados (KUBAT; MATWIN et al., 1997).

Na subamostragem aleatória a classe majoritária é subamostrada pela remoção aleatória de amostras da classe majoritária até que a classe minoritária torne-se alguma porcentagem especificada da classe majoritária (KUBAT; MATWIN et al., 1997). A apresentação formal do RUS é apresentada no Algoritmo 2.

Algoritmo 2: Random Under-Sampling - RUS

1: **Entrada**: Matriz de features X, Vetor de rótulos y, Razão desejada entre as classes *ratio* (opcional)
2: **Saída:** Conjuntos de dados balanceados X, y
3: Identificar a classe majoritária C_{maj} e a classe minoritária C_{min}
4: Calcular o número de exemplos a serem removidos da classe majoritária
5: **while** número de exemplos em $C_{maj} >$ número desejado **do**
6: Selecionar aleatoriamente um exemplo de C_{maj}
7: Remover o exemplo selecionado de X e y
8: **end while**
9: **return** X, y

Os parâmetros de entrada do algoritmo são: X é a matriz de *features* (atributos) do conjunto de dados, y é o vetor de rótulos (*labels*) correspondentes às classes, e *ratio* é a razão desejada entre o número de exemplos da classe majoritária e minoritária (opcional). Se não for especificado, o algoritmo busca equilibrar as classes. X_{res}, y_{res} define o resultado esperado, onde X_{res} é a matriz de *features* após o balanceamento e y_{res} é o vetor de rótulos após o balanceamento, Algoritmo 2.

Na linha 1 do Algoritmo 2, as classes majoritárias C_{maj} e minoritária C_{min} são identificadas (**linha 1**). Em seguida, é calculado quantos exemplos da classe majoritária precisam ser removidos para atingir o balanceamento desejado (**linha 2**). Se *ratio* for especificado, o cálculo leva em conta a razão desejada entre as classes. Caso contrário, o objetivo é igualar o número de exemplos da classe majoritária ao da classe minoritária.

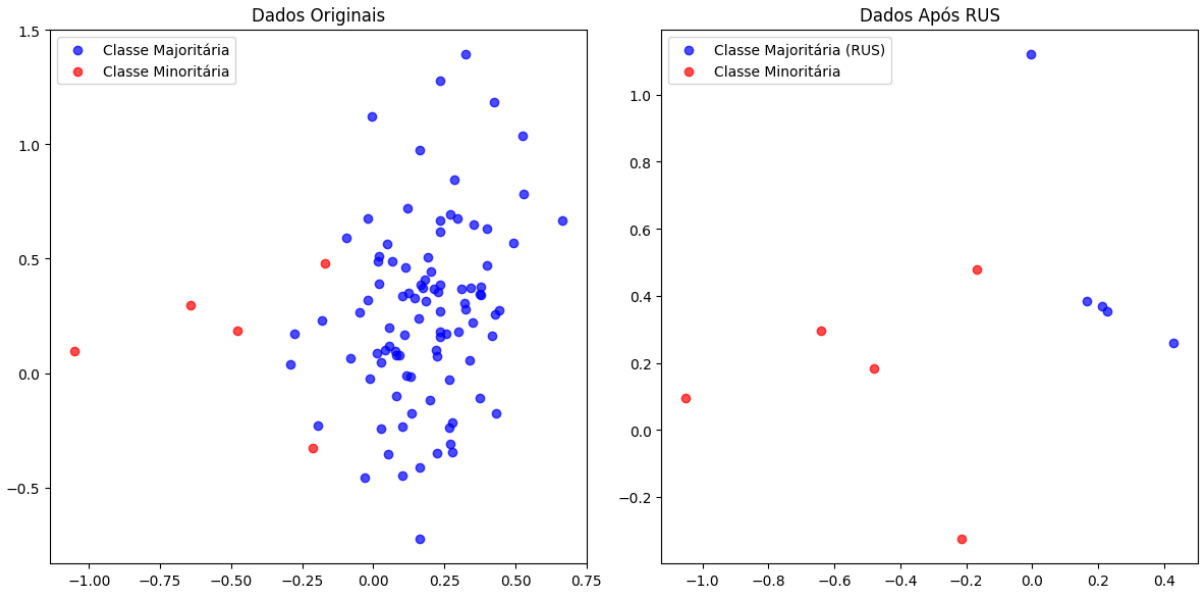
Após isto, inicia-se um *loop* que continua até que o número de exemplos da classe majoritária seja reduzido ao valor desejado (calculado na linha anterior) (**linha 3**). Depois

é selecionado aleatoriamente um exemplo da classe majoritária C_{maj} para ser removido. A seleção é feita de forma aleatória para evitar viés na remoção de exemplos (**linha 4**). O exemplo selecionado de X e y é removido (linha 5). O *loop* é finalizado quando o número de exemplos da classe majoritária atinge o valor desejado (**linha 6**). O conjunto de dados balanceado é retornado X, y (**linha 7**).

O ponto principal da estratégia reside na busca por uma reamostragem equilibrada, onde todas as classes possuam uma representação iqualitária. Este aspecto é crucial para mitigar quaisquer viéses inerentes ao modelo, direcionando-o de maneira mais imparcial e equitativa na tarefa de classificação.

O resultado almejado é a melhoria da capacidade de generalização do modelo, especialmente em relação às classes minoritárias, que, de outra forma, poderiam ser sub-representadas (KUBAT; MATWIN et al., 1997). A Figura 5 mostra a distribuição dos dados antes e após a aplicação do algoritmo RUS.

Figura 5: Representação gráfica do uso do algoritmo *Random Under-Sampling* - RUS para balanceamento de dados



Fonte: Elaborada pela autora

O RUS apresenta vantagens, como sua simplicidade, uma vez que é de fácil implementação e compreensão, e sua eficiência, já que reduz o tamanho do conjunto de dados, diminuindo o tempo de treinamento dos modelos. No entanto, o método também apresenta desvantagens. A principal delas é a perda de informações relevantes associadas à classe majoritária, decorrente da redução do número de exemplos disponíveis para treinamento. Além disso, a seleção aleatória de exemplos pode introduzir viés no conjunto de dados, especialmente se os exemplos removidos forem representativos da distribuição original da

classe majoritária, comprometendo a generalização do modelo (GARCÍA; HERRERA, 2009).

Em casos de extremo desequilíbrio do conjunto de dados, onde a classe majoritária pode ser centenas ou milhares de vezes maior que a classe minoritária, o *undersampling* pode não ser uma opção viável. Isso ocorre porque a remoção excessiva de exemplos da classe majoritária pode levar à sub-representação dessa classe, resultando em perda de informações valiosas e prejudicando a capacidade do modelo de generalizar para novos dados. Por exemplo, em um conjunto de dados com 100.000 exemplos, onde apenas 1.000 pertencem à classe minoritária, reduzir a classe majoritária para 1.000 exemplos significaria descartar 99% dos dados, o que pode ser inviável em muitos cenários.

Assim como na sobreamostragem, a realização de experimentos e a adaptação dos parâmetros de *undersampling* são fundamentais para identificar a abordagem mais adequada ao conjunto de dados e ao problema em análise. Entre os parâmetros que podem ser ajustados estão:

- Proporção de balanceamento (*ratio*): define a razão entre o número de exemplos da classe majoritária e minoritária após o *undersampling*. Por exemplo, um ratio de 1:1 equilibra as classes, enquanto um ratio de 2:1 mantém o dobro de exemplos na classe majoritária.

- Método de seleção de exemplos: pode ser aleatório (como no *Random Undersampling*) ou baseado em técnicas mais sofisticadas, como a remoção de exemplos redundantes ou próximos à fronteira de decisão.

- Tamanho final do conjunto de dados: define quantos exemplos devem ser mantidos no conjunto de dados após o *undersampling*, considerando o *trade-off* entre balanceamento e retenção de informações.

A escolha desses parâmetros deve ser guiada por experimentos e validações, como a avaliação da Precisão, Revocação, *F1-score* ou outras métricas relevantes para o problema específico. Em casos de extremo desequilíbrio, onde uma classe é significativamente mais frequente que a outra (por exemplo, proporções de 1:100 ou mais), técnicas convencionais de balanceamento podem não ser suficientes para garantir um bom desempenho do modelo. Nessas situações, técnicas híbridas, que combinam subamostragem (redução da classe majoritária) e sobreamostragem (aumento da classe minoritária), ou métodos avançados, como *NearMiss* ou *Tomek Links*, podem ser mais adequados. Essas abordagens buscam preservar a representatividade das classes, evitando a perda de informações importantes da classe majoritária ou a geração de ruído excessivo na classe minoritária, melhorando assim a capacidade do modelo de generalizar e prever corretamente ambas as classes (SUN et al., 2024).

2.3 Modelos de Balanceamento em Nível de Algoritmos

As técnicas de balanceamento no nível de algoritmo são métodos projetados para ajustar o funcionamento dos algoritmos de aprendizado de máquina de modo que eles possam lidar com conjuntos de dados desbalanceados sem a necessidade de modificar diretamente os dados. Isso é feito principalmente ajustando a função de perda, atribuindo diferentes penalidades ou custos para os erros cometidos nas classes minoritárias e majoritárias (ZHANG; ZHANG, 2017).

Entre os algoritmos que utilizam essa abordagem temos o SVM (Support Vector Machine) (SCHöLKOPF et al., 2001), OC-SVM (*One Class Support Vector Machine*) (DIEZ-OLIVAN et al., 2018) e OCC (*One Class Classification*) (PERERA; OZA; PATEL, 2021), dentre outros.

Algoritmos sensíveis ao custo ajustam a função de perda do modelo para aumentar o custo de erros na classificação da classe minoritária, fazendo com que o modelo dê mais atenção a essas classes. Essas abordagens permitem lidar com o desequilíbrio de classes sem modificar o conjunto de dados original, preservando sua integridade e reduzindo o risco de *overfitting* (DING et al., 2023). Os modelos são customizados atribuindo pesos maiores às classes minoritárias, penalizando mais os erros e melhorando o desempenho nessas classes (ZHANG; ZHANG, 2017).

No entanto, esses métodos exigem a definição de matrizes de custos, o que pode ser complexo e difícil de transferir entre domínios diferentes. Além disso, definir corretamente os pesos pode ser desafiador, especialmente em dados de alta dimensionalidade, e ajustes excessivos podem enviesar o modelo, prejudicando a classe majoritária (KHAN; CHAUDHARI; CHANDRA, 2024).

Nestes algoritmos a função de perda quantifica a discrepância entre as predições do modelo e os valores reais. Em problemas de classificação, a perda é geralmente calculada como a soma de perdas individuais para cada exemplo. Ao atribuir pesos diferentes às classes, a contribuição de cada exemplo para a perda total é ajustada (DING et al., 2023).

Os pesos de classe são multiplicadores que aumentam a importância de exemplos da classe minoritária na função de perda. Matematicamente, a perda total ponderada pode ser expressa pela Equação 2.1:

$$L = \sum_{i=1}^{N} w_{c(i)} \cdot \ell(y_i, \hat{y}_i), \tag{2.1}$$

onde: N: número total de exemplos no conjunto de dados
$w_{c(i)}$: peso associado à classe $c(i)$ do exemplo i
$l(y_i, \hat{y}_i)$: função de perda individual para o exemplo i

y_i: rótulo verdadeiro do exemplo i
$\hat{y}_i$: rótulo previsto pelo modelo para o exemplo i

Ao atribuir pesos maiores aos exemplos da classe minoritária, o modelo é incentivado a aprender melhor suas características, reduzindo o viés em direção à classe majoritária e melhorando o desempenho geral. Essa abordagem é especialmente útil em cenários onde o desequilíbrio de classes pode prejudicar a capacidade de generalização do modelo.

A escolha dos pesos de classe é um desafio. Uma abordagem comum é atribuir pesos inversamente proporcionais à frequência das classes no conjunto de dados. No entanto, outras estratégias, como a análise de custo-benefício, podem ser mais adequadas em determinados contextos.

Os algoritmos de balanceamento com ponderação de classe representam uma vertente alternativa no tratamento de conjuntos de dados desbalanceados. Em vez de intervir diretamente nos dados originais, esses algoritmos ajustam os parâmetros do processo de aprendizado do modelo, a fim de acomodar o desequilíbrio presente entre as classes (GUPTA et al., 2020). Tal ajuste é realizado por meio da atribuição de pesos distintos às diferentes classes durante a fase de treinamento do modelo, conferindo maior relevância às classes minoritárias (XU et al., 2020). A operacionalização desses algoritmos pode variar, a depender do algoritmo de aprendizado de máquina empregado. Contudo, o princípio fundamental reside na elevação dos pesos associados às classes minoritárias, em detrimento daqueles atribuídos às classes majoritárias (XU et al., 2020). Este ajuste pode ser executado por meio de diversos procedimentos:

1. A Ponderação Inversa da Frequência da Classe (*Inverse Class Frequency Weighting*) consiste em uma estratégia na qual os pesos atribuídos às classes são determinados de maneira inversamente proporcional à frequência das mesmas no conjunto de dados. Desse modo, as classes menos frequentes são agraciadas com pesos mais substanciais, ao passo que aquelas mais frequentes são dotadas de pesos relativamente menores. Não se vinculam algoritmos específicos a essa técnica; em vez disso, sua aplicabilidade se estende a uma variedade de algoritmos de aprendizado de máquina que possuem suporte para a ponderação de classe, como a regressão logística, árvores de decisão, SVM (*Support Vector Machine*), dentre outros (LI; KAMNITSAS; GLOCKER, 2019).

2. A Ponderação Manual (*Manual Weighting*) se refere à prática na qual conhecimento especializado sobre o domínio do problema é empregado para a atribuição manual de pesos às classes. Por exemplo, quando se considera que uma classe minoritária possui uma relevância superior em relação à classe majoritária, pode-se optar por atribuir um peso mais elevado àquela. Nessa abordagem, os pesos das classes são

ajustados de maneira manual para cada algoritmo de aprendizado de máquina, em conformidade com as necessidades específicas do contexto em análise (XU et al., 2020).

3. Na abordagem da Ponderação Proporcional (*Proportional Weighting*), os pesos atribuídos às classes são determinados proporcionalmente à relação entre o número de instâncias pertencentes à classe majoritária e à classe minoritária. Esta metodologia assegura que o modelo conceda maior relevância às classes minoritárias, sem considerar a discrepância absoluta na quantidade de instâncias entre tais classes (GUPTA et al., 2020).

4. Algoritmos de Aprendizado com Suporte para Ponderação de Classe Incorporada (*Learning Algorithms with Built-in Class Weighting Support*) constituem uma categoria de algoritmos de aprendizado de máquina que integram nativamente a funcionalidade de ponderação de classe. Estes algoritmos permitem a especificação dos pesos associados às classes como parâmetros de entrada durante o processo de treinamento do modelo. Tal incorporação simplifica significativamente o procedimento de aplicação de ponderação de classe, eliminando a necessidade de intervenção manual na definição dos pesos (XU et al., 2020). Por exemplo o OC-SVM é uma variação do SVM projetada para detecção de *outliers* e problemas de classificação de uma única classe (detecção de anomalias). Ele cria uma fronteira que envolve a maior parte dos dados de treinamento da classe positiva, separando-os do restante do espaço de características. Isso permite que o modelo identifique novos dados que não se encaixam no padrão estabelecido, tratando-os como anomalias ou pertencentes a outra classe (DIEZ-OLIVAN et al., 2018).

   Dentro dessa categoria de algoritmos, diversas bibliotecas oferecem suporte ao tratamento de classes desbalanceadas por meio de técnicas de ponderação de classes, tais como:

   - `Scikit-learn`: apresenta uma ampla gama de algoritmos de classificação que suportam ponderação de classe diretamente em seus parâmetros. A classe `class_weight` possibilita a especificação dos pesos individuais para cada classe durante a fase de treinamento do modelo (KRAMER; KRAMER, 2016).

   - *XGBoost*: destaca-se como uma implementação otimizada de algoritmo de `gradient boosting` que incorpora suporte nativo à ponderação de classe. O parâmetro `scale_pos_weight` oferece a flexibilidade de ajustar os pesos das classes em consonância com sua frequência relativa no conjunto de dados (VELARDE et al., 2024).

   - *LightGBM*: figura como uma biblioteca de `gradient boosting` que contempla

a ponderação de classe. O parâmetro `is_unbalance` possibilita o tratamento automatizado de classes desbalanceadas, atribuindo pesos às classes de maneira inversamente proporcional à sua frequência (KE et al., 2017).

- *CatBoost*: outra biblioteca de `gradient boosting`, também incorpora suporte para classes desbalanceadas. Através do parâmetro `class_weights`, é viável especificar os pesos individuais para cada classe durante o procedimento de treinamento do modelo (PROKHORENKOVA et al., 2018).

2.4 Modelos de Balanceamento Híbridos

É a combinação dos métodos de nível de dados e de nível de algoritmo aplicados a problemas de desequilíbrio de classe (KRAWCZYK, 2016). Utilizam amostragem para reduzir o ruído e o desequilíbrio, seguida de aprendizado ou ajuste de limites sensíveis a custos para diminuir ou viés em favor da classe majoritária (JOHNSON; KHOSHGOFTAAR, 2019).

Houve um desenvolvimento de métodos híbridos de balanceamento que combinaram diferentes abordagens para maximizar a eficácia do aprendizado. Esses métodos integram técnicas tradicionais de amostragem (como *oversampling* e *undersampling*) com algoritmos de aprendizado sensíveis a custos e estratégias de conjunto para criar soluções robustas (DABLAIN; KRAWCZYK; CHAWLA, 2021). Foram exploradas em várias pesquisas como o *EasyEnsemble* e *BalanceCascade* (LIU; WU; ZHOU, 2008), que treinaram múltiplos classificadores combinando subconjuntos da classe majoritária com a minoritária, formando conjuntos de treinamento balanceados. São exemplos de métodos desta abordagem: *SMOTEBoost* (CHAWLA et al., 2003), *DataBoost-IM* (GUO; VIKTOR, 2004) e *JOUS-Boost* (MEASE; WYNER; BUJA, 2007).

2.5 Modelos de Balanceamento Generativos

Os modelos generativos são algoritmos que aprendem a modelar a distribuição de probabilidade dos dados de entrada. O objetivo é capturar as características dos dados originais para criar novas amostras que sejam semelhantes aos dados reais (DING et al., 2023).

Entre os modelos generativos, destacam-se os *Autoencoders* Variacionais (VAEs) (BANK; KOENIGSTEIN; GIRYES, 2023) e as Redes Adversariais Gerativas (GANs) (GOODFELLOW, 2017). Os VAEs aprendem uma representação latente dos dados e gera amostras com base nessa distribuição, mantendo as características principais dos dados originais. As GANs são um tipo específico de modelo generativo que inclui um componente adversário em seu treinamento (GOODFELLOW et al., 2014).

A utilização de modelos generativos e adversários tem se expandido para além de imagens, incluindo dados tabulares e textuais para capturar as complexidades deste tipo de dado (JEONG; JEONG; KIM, 2023), gerando amostras sintéticas que mantêm a integridade tanto de variáveis numéricas quanto categóricas.

2.5.1 DeepSMOTE

O DeepSMOTE é composto por uma arquitetura codificador/decodificador, uma abordagem de sobreamostragem fundamentada no SMOTE, e uma função de perda que incorpora tanto uma componente de reconstrução quanto um termo de penalização, como discutido por (DABLAIN; KRAWCZYK; CHAWLA, 2021).

O DeepSMOTE não depende da presença de um discriminador durante o ciclo de geração das instâncias artificiais. Em vez disso, uma função penalizadora é empregada, assegurando a utilização eficaz dos dados de treinamento para aprimorar o gerador.

Tanto o codificador quanto o decodificador são submetidos a um treinamento. No decorrer do treinamento do DeepSMOTE, lotes de dados desbalanceados são processados pelo codificador/decodificador. Uma função de perda é então computada sobre esses lotes de dados. Todas as classes são empregadas durante esse procedimento, visando permitir ao codificador/decodificador aprender tanto as classes majoritárias quanto as minoritárias (DABLAIN; KRAWCZYK; CHAWLA, 2021). A visão geral do pseudocódigo do DeepSMOTE é apresentado no Algoritmo 3.

Algoritmo 3: DeepSMOTE: Balanceamento de Dados com SMOTE e Autoencoders

Entrada: Conjunto de dados $X = \{x_1, x_2, \ldots, x_n\}$, onde $x_i \in \mathbb{R}^d$, e rótulos $Y = \{y_1, y_2, \ldots, y_n\}$, onde $y_i \in \{0, 1\}$.
Saída: Conjunto de dados balanceado X_{balanced}.
Passo 1: Separar as classes
$X_{\text{minoritaria}} \leftarrow \{x_i \in X \mid y_i = 1\}$
$X_{\text{majoritaria}} \leftarrow \{x_i \in X \mid y_i = 0\}$
Passo 2: Treinar um Autoencoder
Construir um Autoencoder com encoder E e decoder D.
Treinar o Autoencoder usando $X_{\text{minoritaria}}$ para aprender uma representação latente.
Passo 3: Projetar dados no espaço latente
$Z_{\text{minoritaria}} \leftarrow E(X_{\text{minoritaria}})$
Passo 4: Aplicar SMOTE no espaço latente
Gerar amostras sintéticas $Z_{\text{sinteticas}}$ aplicando SMOTE em $Z_{\text{minoritaria}}$.
Passo 5: Reconstruir amostras sintéticas no espaço original
$X_{\text{sinteticas}} \leftarrow D(Z_{\text{sinteticas}})$
Passo 6: Combinar os dados
$X_{\text{balanced}} \leftarrow X_{\text{majoritaria}} \cup X_{\text{minoritaria}} \cup X_{\text{sinteticas}}$
return X_{balanced}

O primeiro passo do algoritmo é separar as classes (**linhas 2-3**). O conjunto de dados é dividido em duas partes: a classe minoritária ($X_{\text{minoritaria}}$) e a classe majoritária ($X_{\text{majoritaria}}$).

Depois *Autoencoder* é treinado usando apenas os dados da classe minoritária. O *Autoencoder* consiste em um *encoder* (E) que mapeia os dados para um espaço latente de dimensão reduzida e um *decoder* (D) que reconstrói os dados a partir do espaço latente (**linhas 4-5**). O objetivo é aprender uma representação latente que capture as características essenciais dos dados da classe minoritária.

Os dados da classe minoritária são projetados no espaço latente usando o encoder (E). Isso resulta em $Z_{\text{minoritaria}}$, que é a representação latente dos dados originais (**linha 6**).

Em seguida, o SMOTE é aplicado no espaço latente para gerar amostras sintéticas (**linhas 9-10**). O SMOTE funciona selecionando pares de amostras no espaço latente e gerando novas amostras por interpolação linear.

As amostras sintéticas geradas no espaço latente são reconstruídas no espaço original usando o *decoder* (D). Isso resulta em $X_{\text{sinteticas}}$, que são amostras sintéticas da classe minoritária no espaço original (**linhas 11-12**).

Logo depois, o conjunto de dados balanceado é criado combinando a classe majoritária, a classe minoritária original e as amostras sintéticas geradas. O resultado é X_{balanced}, que agora tem um número equilibrado de amostras para ambas as classes. Após isso, o algoritmo retorna o conjunto de dados balanceado (**linhas 14-15**).

A geração de amostras sintéticas com SMOTE no espaço latente é definida de forma explicativa através do Algoritmo 4 .

Algoritmo 4: Geração de Amostra Sintética com SMOTE no Espaço Latente

1: **Entrada:** Dados da classe minoritária no espaço latente (X), Objeto Nearest Neighbors (nn)
2: **Saída: Amostra sintética (*synthetic_sample*)**
3: Ajuste o modelo Nearest Neighbors aos dados X
4: nn.fit(X)
5: Para cada amostra base X_{base} em X:
6: Encontre os k vizinhos mais próximos de X_{base} em X
7: $dist, ind \leftarrow nn$.kneighbors(X_{base})
8: Selecione aleatoriamente um vizinho $X_{neighbor}$ entre os k vizinhos
9: Gere uma nova amostra sintética por interpolação linear
10: $synthetic\_sample \leftarrow X_{base} + \text{np.random.rand}() \times (X_{neighbor} - X_{base})$
11: **return** ($synthetic\_sample$)

Define as entradas (dados da classe minoritária no espaço latente X e o objeto Nearest Neighbors nn) e a saída amostra sintética ($synthetic\_sample$) (**linhas 1-2**).

Ajusta o modelo nn aos dados para permitir consultas de vizinhança (**linha 3**), realiza o treinamento do modelo nn.fit(X) (**linha 4**). Itera para cada amostra base X_{base} em X (**linha 5**). Para cada X_{base}, encontra os k vizinhos mais próximos usando $nn.kneighbors$ (**linha 6)**.

Seleciona aleatoriamente um vizinho $X_{neighbor}$ entre os k vizinhos (**linha 7)**. Interpola linearmente entre X_{base} e ($X_{neighbor}$) O termo $np.random.rand()$ gera um fator aleatório entre 0 e 1 para a interpolação (**linha 8)**.

$$(X_{neighbor} synthetic\_sample \leftarrow X_{base} + \text{np.random.rand}() \times (X_{neighbor} - X_{base})$$

Retorna a amostra sintética gerada (**linha 9)**.

Os passos do treinamento do DeepSMOTE podem ser explicados de forma mais detalhada através do algoritmo 5

Algoritmo 5: Passos de Treinamento do DeepSMOTE

1: **Entrada:** Batch de imagens X, função de perda $\mathcal{L}_{\text{MSE}}$
2: **Parâmetros:** Encoder E, Decoder D, taxa de aprendizado α
3: **Passo de Treinamento:**
4: $z \leftarrow E(X)$ {Codifica para espaço latente}
5: $\hat{X} \leftarrow D(z)$ {Decodifica para reconstrução}
6: $\mathcal{L} \leftarrow \mathcal{L}_{\text{MSE}}(\hat{X}, X)$ {Calcula erro}
7: $\nabla_E, \nabla_D \leftarrow \text{Backprop}(\mathcal{L})$ {Gradientes}
8: $\theta_E \leftarrow \theta_E - \alpha \nabla_E$ {Atualiza encoder}
9: $\theta_D \leftarrow \theta_D - \alpha \nabla_D$ {Atualiza decoder}

Define as entradas Batch de imagens X (dados de treinamento), função de perda $\mathcal{L}_{\text{MSE}}$ (erro quadrático médio) (**linha 1)**. Os parâmetros são especificados Encoder E, *decoder* D (redes neurais do autoencoder), taxa de aprendizado α (controla a magnitude das atualizações) (**linha 2)**.

O encoder E mapeia as imagens X para o espaço latente $z \leftarrow E(X)$, para extrair características compactas e representativas dos dados (**linha 3)**.

O *decoder* D reconstrói as imagens a partir do espaço latente $\hat{X} \leftarrow D(z)$, para garantir que o autoencoder aprenda uma representação fiel dos dados originais (**linha 4)**.

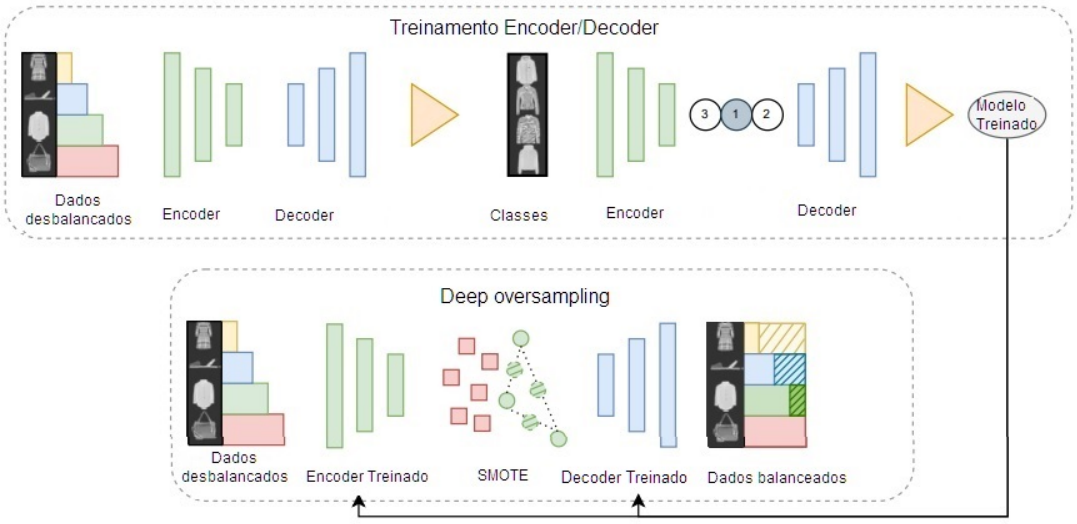
Calcula o erro de reconstrução usando MSE (Mean Squared Error) $\mathcal{L} \leftarrow \mathcal{L}_{\text{MSE}}(\hat{X}, X)$ (**linha 5)**.

Calcula os gradientes em relação aos parâmetros do *encoder* e *decoder* $\nabla_E, \nabla_D \leftarrow$ Backprop($\mathcal{L}$), através de derivação automática (autograd) para otimização (**linha 6)**.

Atualiza os parâmetros θ_E (*encoder*) e θ_D (*decoder*) via gradiente descendente , para minimizar $\mathcal{L}$ ajustando os pesos das redes (**linha 7-8)**.

A Figura 6 apresenta a arquitetura do DeepSMOTE (DABLAIN; KRAWCZYK; CHAWLA, 2021). Durante o treinamento, os dados são amostrados e codificados. E a ordem dos exemplos é permutada antes da decodificação. O codificador e o decodificador treinados são então combinados com SMOTE para produzir dados sobreamostrados.

Figura 6: Arquitetura DeepSMOTE



Fonte: Elaborado pela autora, adaptado de (DABLAIN; KRAWCZYK; CHAWLA, 2021)

Conforme delineado por Dablain, Krawczyk e Chawla (2021), para que um método de sobreamostragem seja bem-sucedido em contextos de aprendizado profundo, é imprescindível que ele satisfaça três critérios fundamentais:

1. Deve operar de maneira holística, admitindo entradas brutas, tais como imagens, e ser compatível com a técnica SMOTE para gerar dados sobre-amostrados.

2. Requer uma representação dos dados primários, incorporando-os a um espaço de características de dimensionalidade inferior, propício à sobreamostragem.

3. Necessita gerar de forma eficiente uma saída (por exemplo, imagens) que seja prontamente inspecionável visualmente, sem a necessidade de extensa manipulação adicional.

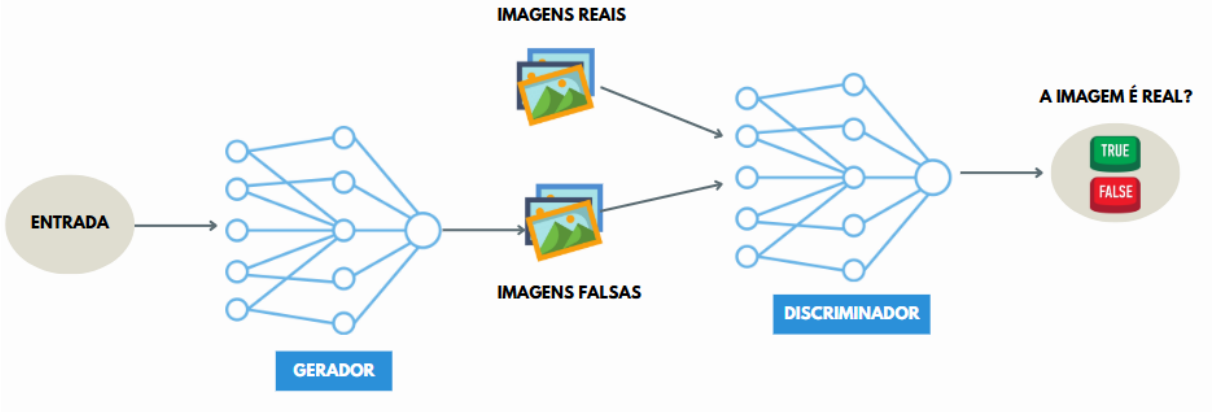
2.5.2 Redes Generativas Adversárias (GANs)

As Redes Generativas Adversárias (GANs), introduzidas por Goodfellow et al. (2014), representam uma abordagem inovadora no campo de modelos generativos. Essas redes

são compostas por dois componentes principais: um gerador (G) e um discriminador (D), que atuam de forma adversária durante o treinamento. O gerador tem como objetivo criar dados sintéticos que se assemelhem aos dados reais, enquanto o discriminador tenta distinguir entre dados reais e falsos. Esse processo é inspirado na teoria dos jogos, onde o equilíbrio de Nash é alcançado quando o gerador produz dados indistinguíveis dos reais, e o discriminador não consegue diferenciá-los com precisão.

A arquitetura das GANs é ilustrada na Figura 7, que mostra a interação entre o gerador e o discriminador. O gerador recebe um vetor de ruído aleatório z como entrada e o transforma em uma amostra sintética $G(z)$. Por outro lado, o discriminador recebe tanto amostras reais quanto as geradas por G e tenta classificá-las corretamente. O treinamento ocorre de forma simultânea, com o gerador buscando enganar o discriminador e o discriminador tentando melhorar sua capacidade de distinção.

Figura 7: Arquitetura das Redes Generativas Adversárias (GANs)



Fonte: Elaborada pela autora, adaptado de (GOODFELLOW et al., 2014)

2.5.2.1 Gerador

O gerador G é responsável por mapear um vetor de ruído z, geralmente amostrado de uma distribuição uniforme ou normal, para o espaço de dados desejado. A função $G(z; \theta^{(G)})$ é tipicamente implementada como uma rede neural profunda, onde $\theta^{(G)}$ representa os parâmetros do modelo. Durante o treinamento, o gerador aprende a gerar amostras que se aproximam da distribuição real dos dados, $p_{dados}(x)$. O sinal de treinamento para o gerador é fornecido pelo discriminador, que avalia a qualidade das amostras geradas.

2.5.2.2 Discriminador

O discriminador D atua como um classificador binário, recebendo tanto amostras reais quanto as geradas pelo gerador. Sua função é estimar a probabilidade de uma amostra ser

real. Durante o treinamento, o discriminador é ajustado para maximizar sua capacidade de distinguir entre dados reais e falsos, enquanto o gerador é ajustado para minimizar a capacidade do discriminador de fazer essa distinção. Quando o discriminador não consegue mais diferenciar entre amostras reais e geradas, o modelo atinge um estado ideal, onde o gerador aprendeu a distribuição dos dados reais.

2.5.2.3 Processo de Treinamento

O treinamento das GANs pode ser visto como o método minimax (SHANNON, 1950)(NEUMANN; MORGENSTERN, 1944), onde o gerador e o discriminador competem para otimizar suas respectivas funções de perda. A função objetivo do processo é dada pela Equação:

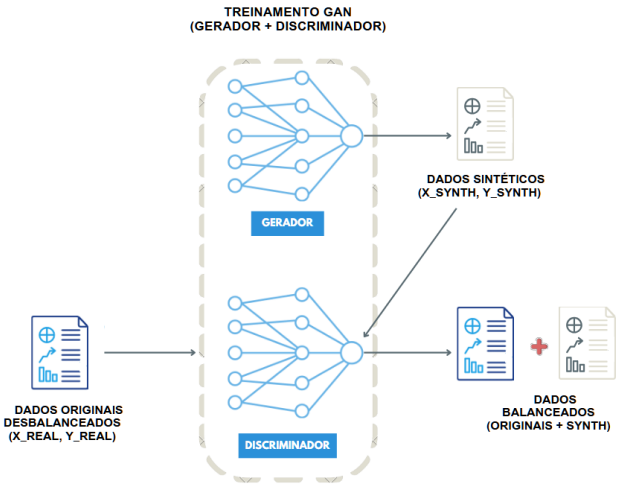
$$\min_G \max_D V(D,G) = E_{x\sim p_{dados}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.2}$$

Nessa equação, $V(D,G)$ representa o valor esperado da função de perda, onde o discriminador tenta maximizar a probabilidade de classificar corretamente as amostras reais e falsas, enquanto o gerador tenta minimizar a probabilidade de o discriminador identificar suas amostras como falsas. O equilíbrio de *Nash* é alcançado quando nenhum dos modelos pode melhorar seu desempenho sem prejudicar o outro.

2.5.3 Redes Generativas Adversárias para dados tabulares (GANs)

Este trabalho concentra-se no uso de modelos generativos para balanceamento de conjunto de dados tabulares; para tanto adaptamos uma GAN para geração de dados sintéticos tabulares que serão concatenados como os dados originais para igualar as classes. Além disso, comparamos a eficácia da GAN adaptada com outros métodos de balanceamento de dados, Figura 8.

Figura 8: GANs adaptada para dados Tabulares



Fonte: Elaborada pela autora.

O Algoritmo 6 recebe dados de treino (features X_train e labels y_train) e gera dados sintéticos (**linhas 1-2**) e os modelos são inicializados. O G (Gerador) transforma ruído aleatório (z_dim) em dados sintéticos com a mesma dimensionalidade dos dados reais ($n\_features$). O D (Discriminador) classifica se os dados são reais ou sintéticos. A função de perda e otimizadores usa *Binary Cross-Entropy (BCELoss)* e otimizadores Adam para ambos os modelos (**linhas 3-8**).

O treinamento da GAN ocorre em duas fases por época e por *batch* (**linhas 9-26**). O discriminador é treinado para classificar corretamente amostras reais (x_real) como 1 (y_real) (**linhas 13-14**). Gera amostras falsas (x_fake) a partir de ruído (z) e treina o discriminador para classificá-las como 0 (y_fake) (**linhas 15-17**). A perda total do discriminador (L_D) é a soma das perdas para dados reais (L_D_real) e falsos (L_D_fake), seguida de backpropagation (**linhas 18-23**).

O gerador é treinado para enganar o discriminador, fazendo-o classificar amostras falsas (x_fake) como reais (y_real). A perda do gerador (L_G) é calculada com base no erro do discriminador, e os pesos são ajustados (**linhas 21-24**).

A distribuição das classes é calculada (*counts*) e a classe majoritária é identificada (n_max) (**linhas 32-33**). Para cada classe minoritária, amostras sintéticas são geradas até igualar n_max (**linha 34**). O ruído (z) é criado com dimensão $n\_samples$ z_dim e gera dados sintéticos (X_synth) via gerador (G) (**linhas 35-37**). O *label* correspondente (y_synth) é atribuido e concatena com os dados originais ($X\_balanced$, $y\_balanced$) (**linhas 38-41**). Os dados sintéticos (X_synth) são retornados(**linha 44**).

Algoritmo 6: GAN para geração de dados sintéticos tabulares

Entrada: Conjunto de dados X_{train} (features), y_{train} (labels)
Saída: Dados sintéticos gerados
Inicialização:
$G \leftarrow \text{Generator}(z_{\text{dim}}, n_{\text{features}})$
$D \leftarrow \text{Discriminator}(n_{\text{features}})$
$\text{loss} \leftarrow \text{BCELoss}()$
$\text{opt}_G \leftarrow \text{Adam}(G.\text{params}, \text{lr})$
$\text{opt}_D \leftarrow \text{Adam}(D.\text{params}, \text{lr})$
Treinamento GAN:
for epoch $= 1$ **to** epochs **do**
  for batch $\in X_{\text{train}}$ **do**
    Discriminador:
    $x_{\text{real}} \leftarrow$ batch from X_{train}
    $y_{\text{real}} \leftarrow 1$
    $z \leftarrow \mathcal{N}(0,1)^{z_{\text{dim}}}$
    $x_{\text{fake}} \leftarrow G(z)$
    $y_{\text{fake}} \leftarrow 0$
    $\text{opt}_D.\text{zero\_grad}()$
    $L_D^{\text{real}} \leftarrow \text{loss}(D(x_{\text{real}}), y_{\text{real}})$
    $L_D^{\text{fake}} \leftarrow \text{loss}(D(x_{\text{fake}}.\text{detach}()), y_{\text{fake}})$
    $L_D \leftarrow L_D^{\text{real}} + L_D^{\text{fake}}$
    $L_D.\text{backward}()$
    $\text{opt}_D.\text{step}()$
    Gerador:
    $\text{opt}_G.\text{zero\_grad}()$
    $L_G \leftarrow \text{loss}(D(x_{\text{fake}}), y_{\text{real}})$
    $L_G.\text{backward}()$
    $\text{opt}_G.\text{step}()$
  end for
end for
Geração de Dados Sintéticos:
$\text{counts} \leftarrow \text{Histogram}(y_{\text{train}})$
$n_{\text{max}} \leftarrow \max(\text{counts})$
for class $\in$ unique(y_{train}) **do**
  $n_{\text{samples}} \leftarrow n_{\text{max}} - \text{counts}[\text{class}]$
  if $n_{\text{samples}} > 0$ **then**
    $z \leftarrow \mathcal{N}(0,1)^{n_{\text{samples}}} \times z_{\text{dim}}$
    $X_{\text{synth}} \leftarrow G(z)$
    $y_{\text{synth}} \leftarrow \text{class}$
    $X_{\text{balanced}} \leftarrow \text{Concatenate}(X_{\text{train}}, X_{\text{synth}})$
    $y_{\text{balanced}} \leftarrow \text{Concatenate}(y_{\text{train}}, y_{\text{synth}})$
  end if
end for
return X_{synth}

2.5.4 Rede Generativa Adversária para Dados Tabulares com Condicional - CTGAN

O CTGAN (*Generative Adversarial Networks For Tabular Data with Conditional*) é um método baseado em GAN para modelar a distribuição de dados tabulares e amostras sintéticas que se assemelham a esses dados (XU et al., 2019a).

O objetivo principal da CTGAN é superar o desafio de gerar dados sintéticos realistas que preservem as distribuições e correlações presentes nos dados reais. Ele é aplicado em situações em que a privacidade ou a segurança dos dados reais é uma preocupação, mas ainda é necessário ter acesso a um conjunto de dados sintético que se assemelhe ao conjunto de dados original (XU et al., 2019a).

O gerador em uma GAN é alimentado com um vetor amostrado de uma distribuição normal multivariada padrão (MVN). Ao treinar em conjunto com redes neurais discriminadoras ou críticas, obtém-se eventualmente uma transformação determinística que mapeia o MVN padrão na distribuição dos dados. Este método de treinamento de um gerador não considera o desequilíbrio nas colunas categóricas (XU et al., 2019a).

Especificamente, o objetivo é reamostrar eficientemente de forma que todas as categorias de atributos discretos sejam amostradas uniformemente (mas não necessariamente de forma uniforme) durante o processo de treinamento e recuperar a distribuição de dados reais (não reamostrados) durante o teste. Seja k^* a coluna discreta D_i^* que deve ser correspondida pelas amostras geradas r , então o gerador pode ser interpretado como a distribuição condicional de linhas dado aquele valor particular naquela coluna particular, ou seja, $r^* \sim \mathbb{P}_G(row|D_i = k^*)$. O gerador condicional, e uma GAN construída sobre ela é chamada de GAN condicional (XU et al., 2019a).

Integrar um gerador condicional na arquitetura de uma GAN requer lidar com os seguintes problemas: 1) é necessário criar uma representação para a condição, bem como preparar uma entrada para ela, 2) é necessário que as linhas geradas preservem a condição como ela é dada, e 3) é necessário que o gerador condicional aprenda a distribuição condicional dos dados reais (XU et al., 2019a), dada pela Equação 2.3.

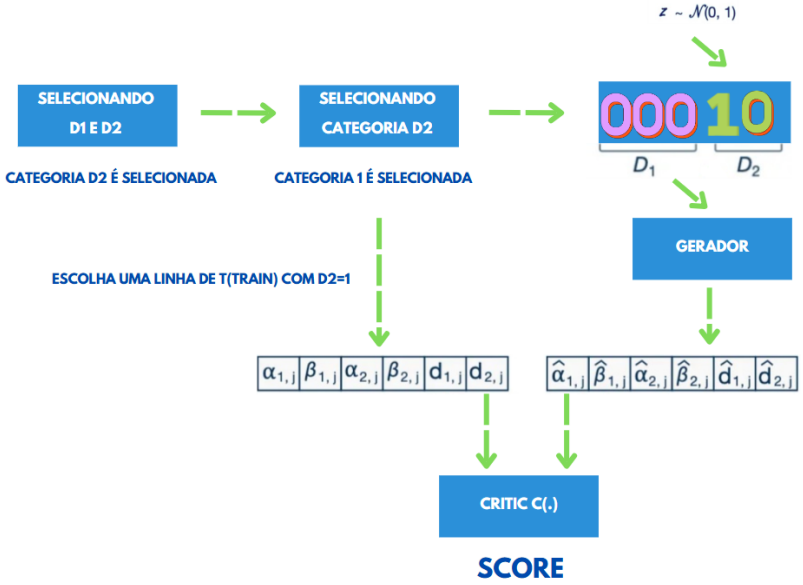
$$\mathbb{P}(row) = \sum_{k \in D_i^*} \mathbb{P}_G(row|D_i* = k^*)\mathbb{P}(D_i* = k^*). \tag{2.3}$$

A CTGAN apresenta um Gerador Condicional para lidar com os desafios impostos por categorias desequilibradas que geralmente levam ao colapso do modo GAN. Para tal, a entrada precisa ser preparada para que o gerador possa interpretar as condições e as linhas geradas precisam preservar uma condição de entrada.

Para tanto, a CTGAN considera um vetor condicional que, quando utilizado em um

treinamento amostra a amostra, faz muita diferença no que diz respeito ao uso do CTGAN (XU et al., 2019a), conforme Figura 9.

Figura 9: Arquitetura da Redes Adversariais Gerativas para Dados Tabulares com Condicional - CTGAN



Fonte: Elaborado pela autora, adaptado de (XU et al., 2019a)

É introduzido o conceito de vetor condicional como forma de indicar a condição ($D_i* = k^*$).[1] Lembrando de que todas as colunas discretas $D_1, ..., D_{ND}$ terminam como vetores one-hot $d_i = [d_i^{(k)}]$, para $k = 1, ..., |D_i|$ ser seu i-ésimo vetor d_i. Portanto, a condição pode ser expressa em termos desses vetores de máscara como (XU et al., 2019a), Equação 2.4:

$$m_i^k = \begin{cases} 1, & i = i^*,\ k = k^* \\ 0, & caso\ contrario \end{cases} \tag{2.4}$$

Sendo o vetor cond como $cond = m_1 \bigoplus ... \bigoplus m_{Nd}$. Por exemplo, para duas colunas discretas, $D_1 = 1, 2, 3$ e $D_2 = 1, 2$, a condição $(D_2 = 1)$ é expressa pelos vetores de máscara $m_1 = [0, 0, 0]$ e $m_2 = [1, 0]$; então $cond = [0, 0, 0, 1, 0]$ (XU et al., 2019a).

O Algoritmo 7 descreve o funcionamento da CTGAN. Primeiramente, o gerador G e o discriminador D são inicializados com pesos aleatórios. Essas redes são as componentes principais da arquitetura GAN (**linha 1**).

No *Loop* de Épocas: o algoritmo é executado por um número fixo de épocas T. Cada época representa uma passagem completa pelo conjunto de dados (**linha 2**).

No *Loop* de Lotes: em cada época, o conjunto de dados é dividido em lotes de tamanho

[1] O asterisco (*) marca qual coluna ($i*$) e qual valor ($k*$) devem ser "ativados"na máscara.

Algoritmo 7: Redes Adversariais Gerativas para Dados Tabulares com Condicional - CTGAN

1: **Entrada:** Dados tabulares reais X, número de épocas T, tamanho do lote B, vetor condicional *cond*.
2: **Saída:** Dados sintéticos gerados X_{syn}.
3: Inicializar o gerador G e o discriminador D com pesos aleatórios.
4: **for** época $t = 1$ até T **do**
5: **for** lote $b = 1$ até B **do**
6: Amostrar um lote de dados reais X_b de X.
7: Amostrar um vetor de ruído $z \sim \mathcal{N}(0, I)$.
8: Calcular o vetor condicional *cond* para o lote atual.
9: Gerar dados sintéticos $X_{syn} = G(z, cond)$.
10: Calcular a perda do discriminador L_D usando X_b e X_{syn}.
11: Atualizar os pesos de D para minimizar L_D.
12: Calcular a perda do gerador L_G usando X_{syn}.
13: Atualizar os pesos de G para minimizar L_G.
14: **end for**
15: **end for**
16: **return** X_{syn}.

B. Isso permite o treinamento em mini-lotes, que é mais eficiente computacionalmente (**linha 3**).

Depois um lote de dados reais X_b é amostrado do conjunto de dados original X. Esses dados serão usados para treinar o discriminador (**linha 4**).

Um vetor de ruído z é amostrado de uma distribuição normal multivariada padrão, denotada por $z \sim \mathcal{N}(0, I)$, onde $N(0, I)$ representa uma distribuição normal com média zero e matriz de covariância identidade (**linha 5**).

O vetor condicional *cond* é calculado para o lote atual. Esse vetor codifica informações sobre as condições desejadas (por exemplo, valores específicos em colunas categóricas) (**linha 6**).

O gerador G usa o vetor de ruído z e o vetor condicional *cond* para gerar dados sintéticos X_syn (**linha 7**).

A perda do discriminador L_D é calculada comparando as amostras reais X_b com as amostras sintéticas X_syn. O objetivo do discriminador é distinguir entre dados reais e sintéticos, (**linha 8**), Algoritmo 7.

Os pesos do discriminador D são atualizados para minimizar a perda L_D. Isso melhora a capacidade do discriminador de distinguir entre dados reais e sintéticos (**linha 9**).

A perda do gerador L_G é calculada com base nas amostras sintéticas X_syn. O objetivo do gerador é enganar o discriminador, fazendo com que ele classifique as amostras sintéticas

como reais (**linha 10**).

Os pesos do gerador G são atualizados para minimizar a perda L_G. Isso melhora a capacidade do gerador de produzir dados sintéticos realistas (**linha 11**). Após o treinamento, o algoritmo retorna os dados sintéticos X_syn gerados pelo gerador (**linha 12**).

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão de estudos que investigam o uso de modelos generativos no balanceamento de dados tabulares, destacando as abordagens propostas, os resultados alcançados e as principais limitações identificadas.

3.1 Redes Generativas Adversariais (GANs) para Balanceamento de Dados Tabulares

As GANs têm sido amplamente utilizadas para gerar dados sintéticos em cenários de classificação desbalanceada. Vu, Bui e Nguyen (2017) foram pioneiros ao aplicar GANs, especificamente AC-GAN, para balancear dados de tráfego de rede (SSH[1] vs. não SSH[2]), superando técnicas tradicionais como SMOTE e *BalanceCascade* em Precisão, AUC e *F1-Score*, apesar do tempo de treinamento mais longo. Wang et al. (2017) aprimoraram essa abordagem com o *PacketGAN*, uma GAN condicional que utiliza rótulos de tipo de tráfego para gerar amostras sintéticas balanceadas, obtendo resultados superiores em conjuntos de dados maiores e com diferentes classificadores.

Outras variações de GANs também foram propostas. Belenko et al. (2018) focaram na geração de dados sintéticos viáveis para redes M2M[3] usando CGAN, destacando a importância da convergência entre gerador e discriminador. O MedGAN, proposto por Park et al. (2018), foi desenvolvido para gerar registros de pacientes de saúde, preservando a complexidade das informações médicas e garantindo privacidade. Jordon, Yoon e Schaar (2018) integraram GANs com a estrutura *Private Aggregation of Teacher Ensembles* (PATE) no PATE-GAN, permitindo a geração de dados sintéticos com privacidade diferencial.

Wang et al. (2019) desenvolveram o FlowGAN, baseado em ACGAN (*GAN Auxiliary Classifier*), para gerar classes raras em dados de tráfego criptografado, resultando em classificadores com desempenho superior ao *Random Oversampling*. Xu et al. (2019b) propuseram o CTGAN, uma GAN condicional projetada para dados tabulares, que melhorou significativamente a Precisão e o Revocação para classes minoritárias. Fiore et al. (2019) utilizaram GANs para gerar dados sintéticos de transações fraudulentas de cartão de crédito, embora as melhorias não tenham sido estatisticamente significativas em comparação com o SMOTE.

[1]O SSH (*Secure Shell*) é um protocolo de rede amplamente utilizado para acessar e gerenciar dispositivos de forma segura em uma rede.

[2]Representa pacotes de rede que pertencem a outros tipos de tráfego (por exemplo, HTTP, FTP, DNS, etc.

[3]Redes M2M (Machine-to-Machine) referem-se à comunicação direta entre dispositivos eletrônicos, como sensores, máquinas, veículos ou equipamentos industriais, sem a necessidade de intervenção humana. Essa tecnologia permite que dispositivos troquem informações e realizem ações automaticamente, com base em dados coletados e processados Belenko et al. (2018).

Lei et al. (2020) propuseram o IGAFN (*Unbalanced Generative Adversarial Fusion Network*), que combina gerador e classificador em um único modelo para classificação binária de pontuação de crédito, superando métodos como SVM com SMOTE. Yilmaz, Masum e Siraj (2020) aplicaram GANs para balancear um conjunto de dados de detecção de intrusão (UGR'16)[4], obtendo Revocação, Precisão e *F1-Score* acima de 99%. Lee e Park (2021) utilizaram uma GAN básica para aumentar classes minoritárias em dados de detecção de anomalias, mostrando melhorias significativas em classes raras.

Engelmann e Lessmann (2021) investigaram a capacidade de GANs (especificamente *Wasserstein* GAN) para gerar dados estruturados, utilizando técnicas como a função de ativação *Gumbel-softmax* e adição de ruído gaussiano. O CTAB-GAN, proposto por (ZHAO et al., 2021), aprimorou o CTGAN com técnicas avançadas para modelar variáveis contínuas, categóricas e mistas, além de incluir um classificador para supervisão adicional. Kim et al. (2021) propuseram o OCT-GAN (*Neural Ode-Based Conditional Tabular Gans*), que utiliza Equações Diferenciais Ordinárias Neurais (*NODEs*) para capturar distribuições irregulares e multimodais.

Modelos baseados em difusão também têm ganhado destaque. Kim, Lee e Park (2022) propuseram o STaSy (*Score-based tabular data synthesis*) e o Sos (*Score-based oversampling for tabular data*), modelos baseados em difusão e pontuação para síntese e sobreamostragem de dados tabulares. Lee, Kim e Park (2023) desenvolveram o CoDi, que utiliza dois modelos de difusão intercondicionados para modelar colunas numéricas e categóricas. Kotelnikov et al. (2023) simplificaram essa abordagem com o TabDDPM (*Modelling Tabular Data with Diffusion Models*), que concatena características numéricas e categóricas. Suh et al. (2023) integraram *Auto-Encoder* com modelos de difusão no AutoDiff, resultando em um modelo poderoso para dados tabulares heterogêneos.

O TAEGAN (*Generating Synthetic Tabular Data For Data Augmentation*), proposto por (LI et al., 2024), é uma arquitetura baseada em GANs que gera dados tabulares de alta qualidade, especialmente eficaz para conjuntos de dados pequenos. O modelo CTAB-GAN+ não apenas gera dados sintéticos, mas também os ajusta para que sejam mais úteis em tarefas específicas de classificação e regressão, que são consideradas "a jusante"(ou seja, etapas posteriores no fluxo de processamento de dados). Isso é feito incorporando uma métrica de avaliação diretamente no processo de treinamento da GAN, garantindo que os dados gerados sejam otimizados para aplicações práticas, como previsões ou análises. Além disso, o CTAB-GAN+ também garante a privacidade dos dados por meio de técnicas de privacidade diferencial (ZHAO et al., 2024).

[4]hrefhttps://nesg.ugr.es/nesg-ugr16

3.2 Autoencoders Variacionais (VAEs) para balanceamento de dados tabulares

Os VAEs também têm sido explorados para o balanceamento de dados tabulares. Ruan et al. (2019) propuseram o CTVAE, uma variação do VAE que gera dados sintéticos condicionados à classe minoritária, preservando a distribuição original dos dados. Chuang e Huang (2023) desenvolveram o B-VAE (*Batched Variational AutoEncoders*), que treina múltiplos VAEs em lote para um treinamento mais eficaz do decodificador, superando outras abordagens de balanceamento. Xiao et al. (2024) criaram o VAE-DF, um modelo de ensemble que combina VAE com floresta profunda[5] para pontuação de crédito em finanças da Internet.

Abdulganiyu et al. (2025) propuseram o XIDINTFL-VAE (*XGBoost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder*), que combina perda focal por classe, VAE e XGBoost para detecção de intrusões, superando métodos tradicionais como SMOTE e ADASYN em Precisão e *F1-Score*.

3.3 Combinação de VAEs e GANs para Balanceamento de Dados Tabulares

Abordagens híbridas que combinam GANs (Generative Adversarial Networks) e VAEs (Variational Autoencoders) têm surgido como uma solução promissora. Anshelevich e Katz (2024) propuseram o T-VAE-GAN, que integra GANs e VAEs para gerar dados tabulares sintéticos de alta qualidade, preservando a distribuição dos dados originais. Huang et al. (2025) desenvolveram um modelo híbrido VAE-GAN focado na proteção da privacidade, capaz de gerar dados sintéticos indistinguíveis dos reais e robustos contra ataques de inferência.

Os ataques de inferência referem-se a tentativas de um adversário de extrair informações sensíveis ou confidenciais a partir dos dados sintéticos gerados. Esses ataques podem ocorrer quando um modelo de geração de dados não é suficientemente seguro, permitindo que um invasor deduza detalhes sobre os dados originais usados no treinamento. Por exemplo, um ataque de inferência pode revelar se um indivíduo específico estava presente no conjunto de dados original, comprometendo a privacidade. O modelo híbrido VAE-GAN, proposto por Huang et al. (2025), é projetado para mitigar esse risco, garantindo que os dados sintéticos sejam robustos contra tais tentativas de inferência, preservando assim a privacidade dos dados originais.

A Tabela 1 resume as principais características dos trabalhos revisados, destacando os conjuntos de dados utilizados, as arquiteturas propostas, as vantagens e as desvantagens de cada abordagem.

[5] A *Deep Forest* é um método de conjunto profundo baseado em Random Florest que pode realizar uma profundidade autoadaptável (ZHOU; FENG, 2019)

Tabela 1: Quadro comparativo dos trabalhos relacionados que utilizam modelos generativos para balanceamento de dados tabulares

Trabalho	Conjunto de Dados	Arquitetura	Vantagens	Desvantagens
(VU; BUI; NGUYEN, 2017)	Dados de tráfego de rede (SSH vs. não SSH)	GAN (AC-GAN)	Superou SMOTE e BalanceCascade em Precisão, AUC e *F1-Score*.	Tempo de treinamento maior.
(LEE; PARK, 2021)	Dados de detecção de anomalias	GAN básico	Melhorias significativas em classes raras.	Não superou métodos tradicionais em todas as métricas.
(WANG et al., 2019)	Dados de tráfego criptografado (ISCX2012)	GAN (FlowGAN)	Classificadores com melhor desempenho após balanceamento.	Dependência do uso de AC-GAN.
(WANG et al., 2017)	Dados de tráfego de rede	GAN (PacketGAN)	Resultados superiores em todas as métricas.	Requer rótulos de tipo de tráfego.
(YILMAZ; MASUM; SIRAJ, 2020)	Conjunto de dados de detecção de intrusão (UGR'16)	GAN	Melhorias significativas no desempenho.	Necessidade de pré-processamento.
(BELENKO et al., 2018)	Dados de redes M2M	GAN (CGAN)	Geração de dados sintéticos viáveis.	Dependência da convergência entre gerador e discriminador.
(XU et al., 2019b)	Dados tabulares	GAN (CTGAN)	Melhoria na Precisão e Revocação para classes minoritárias.	Complexidade na captura de distribuições complexas.
(FIORE et al., 2019)	Dados de transações fraudulentas de cartão de crédito	GAN	Melhorias ao dobrar casos fraudulentos.	Melhorias não estatisticamente significativas.
(LEI et al., 2020)	Dados de cartão de crédito e previsão de risco de crédito	GAN (IGAFN)	Superou SVM com SMOTE e GAN tradicional.	Complexidade na harmonização de dados.
(ENGELMANN; LESSMANN, 2021)	Dados estruturados para pontuação de crédito	GAN (Wasserstein GAN)	Geração de exemplos com distribuições semelhantes.	Necessidade de ajustes específicos.
(PARK et al., 2018)	Registros de pacientes de saúde	GAN (MedGAN)	Preservação das características médicas.	Foco específico em dados médicos.
(JORDON; YOON; SCHAAR, 2018)	Dados com garantia de privacidade diferencial	GAN (PATE-GAN)	Geração de dados com privacidade diferencial.	Complexidade na implementação.
(ZHAO et al., 2021)	Dados tabulares	GAN (CTAB-GAN)	Aprimoramento do CTGAN com técnicas avançadas.	Complexidade no pré-processamento.
(KIM et al., 2021)	Dados tabulares	GAN (OCT-GAN)	Captura de distribuições irregulares e multimodais.	Dependência de NODEs.
(KIM; LEE; PARK, 2022)	Dados tabulares	Modelos baseados em difusão (STaSy, Sos)	Alternativa promissora para síntese e sobreamostragem.	Complexidade na implementação.
(LEE; KIM; PARK, 2023)	Dados tabulares	Modelos de difusão (CoDi)	Captura da relação entre diferentes tipos de dados.	Complexidade na modelagem.
(KOTELNIKOV et al., 2023)	Dados tabulares	Modelos de difusão (TabDDPM)	Abordagem mais eficiente.	Limitação na capacidade de lidar com diferentes tipos de dados.
(SUH et al., 2023)	Dados tabulares	Modelos de difusão (AutoDiff)	Combinação de Auto-Encoder com modelos de difusão.	Complexidade na integração.
(LI et al., 2024)	Dados tabulares	GAN (TAEGAN)	Eficaz para conjuntos de dados pequenos.	Limitação em conjuntos de dados maiores.
(ZHAO et al., 2024)	Dados tabulares	GAN (CTAB-GAN+)	Geração de dados que preservam a privacidade.	Complexidade na implementação.
(RUAN et al., 2019)	Dados de alta dimensionalidade	VAE (CTVAE)	Geração de dados que preservam a distribuição original.	Complexidade na implementação.
(ABDULGANIYU et al., 2025)	Conjuntos de dados NSL-KDD e CSE-CIC-IDS2018	VAE (XIDINTFL-VAE)	Superou métodos tradicionais em Precisão e *F1-Score*.	Ligeira compensação no Revocação.
(CHUANG; HUANG, 2023)	Dados de detecção de intrusão	VAE (B-VAE)	Superou outras abordagens de balanceamento.	Complexidade na implementação.
(XIAO et al., 2024)	Conjuntos de dados de pontuação de crédito em finanças da Internet	VAE (VAE-DF)	Bom desempenho e profundidade autoadaptável.	Complexidade na combinação com floresta profunda.
(ANSHELEVICH; KATZ, 2024)	Dados tabulares	Híbrido (GAN + VAE)	Melhoria no desempenho do modelo.	Complexidade na implementação.
(HUANG et al., 2025)	Dados tabulares	Híbrido (VAE-GAN)	Gera dados sintéticos com alta utilidade e proteção de privacidade.	Complexidade e custo computacional.

Diferentemente de FlowGAN e PacketGAN, que são voltados para dados de tráfego de rede, o DSTO-GAN é aplicável a uma ampla variedade de conjuntos de dados tabulares. Métodos como STaSy, CoDi e TabDDPM baseiam-se em Modelos de Difusão (*Diffusion Models*), uma classe de modelos generativos que tem ganhado destaque devido à sua capacidade de gerar dados de alta qualidade em domínios como imagens, áudio e dados tabulares. Esses modelos utilizam um processo de difusão, inspirado em princípios físicos, no qual adicionam ruído progressivamente aos dados (*forward process*) e posteriormente aprendem a reverter esse processo (*reverse process*) para reconstruir os dados originais a

partir de uma distribuição de ruído.

Embora os modelos de difusão apresentem vantagens significativas, como a geração de dados altamente realistas e um treinamento mais estável em comparação com abordagens baseadas em GANs, eles também possuem desafios. Entre as principais limitações estão o alto custo computacional, e a necessidade de um ajuste fino de hiperparâmetros para obter um desempenho otimizado. Essas restrições podem comprometer a escalabilidade e a aplicabilidade prática desses modelos em cenários que demandam maior eficiência computacional e simplicidade na implementação.

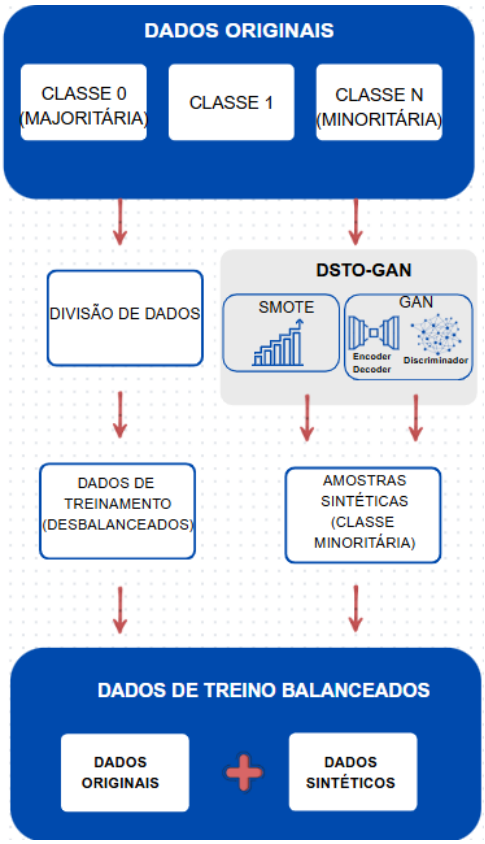
O CTAB-GAN+ (ZHAO et al., 2024) se destaca por sua abordagem especializada na geração de dados sintéticos, integrando privacidade diferencial para proteger informações individuais através da adição controlada de ruído durante o treinamento. Adicionalmente, ele é otimizado para ajuste fino em aplicações *downstream*, assegurando a utilidade prática imediata dos dados sintéticos em tarefas analíticas como classificação e regressão. Em contraste, o DSTO-GAN adota uma filosofia generalista, priorizando a adaptabilidade a diferentes estruturas de dados sem demandar reconfigurações extensivas.

Em contraste com outras técnicas, como FlowGAN, PacketGAN e CTAB-GAN+, ou modelos de difusão como STaSy, CoDi e TabDDPM, que demandam um alto custo computacional, o DSTO-GAN se destaca por sua versatilidade. Ele pode ser aplicado a uma ampla variedade de dados tabulares sem a necessidade de ajustes complexos. Em outras palavras, o DSTO-GAN diferentemente de métodos específicos como FlowGAN, PacketGAN e CTAB-GAN+, ou modelos de difusão como STaSy, CoDi e TabDDPM, que exigem muitos recursos computacionais, oferece a vantagem de ser facilmente utilizável em diversos tipos de dados tabulares, sem demandar configurações demoradas.

4 DEEPSMOTE TABULAR OPTIMIZE GAN (DSTO-GAN)

Este capítulo descreve a arquitetura e operação do *DeepSMOTE Tabular Optimized GAN* (DSTO-GAN), uma abordagem híbrida que combina SMOTE e GANs para tratamento de dados desbalanceados baseada no método DeepSMOTE (DABLAIN; KRAWCZYK; CHAWLA, 2021). Uma arquitetura GAN, composta por um *Encoder-Decoder* (para geração de representações latentes e reconstrução de dados) e um Discriminador (para classificação entre amostras reais e sintéticas), é treinada no conjunto parcialmente balanceado, Figura 10.

Figura 10: DSTO-GAN



Fonte: Elaborada pela autora

O *Encoder* mapeia os dados de entrada em um espaço latente parametrizado por média (*mean*) e variância (*logvar*), enquanto o *Decoder* reconstrói os dados a partir desse espaço, utilizando normalização por *BatchNorm* e funções de ativação *LeakyReLU* e *Tanh* para garantir estabilidade. O Discriminador, por sua vez, emprega camadas densas com *LeakyReLU* e uma saída sigmoide para estimar a probabilidade de uma amostra ser real, Figura 10.

O treinamento adversarial neste trabalho é aprimorado através da utilização de *Wasserstein GAN* (WGAN) (ARJOVSKY; CHINTALA; BOTTOU, 2017), uma variação das

redes generativas adversariais que introduz melhorias em relação às GANs tradicionais. A WGAN substitui a função de perda convencional baseada em divergência de *Jensen-Shannon* pela distância de *Wasserstein* (também conhecida como *Earth Mover's Distance*), que mede de forma mais eficiente a dissimilaridade entre as distribuições de dados reais e gerados.

A implementação incorpora ainda um *Gradient Penalty* (WGAN-GP), que impõe uma restrição de suavidade (condição *Lipschitz*) a função discriminadora, garantindo maior estabilidade no treinamento, evitando problemas como desaparecimento de gradientes, gerando amostras sintéticas mais realistas e diversificadas.

Esta abordagem é combinada com o SMOTE, onde: o SMOTE provê um balanceamento inicial através de interpolação linear e a WGAN-GP refina essas amostras, capturando melhor a distribuição subjacente dos dados. A integração dessas técnicas representa uma solução completa para o problema de desbalanceamento em conjuntos de dados tabulares, (Figura 10).

Tabela 2: Arquitetura detalhada dos componentes do DSTO-GAN

Componente	**Camadas**	**Ativações**	**Saída**
Encoder	Linear(entrada→64) → BN → LeakyReLU Linear(64→64) → BN → LeakyReLU Linear(64→64) → BN → LeakyReLU	LeakyReLU (0.2)	mean (16), logvar (16)
Decoder	Linear(16→64) → BN → LeakyReLU Linear(64→64) → BN → LeakyReLU Linear(64→64) → BN → LeakyReLU Linear(64→entrada)	LeakyReLU (0.2), Tanh (final)	Dados reconstruídos
Discriminador	Linear(entrada→64) → LeakyReLU Linear(64→64) → LeakyReLU Linear(64→32) → LeakyReLU Linear(32→1)	LeakyReLU (0.2), Sigmoid (final)	Probabilidade [0,1]

A seleção dos parâmetros da arquitetura DSTO-GAN foi baseada em experimentos e práticas consagradas na literatura de redes generativas. A dimensão latente de 32 unidades oferece um equilíbrio ideal entre capacidade representacional e eficiência computacional, sendo suficiente para capturar os padrões essenciais dos dados sem introduzir complexidade excessiva. As camadas ocultas com 64 neurônios permitem a modelagem adequada de transformações não-lineares, enquanto mantêm a arquitetura computacionalmente tratável, veja Tabela 3.

A taxa de aprendizado de 0,001 foi escolhida para garantir estabilidade durante o treinamento adversarial, sendo pequena o suficiente para evitar oscilações bruscas, mas grande o bastante para permitir convergência em tempo razoável. O *batch size* de 64 amostras proporciona estimativas confiáveis dos gradientes enquanto otimiza o uso de recursos computacionais, particularmente em GPUs, veja Tabela 3.

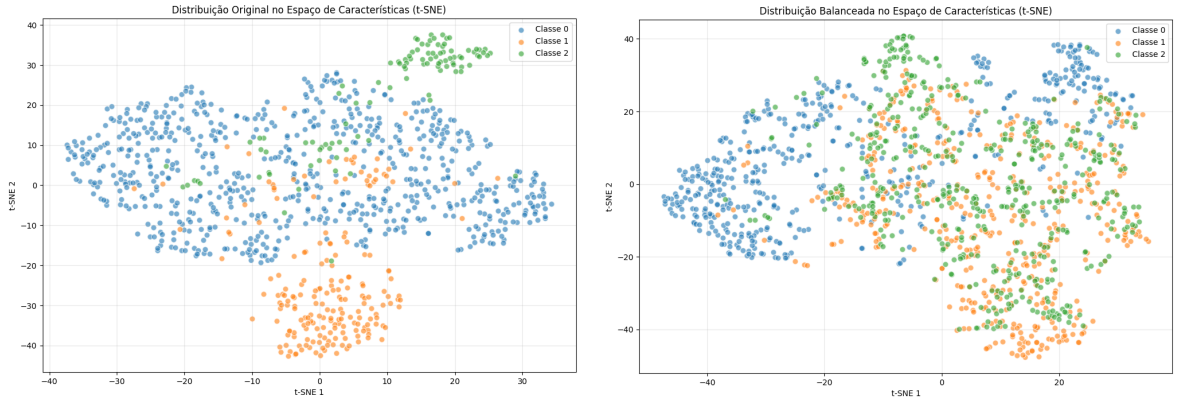
O coeficiente de *gradient penalty* ($\lambda_{gp} = 10$) segue a formulação padrão das *WGAN-GP*, sendo crítico para manter a condição *Lipschitz* durante o treinamento. O peso da *loss* de reconstrução ($\lambda_{rec} = 1$) foi determinado experimentalmente como o ponto ótimo que equilibra a qualidade das amostras geradas com a fidelidade aos dados originais. As 200 épocas de treinamento demonstraram ser suficientes para a convergência do modelo na maioria dos cenários, veja Tabela 3.

Tabela 3: Parâmetros da Arquitetura DSTO-GAN

Parâmetro	**Valor**
Dimensão Latente (n_z)	16
Dimensão Oculta (dim_h)	64
Taxa de Aprendizado (lr)	0,001
Batch Size	64
λ_{gp} (Gradient Penalty)	10
λ_{rec} (Reconstrução)	1
Épocas de Treinamento	100

A Figura 11 mostra a distribuição dos dados originais no espaço de características t-SNE[1] antes e após a aplicação do algoritmo DSTO para balanceamento de dados por sobreamostragem.

Figura 11: Representação gráfica do uso do algoritmo DSTO-GAN para balanceamento de dados



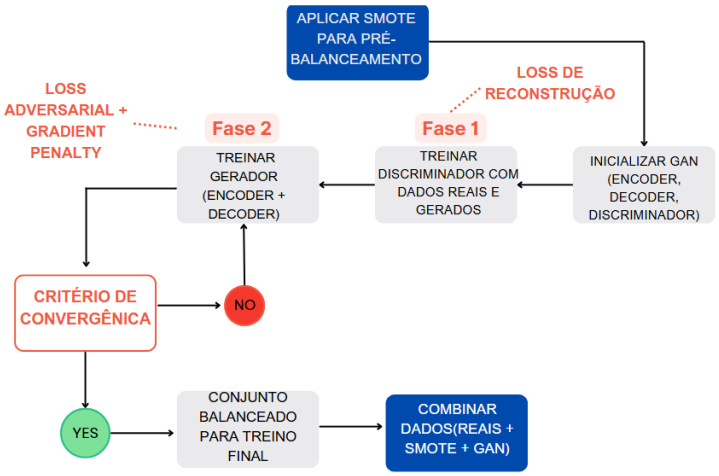
Fonte: Elaborada pela autora

[1] *t-Distributed Stochastic Neighbor Embedding* é uma técnica de redução de dimensionalidade não linear utilizada para visualização de dados em espaços de baixa dimensão (geralmente 2D ou 3D), é útil para representar conjuntos de dados complexos e de alta dimensionalidade de forma que padrões e agrupamentos se tornem visíveis.

4.1 Processo de Treinamento DSTO-GAN

O treinamento do método DSTO-GAN segue uma abordagem adversarial, baseada no paradigma das GANs, com otimizações adicionais para garantir estabilidade e fidelidade na geração de amostras sintéticas. O processo é dividido em duas fases principais, alternadas iterativamente: treinamento do discriminador e treinamento do gerador (composto pelo *Encoder* e *Decoder*), Figura 12.

Figura 12: Treinamento do DSTO-GAN



Fonte: Elaborada pela autora

O discriminador é treinado para maximizar sua capacidade de distinguir entre amostras reais (rótulo 1) e amostras sintéticas geradas pelo gerador (rótulo 0). Simultaneamente, o gerador busca minimizar essa habilidade do discriminador, produzindo amostras sintéticas cada vez mais realistas que possam enganá-lo. Esse jogo minimax é formalizado pela *loss adversarial*, que equilibra a competição entre os dois componentes, Figura 12.

Para assegurar que as amostras geradas preservem as características essenciais dos dados originais, uma *loss* de reconstrução (como o *Mean Squared Error*- MSE) é incorporada ao gerador. Essa penalidade calcula a diferença entre os dados de entrada e sua reconstrução pelo *Decoder*, garantindo que o espaço latente aprendido pelo *Encoder* mantenha informações estruturalmente relevantes, Figura 12.

Visando estabilizar o treinamento adversarial, especialmente em cenários de desbalanceamento crítico, o método emprega *Gradient Penalty*, técnica característica das *Wasserstein GANs* (WGANs). Essa penalidade restringe a norma dos gradientes do discriminador, evitando oscilações bruscas e colapso do gerador, o que é particularmente relevante quando a base de dados possui poucas amostras da classe minoritária, Figura 12.

O processo ocorre em duas fases alternadas, Figura 12:

- Fase 1 - Treino do Discriminador: São utilizados lotes contendo amostras reais e sintéticas, com o objetivo de atualizar os pesos do discriminador para maximizar sua acurácia na discriminação.
- Fase 2 - Treino do Gerador: O gerador é atualizado para minimizar tanto a *loss adversarial* (enganando o discriminador) quanto a *loss* de reconstrução (preservando a fidelidade dos dados).

4.2 Algoritmo DSTO-GAN

Algoritmo 8: DeepSMOTE Tabular Otimize GAN - DSTO-GAN

Entrada:
  X - Dados de treino (features)
  y - Rótulos das classes (desbalanceados)
  $args$ - Parâmetros da arquitetura (Tabela 3)
Saída:
  $X_{balanced}$, $y_{balanced}$ (Dados balanceados)
  Modelos treinados (Encoder, Decoder, Discriminador)

Balanceamento DSTO-GANX, y, $args$
Calcular $n\_to\_sample = \max(\text{contagens\_classes}) - \text{contagens\_classes}$
Fase SMOTE:
Aplicar SMOTE para gerar X_{smote}, y_{smote} com $k = 5$
Fase GAN:
Inicializar:
  Encoder E: $\text{Linear}(d_{in} \to 64) \to \text{BN} \to \text{LeakyReLU} \to \ldots \to (\mu_{32}, \sigma_{32})$
  Decoder D: $\text{Linear}(32 \to 64) \to \text{BN} \to \text{LeakyReLU} \to \ldots \to \text{Tanh}$
  Discriminador C: $\text{Linear}(d_{in} \to 256) \to \text{LeakyReLU} \to \ldots \to \text{Sigmoid}$
for época $= 1$ **to** $args.epochs$ **do**
  Treinamento Adversarial:
  1. Treinar C para maximizar $L_{adv} = \mathbb{E}[\log C(x)] + \mathbb{E}[\log(1 - C(D(z)))]$
  2. Treinar $(E + D)$ para minimizar $L_{total} = L_{adv} + \lambda_{rec}||x - D(E(x))||_2$
  3. Aplicar Gradient Penalty: $L_{gp} = \lambda_{gp}(\|\nabla C\|_2 - 1)^2$
end for
Gerar $X_{gan} = D(z)$ onde $z \sim \mathcal{N}(0, 1)$
Combinação:
$X_{balanced} \leftarrow X \cup X_{smote} \cup X_{gan}$
$y_{balanced} \leftarrow y \cup y_{smote} \cup y_{gan}$
return $(X_{balanced}, y_{balanced})$

O algoritmo 8 combina duas técnicas de geração de dados sintéticos (SMOTE e GAN) para balanceamento de classes, seguindo três fases principais:

1. Pré-processamento e SMOTE (Linhas 1-12) Entrada: Recebe os dados desbalanceados (X, y) e parâmetros da arquitetura ($args$) (Linhas 1-6).

   - Cálculo de amostras necessárias: Determina quantas amostras sintéticas são requeridas para cada classe minoritária, baseado na diferença entre a classe majoritária e as demais (Linha 10: n_to_sample = max(contagens_classes) - contagens_classes).

   - Geração SMOTE: Cria amostras sintéticas via interpolação linear entre vizinhos próximos (k=5) (Linha 13: X_smote, y_smote). Esta fase reduz o desbalanceamento inicial, facilitando o treino da GAN.

2. Treinamento da GAN (Linhas 15-28)

   - Inicialização dos modelos (Linhas 16-19):

     – Encoder (E): Reduz a dimensionalidade dos dados para um espaço latente Gaussiano (μ_{32}, σ_{32}) com camadas Lineares, BatchNorm (BN) e LeakyReLU.

     – Decoder (D): Reconstroi os dados a partir do espaço latente, usando Tanh na saída para estabilidade.

     – Discriminador (C): Classifica amostras como reais ou geradas, com saída sigmoide.

   - Loop de treinamento adversarial (Linhas 21-28):

     – Treino do Discriminador (C) (Linha 23): Maximiza a capacidade de distinguir amostras reais ($\mathbb{E}[\log C(x)]$) de geradas ($\mathbb{E}[\log(1 - C(D(z)))]$).

     – Treino do Gerador ($E + D$) (Linha 24): Minimiza a loss adversarial (para enganar C) e a loss de reconstrução ($||x - D(E(x))||_2$ para preservar características dos dados).

     – Gradient Penalty (Linha 25): Penaliza gradientes grandes em C (L_{gp}), estabilizando o treino (Wasserstein GAN).

3. Geração e Combinação Final (Linhas 30-33)

   - Geração de amostras GAN: Produz amostras sintéticas X_{gan} a partir de ruído Gaussiano $z \sim \mathcal{N}(0, 1)$ (Linha 30: X_gan = D(z)).

   - Balanceamento final: Combina dados originais (X), amostras SMOTE (X_{smote}) e amostras GAN (X_{gan}) em um conjunto balanceado (Linhas 32-33:

X_balanced, y_balanced).

4.3 Semelhanças e Diferenças entre DSTO-GAN e DeepSMOTE

Ambos os métodos, DeepSMOTE e DSTO-GAN, surgem como avanços em relação ao SMOTE clássico, incorporando técnicas de aprendizado profundo para melhorar a geração de amostras sintéticas em problemas de desbalanceamento de classes. Enquanto o DeepSMOTE (DABLAIN; KRAWCZYK; CHAWLA, 2021) combina SMOTE com *autoencoders* para operar em espaços latentes. O DSTO-GAN integra SMOTE a uma arquitetura GAN, adicionando um mecanismo adversarial para refinar a geração. Esta seção explora suas similaridades, diferenças fundamentais e implicações teóricas. Ambos os métodos compartilham três princípios fundamentais:

1. Uso do SMOTE como base: Aplicam SMOTE para gerar amostras sintéticas iniciais, garantindo um pré-balanceamento que facilita o treinamento de componentes profundos.

2. Compressão de dados via espaços latentes: O DeepSMOTE: Projeta os dados em um espaço latente usando um *autoencoder*. DSTO-GAN: Utiliza um *encoder* (parte da GAN) para mapear os dados em um espaço Gaussiano.

3. Objetivo comum: Reduzir a dimensionalidade e capturar características essenciais antes da geração.

4. Combinação de técnicas determinísticas e probabilísticas: Ambos mesclam interpolação linear (SMOTE) com modelagem não-linear (redes neurais), buscando equilibrar controle e flexibilidade na geração.

DeepSMOTE e DSTO-GAN adotam paradigmas distintos de geração, fundamentados em diferentes princípios matemáticos e computacionais, Tabela 4.

DeepSMOTE baseia-se em um paradigma determinístico, utilizando *autoencoders* para projetar os dados em um espaço latente de características, onde o SMOTE é aplicado para interpolação linear. Esse método assume que a estrutura de vizinhança no espaço latente preserva relações semânticas válidas no espaço original, Tabela 4.

No DeepSMOTE os dados são mapeados para um espaço latente de menor dimensionalidade via *autoencoder*, onde E é o *encoder* e D o *decoder*.

$$E : \mathcal{X} \to \mathcal{Z}, \quad D : \mathcal{Z} \to \mathcal{X} \tag{4.1}$$

Interpolação no espaço latente

$$\mathbf{z}_{\text{new}} = \mathbf{z}_i + \lambda(\mathbf{z}_j - \mathbf{z}_i), \quad \lambda \sim \mathcal{U}(0,1) \tag{4.2}$$

com $\mathbf{z}_i, \mathbf{z}_j$ vizinhos no espaço latente $\mathcal{Z}$.

Reconstrução: As amostras sintéticas são projetadas de volta ao espaço original

$$\mathbf{x}\text{novo} = D(\mathbf{z}\text{novo}) \tag{4.3}$$

DSTO-GAN emprega um paradigma adversarial, onde a geração de amostras é formulada como um problema de otimização minimax entre um gerador (que produz dados sintéticos) e um discriminador (que os classifica como reais ou falsos). Essa abordagem é fundamentada na teoria de Jogos de *Nash*, com o objetivo de alcançar um equilíbrio onde as amostras geradas são indistinguíveis das reais, Tabela 4.

No DSTO-GAN o SMOTE é usado inicialmente para reduzir o desbalanceamento crítico.

A geração dos dados é realizada de forma adversarial:

$$G : \mathbf{z} \to \mathbf{x}_{\text{synth}}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{4.4}$$

$$D : \mathbf{x} \to [0,1] \tag{4.5}$$

Através da Função Objetivo:

$$\min_G \max_D \mathbb{E}[\log D(\mathbf{x})] + \mathbb{E}[\log(1 - D(G(\mathbf{z})))] + \lambda_{gp} L_{gp} \tag{4.6}$$

O gerador é regularizado por uma *loss* de reconstrução para preservar fidelidade aos dados originais

$$L_{\text{rec}} = |\mathbf{x} - G(E(\mathbf{x}))|_2 \tag{4.7}$$

Tabela 4: Comparação Quantitativa dos Paradigmas de Geração

Critério	**DeepSMOTE**	**DSTO-GAN**
Base Matemática	Álgebra linear + Espaços latentes	Teoria de jogos + Otimização minimax
Tipo de Geração	Determinística (interpolação)	Probabilística (amostragem adversarial)
Complexidade	$\mathcal{O}(n^2)$ (k-NN no espaço latente)	$\mathcal{O}(n \cdot \text{épocas})$ (GAN)
Expressividade	Limitada por linearidade	Alta (captura não-linearidades)

5 MATERIAIS E MÉTODOS

Esta pesquisa aborda o desafio do balanceamento de dados tabulares, com foco na implementação e adaptação do algoritmo DeepSMOTE, denominado DeepSMOTE Tabular Optimize GAN (DSTO-GAN). O DSTO-GAN é uma versão adaptada do DeepSMOTE, projetada especificamente para dados tabulares, que combina a técnica de *oversampling* do DeepSMOTE com uma arquitetura de Rede Adversarial (GAN). Essa integração visa melhorar a geração de amostras sintéticas para classes minoritárias, garantindo um balanceamento mais eficaz e a preservação da distribuição original dos dados.

A pesquisa foi desenvolvida como um estudo experimental, incorporando elementos de pesquisa aplicada e desenvolvimento de software. A implementação do método DSTO-GAN, realizada por meio de um processo iterativo, envolveu a análise crítica da literatura, experimentação sistemática e refinamento contínuo dos procedimentos metodológicos.

A Figura 13 apresenta uma visão geral da metodologia adotada neste trabalho, abrangendo desde o levantamento bibliográfico inicial e o desenvolvimento do método proposto até a etapa de validação do DSTO-GAN, tanto nas bases de dados utilizadas quanto no estudo de caso selecionado.

Figura 13: Metodologia de Pesquisa

Fonte: Elaborada pela autora

As subseções seguintes descrevem cada uma dessas etapas.

5.1 Descrição dos Processos Metodológicos para as Etapas Iniciais da Pesquisa

Na fase inicial da pesquisa, são realizadas duas etapas cruciais: o levantamento bibliográfico e a implementação do DeepSMOTE Tabular Optimize GAN (DSTO-GAN).

1. *Levantamento Bibliográfico*: Esta etapa foi essencial para estabelecer uma base de conhecimento sobre o tema da pesquisa. Inicialmente, foi conduzido um levantamento bibliográfico, visando compreender os conceitos fundamentais relacionados ao desequilíbrio de classes em conjuntos de dados tabulares. Além disso, durante o levantamento bibliográfico, foram explorados os diferentes métodos de balanceamento de dados existentes na literatura. Compreender os princípios e as aplicações de cada método foi crucial para a tomada de decisão sobre qual técnica de balanceamento seria mais adequada para a pesquisa em questão.

2. *Implementação do método DSTO-GAN*: Com base no conhecimento adquirido durante o levantamento bibliográfico, foi realizada a implementação do DeepSMOTE Tabular Optimize GAN (DSTO-GAN). A implementação do DSTO-GAN envolveu a adaptação do algoritmo original DeepSMOTE para atender às especificidades de conjuntos de dados tabulares. Durante o processo de implementação, desenvolvido em *Python*, foram considerados aspectos fundamentais, como a escolha de bibliotecas e *frameworks* apropriados (e.g., *TensorFlow*, *PyTorch*, *scikit-learn*), a definição dos parâmetros do modelo (e.g., dimensão do espaço latente, taxas de aprendizado, número de épocas) e a validação da implementação por meio de testes e experimentos preliminares. A estrutura do DSTO-GAN foi projetada para lidar com as características distintas de dados tabulares, que diferem significativamente de dados não estruturados, como imagens ou textos, por exemplo. O DSTO-GAN integra uma arquitetura de Rede Adversarial (GAN) ao *framework* do DeepSMOTE, com o objetivo de gerar amostras sintéticas de alta qualidade para classes minoritárias. O Gerador é responsável por criar amostras a partir do espaço latente, enquanto o Discriminador avalia a qualidade dessas amostras em relação aos dados reais. Essa abordagem adversarial permite que o Gerador aprenda a produzir amostras que preservem a distribuição original dos dados, mantendo as relações complexas entre os atributos.

3. *Validação do método DSTO-GAN*: Para validar o método proposto, foram utilizadas 48 bases de dados, com diferentes características quanto a quantidade de instâncias, atributos, desbalanceamento, dentre outros aspectos. Além disto, o método foi aplicado em um estudo de caso com dados reais sobre o vírus da Zika. Maiores detalhes da validação e estudo de caso encontram-se na seção seguinte.

4. *Disponibilização do DSTO-GAN*: A biblioteca dsto_gan está disponível no PyPI. Os conjuntos de dados não devem conter valores ausentes (NaN). Esses valores devem

ser tratados antes de usar a biblioteca. A biblioteca não aceita variáveis categóricas. Se o conjunto de dados contiver colunas categóricas, elas devem ser convertidas em valores numéricos usando técnicas como *one-hot encoding* ou *label encoding*, por exemplo.

5.2 Validação do Método Proposto - Estudo experimental

1. *Escolha das bases de dados para realização dos experimentos*: Para validar a eficácia do DSTO-GAN, foram conduzidos testes em um conjunto diversificado de 48 bases de dados tabulares. Para uma análise mais estruturada, as características dos conjuntos de dados — instâncias, atributos e índice de desbalanceamento (IR) — foram classificadas em três categorias: pequeno (P), médio (M) e grande (G). Essa categorização foi baseada no cálculo dos percentis, que dividem os dados em três grupos de tamanho equivalente. Para as instâncias, os intervalos definidos foram: pequeno (P) de 116 a 748, médio (M) de 749 a 2665 e grande (G) de 2666 a 110.204. Para os atributos, os intervalos estabelecidos foram: pequeno (P) de 4 a 9, médio (M) de 10 a 20 e grande (G) de 21 a 309. Já para o índice de desbalanceamento (IR), os intervalos foram: pequeno (P) de 1,2 a 2, médio (M) de 3 a 11 e grande (G) de 12 a 439. A escolha das bases de dados considerou os seguintes aspectos:

    - *Diferentes dimensões*: Foram selecionados conjuntos de dados com diferentes dimensões, abrangendo variações na quantidade de instâncias e de atributos. Assim, o número de instâncias e de atributos foi categorizado como pequeno (P), médio (M) ou grande (G). Maiores detalhes sobre essa classificação encontra-se no Apêndice A, página 129.

    - *Diferentes tipos de dados:* as nuances inerentes aos dados tabulares, como a coexistência dos seguintes tipos de atributos: inteiros (I), real (R), *booleano* (B) e categórico (C) (ver Apêndice A, página 129);

    - *Diferentes cardinalidade de classe:* A presença de conjuntos de dados binários (B) e multiclasse (M) na avaliação permitiu verificar a adaptabilidade do DSTO-GAN a diferentes naturezas de problemas de classificação; e

    - *Diferentes graus de desbalanceamento:* finalmente, a inclusão de dados com diferentes graus de desbalanceamento (medidos pelo Índice de Desbalanceamento - IR) possibilitou analisar o impacto do desbalanceamento no desempenho do modelo e sua capacidade de mitigar seus efeitos negativos. Assim, a quantidade de amostras (Quant) de cada classe, para cada base de dados, também foi apresentada.

    Na classificação geralmente é esperado que quanto maior a extensão do

desequilíbrio, pior o desempenho da classificação, e, portanto, esta métrica é bastante apropriada para mostrar uma correlação negativa com o desempenho da classificação (ZHU; GUO; XUE, 2020).

A medida de extensão de desequilíbrio de classe mais popular é o *Imbalance Ratio* (IR), calculada como a razão entre o tamanho da amostra da maior classe majoritária e o da menor classe minoritária (veja Equação 5.1). Assim, quanto maior o valor da IR, maior a extensão do desequilíbrio (ZHU; GUO; XUE, 2020).

$$IR = \frac{N_{maj}}{N_{min}} \tag{5.1}$$

onde N_{maj} é o tamanho da amostras da classe majoritária e N_{min} é o tamanho da amostra da classe minoritária. Quando $IR = 1$, temos um conjunto de dados exatamente balanceado. Quando $IR > 1$, quanto maior o IR, maior a extensão do desequilíbrio do conjunto de dados.

No entanto, a IR não é uma medida eficaz de extensão de desequilíbrio quando temos várias classes, porque as informações das classes com tamanhos de amostra entre os dois extremos não são consideradas (ZHU; GUO; XUE, 2020).

A Tabela A apresenta as 48 (quarenta e oito) bases de dados com suas respectivas características, página 129. Essas variações nas bases de dados fornecem uma visão abrangente das propriedades dos conjuntos de dados utilizados, permitindo uma avaliação mais robusta do desempenho do DSTO-GAN em diferentes cenários de desbalanceamento, tipos de dados e complexidade.

2. *Pré-Processamento das bases de dados:*

Após a seleção dos conjuntos de dados, foi realizado o pré-processamento das bases, que englobou a imputação de dados ausentes, a codificação dos atributos, a normalização dos dados e o balanceamento de classes.

A imputação de dados ausentes foi realizada utilizando a média, no caso de dados numéricos, e a moda para dados nominais.

Quanto à codificação, os atributos nominais ordinais foram codificados utilizando o ordinal *encoding* e a codificação *one-hot encoding* foi utilizada nos atributos nominais não ordinais. A codificação *One-Hot* é uma técnica utilizada para transformar variáveis categóricas em representações numéricas, essencial para algoritmos de aprendizado de máquina que exigem entradas numéricas (AGGARWAL; ZHAI, 2012; HAN; KAMBER; PEI, 2012). Cada valor categórico é convertido em um vetor binário, onde a presença de uma categoria é indicada por 1 e sua ausência por 0 (HAN; KAMBER; PEI, 2012).

Para a normalização dos dados, foi utilizado o método Min-Max, padronizando os atributos entre os valores 0 e 1 (HAN; KAMBER; PEI, 2011).

Para o balanceamento dos dados, foram utilizados os algoritmos de sobreamostragem DSTO-GAN (proposto neste trabalho), SMOTE, DCGAN e CTGAN e subamostragem (RUS). O balanceamento foi aplicado exclusivamente aos conjuntos de treinamento dentro de cada *fold*, evitando vazamento de dados. Além disso, testamos também os resultados com os dados não balanceados.

3. *Treinamento e testes dos algoritmos de aprendizado de máquina:*

   Para a construção e avaliação dos modelos gerados, foram selecionados os seguintes classificadores: *Random Forest*, *K-Nearest Neighbors* (KNN), *Neural Network*, *XGBoost* e *Decision Tree*, Tabela 5.

   A *Neural Network* desenvolvida foi uma rede rasa, pois possui apenas uma camada oculta (hidden_layer_sizes=(100,)). A arquitetura é composta por uma camada de entrada (tamanho definido pelos dados de entrada), uma camada oculta com 100 neurônios e uma camada de saída (tamanho definido pelo número de classes). A configuração foi uma escolha comum para problemas de classificação moderadamente complexos, veja Tabela 5.

   A *Decision Tree* implementada é um árvore não podada para classificação, usando o critério Gini para medir a qualidade das divisões. Os parâmetros padrão (max_depth=None, min_samples_split=2, etc.) indicam que esta é uma configuração não-restritiva, que permite que a árvore cresça até sua máxima profundidade. O calculo da impureza (criterion='gini') em vez de *entropy* é mais rápido computacionalmente que entropia. Com a escolha de max_depth=None a árvore cresce até que todas as folhas sejam puras ou até que todos os *splits* atendam a min_samples_split, capturando as relações complexas nos dados. Permite que um nó seja dividido mesmo com apenas 2 amostras (min_samples_split=2) e folhas com uma única amostra (min_samples_leaf=1). Essa configuração específica é ideal para entender o máximo desempenho (não-generalizado) nos dados de treino, sendo usada como componente em ensembles (como *Random Forests*) e situações onde a interpretabilidade não é crucial e *overfitting* será controlado por outras técnicas (ex: *pruning* posterior), veja Tabela 5.

   A configuração do algoritmo *Random Forest* empregou 100 árvores *(n_estimators)*, um valor padrão que otimiza o desempenho em relação ao custo computacional. A profundidade máxima das árvores foi configurada como None, permitindo que cresçam até que todas as folhas sejam puras; essa abordagem é eficaz porque a amostragem aleatória inerente ao algoritmo já previne o *overfitting*,

e árvores completas conseguem capturar relações complexas nos dados. Para o crescimento das árvores, o número mínimo de amostras para dividir um nó (min_samples_split=2) e o número mínimo de amostras em uma folha *(min_samples_leaf=1)* foram definidos de forma a permitir a máxima flexibilidade e o crescimento livre das árvores. A amostragem aleatória com reposição *(bootstrap=True)* foi crucial para a diversidade das árvores, pois cada uma é treinada em um subconjunto diferente dos dados. Para agilizar o processo, todos os núcleos da CPU foram utilizados para treinamento paralelo *(n_jobs=-1)*, aproveitando a capacidade de paralelização do *Random Forest*. Por fim, uma semente aleatória *(random_state=42)* foi definida para garantir a reprodutibilidade dos resultados, facilitando a depuração e comparações, veja Tabela 5.

No algoritmo, a configuração utilizada para classificação é não-paramétrica e baseada na distância Euclidiana, tornando-o ideal para problemas de pequeno a médio porte com relações locais bem definidas. É um bom ponto de partida antes de explorar modelos mais complexos. Utiliza-se 5 vizinhos (*n_neighbors=5*) na votação, um valor que equilibra a captura de padrões locais e a suavização de ruídos. Todos os vizinhos têm o mesmo peso (*weights='uniform'*), presumindo que são igualmente relevantes. A otimização do algoritmo (*algorithm='auto'*) permite que o sistema escolha automaticamente entre *'brute'*, *'kd_tree'* ou *'ball_tree'*, selecionando o mais eficiente com base nos dados. O tamanho das folhas (*leaf_size=30*) em estruturas como *KDTree* ou *BallTree* é um valor intermediário para equilibrar a velocidade das consultas e o uso de memória. A distância Euclidiana (L2) é a métrica padrão para dados contínuos, Tabela 5.

A configuração do algoritmo *XGBoost* utiliza 100 árvores (n_estimators), um número padrão que equilibra aprendizado e custo computacional. A profundidade máxima de cada árvore (max_depth) é definida como 3, o que ajuda a evitar *overfitting* e mantém o modelo interpretável. A taxa de aprendizado (learning_rate) de 0.1 controla a contribuição de cada árvore, enquanto *subsample* e *colsample_bytree* são configurados como 1.0 para usar todas as amostras e *features*, respectivamente, sem aleatoriedade. O parâmetro *gamma* é 0, o que significa que os nós são divididos mesmo com ganho mínimo. A regularização L1 (*reg_alpha*) está desativada (0), e a regularização L2 (*reg_lambda*) é definida como 1 para suavizar os pesos e prevenir *overfitting*. Uma semente aleatória (*random_state=42*) garante a reprodutibilidade do modelo, e a paralelização (*n_jobs=-1*) utiliza todos os núcleos da CPU para acelerar o treinamento, veja Tabela 5.

A escolha desses algoritmos foi fundamentada em suas características complementares e ampla utilização na literatura para tarefas de classificação. O *Random Forest* foi incluído por sua robustez e capacidade de lidar com dados

Tabela 5: Parâmetros dos classificadores

Classificador	**Parâmetros**
Decision Tree	max_depth=None, min_samples_split=2, min_samples_leaf=1, criterion='gini', random_state=42
Random Forest	n_estimators=100, max_depth=None, min_samples_split=2, min_samples_leaf=1, bootstrap=True, n_jobs=-1, random_state=42
Neural Network	hidden_layer_sizes=(100), activation='relu', solver='adam', alpha=0,0001, learning_rate='constant', max_iter=200, random_state=42
KNN	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2
XGBoost	n_estimators=100, max_depth=3, learning_rate=0,1, subsample=1.0, colsample_bytree=1.0, gamma=0, reg_alpha=0, reg_lambda=1, random_state=42, n_jobs=-1

complexos e não lineares, além de ser menos suscetível a *overfitting* (BREIMAN, 2001). O KNN foi escolhido por sua simplicidade e eficácia em capturar padrões locais nos dados, sendo útil para avaliar a qualidade da geração de amostras sintéticas (COVER; HART, 1967). As Redes Neurais foram selecionadas por sua habilidade em modelar relações não lineares complexas e por serem amplamente aplicadas em problemas de classificação (LECUN; KAVUKCUOGLU; FARABET, 2010). O *XGBoost* foi incluído por seu alto desempenho em competições de ciência de dados e por sua eficiência em otimização e generalização (CHEN, 2015). Por fim, a Árvore de Decisão foi escolhida por sua interpretabilidade e capacidade de servir como base para comparação com modelos mais complexos (QUINLAN, 2014). A combinação desses classificadores permite uma análise abrangente e confiável do impacto dos métodos de balanceamento na qualidade dos dados e no desempenho dos modelos de classificação.

Esta etapa foi organizada em duas fases principais: (i) a divisão dos conjuntos de dados em uma proporção de 80% para treino e 20% para teste, e (ii) a aplicação de validação cruzada nos 80% destinados ao treino. O balanceamento foi aplicado exclusivamente às amostras de treino, preservando a integridade dos dados de teste para avaliação imparcial do modelo. A metodologia de treino e teste é ilustrada na Figura 14 e descrita a seguir.

- **Divisão dos dados**: o conjunto de dados foi dividido em 10 *folds* (k=10) para validação cruzada.

Figura 14: Metodologia Treino e Teste - DeepSMOTE Tabular Optimize

Fonte: Elaborada pela autora

- **Balanceamento**: aplicou-se a técnica de sobreamostragem ou subamostragem apenas no conjunto de treinamento.
- **Treinamento do modelo**: o modelo foi treinado com o conjunto de treinamento balanceado.
- **Avaliação**: o desempenho foi avaliado no conjunto de teste, desbalanceado, utilizando métricas como Precisão, Revocação e *F1-score*.

É importante mencionar que a validação cruzada é uma técnica de reamostragem utilizada para avaliar e calibrar modelos preditivos, garantindo uma estimativa confiável de sua capacidade de generalização. Neste estudo, empregamos a validação cruzada com *k-folds*, sendo k=10, onde o conjunto de dados é dividido em k subconjuntos. Em cada iteração, um *fold* é utilizado para validação, enquanto os demais $k-1$ *folds* são usados para treinamento. Essa abordagem evita problemas como superajuste e viés de seleção, proporcionando uma avaliação robusta do desempenho do modelo (KRSTAJIC et al., 2014; VAROQUAUX et al., 2017; SERAJ et al., 2023).

Após a conclusão do ciclo de validação cruzada, realizou-se uma avaliação global do desempenho do modelo, considerando sua capacidade de lidar com o desequilíbrio de classes.

4. *Métricas de Avaliação:* A avaliação experimental da abordagem proposta foi realizada empregando métricas de desempenho adequadas à natureza dos problemas

de classificação abordados e amplamente reconhecidas na literatura, tais como Precisão, Revocação, *F1-Score*, visando uma avaliação abrangente e comparativa do desempenho da abordagem proposta em relação a métodos tradicionais de balanceamento de dados.

- Precisão: É a proporção de verdadeiros positivos (VP) em relação ao total de itens classificados como positivos pelo modelo (verdadeiros positivos + falsos positivos), mede a capacidade do modelo de classificar corretamente os exemplos positivos (YACOUBY; AXMAN, 2020).

$$Precisao = \frac{VP}{VP + FP} \tag{5.2}$$

- Revocação (Revocação/Sensibilidade): É a proporção de verdadeiros positivos (VP) em relação ao total de itens que realmente pertencem à classe positiva (verdadeiros positivos + falsos negativos). O Revocação mede a capacidade do modelo de encontrar corretamente todos os exemplos positivos (YACOUBY; AXMAN, 2020).

$$Revocacao = \frac{VP}{VP + FN} \tag{5.3}$$

  sendo: VP = *Verdadeiro Positivo*, VN = *Verdadeiro Negativo*, FP = *Falso Positivo* e FN= *Falso Negativo*

- *F1-Score* é a média harmônica da Precisão e Revocação, e indica a qualidade geral do modelo, dada pela Equação 5.4. O F1-Score é uma métrica de avaliação de desempenho amplamente utilizada em tarefas de classificação, especialmente quando as classes estão desbalanceadas. Ele é uma medida da acurácia geral do modelo, combinando tanto a acurácia quanto o Revocação (revocação) do sistema(YACOUBY; AXMAN, 2020).

$$F - Score = \frac{2 \times Precisao \times Revocacao}{Precisao + Revocacao} \tag{5.4}$$

As métricas escolhidas (F1-Score, Precisão, Revocação) são adequadas para avaliar balanceamento, pois focam nos erros mais relevantes em dados desbalanceados. Em estudos subsequentes, pretendemos incorporar o Coeficiente Kappa de Cohen como parte da avaliação, juntamente com as métricas já adotadas. Isso permitirá uma análise ainda mais abrangente, considerando não apenas os erros de classificação, mas também a concordância do modelo além do esperado aleatoriamente.

5. *Comparação dos Algoritmos de Balanceamento de Dados*:

Nesta etapa, foram comparados os desempenhos dos algoritmos DCGAN, CTGAN, SMOTE, RUS e o método proposto DSTO-GAN em um conjunto diversificado de bases de dados desbalanceados. A seleção desses métodos foi baseada em suas abordagens distintas e complementares para lidar com o desbalanceamento de classes. O RUS (*Random Under-Sampling*) foi escolhido por sua simplicidade e eficácia em reduzir a classe majoritária, equilibrando rapidamente a distribuição das classes. O SMOTE (*Synthetic Minority Over-sampling Technique*) foi incluído por sua capacidade de gerar amostras sintéticas da classe minoritária, preservando a estrutura dos dados originais. Já o DCGAN (*Deep Convolutional Generative Adversarial Network*) e o CTGAN (*Conditional Tabular GAN*) foram selecionados por sua habilidade em gerar dados sintéticos de alta qualidade, utilizando redes neurais profundas, com o CTGAN sendo especialmente adaptado para dados tabulares. Por fim, o método proposto DSTO-GAN foi incorporado para avaliar sua eficácia em relação às técnicas tradicionais e estado da arte. Foram consideradas métricas de avaliação como Precisão, Revocação e *F1-score* para avaliar a capacidade de cada algoritmo em lidar com o desbalanceamento de classes e preservar a qualidade dos dados.

6. *Análise Estatística dos Resultados*:

   A análise dos resultados foi conduzida mediante um levantamento estatístico, visando discernir disparidades de significância entre os diferentes métodos de balanceamento (CTGAN, GAN, DSTO-GAN, SMOTE, RUS, Unbalanced).

   Para comparar estatisticamente, seguimos uma abordagem sequencial. Primeiro utilizamos o teste Normalidade Shapiro-Wilk (GONZÁLEZ-ESTRADA; COSMES, 2019) para verificar se os dados seguem uma distribuição normal. Porque muitos testes paramétricos como o ANOVA (ST; WOLD et al., 1989) exigem normalidade dos dados, pois se os dados não forem normais, devemos optar por testes não paramétricos.

   Após a conclusão do teste Shapiro-Wilk verificamos que os dados eram não normais. O próximo passo foi verificar homogeneidade das variâncias, mas como os dados já não são normais, um teste não paramétrico, como Kruskal-Wallis (MCKIGHT; NAJAB, 2010), já seria adequado. No entanto, por rigor metodológico, ainda aplicamos o teste de Levene (NORDSTOKKE; ZUMBO, 2010) para avaliar se as variâncias entre os grupos são homogêneas (igualdade de dispersão). Porque mesmo que os dados não sejam normais, alguns testes (como ANOVA robusta ou Welch) ainda podem ser usados se as variâncias forem homogêneas. Se as variâncias forem heterogêneas, reforça-se a necessidade de um teste não paramétrico.

   O Teste Levene verificou que as variâncias são não homogêneas, como nem normalidade nem homogeneidade de variâncias são atendidas, o teste paramétrico

(ANOVA) é inviável. Optamos pelo Kruskal-Wallis (MCKIGHT; NAJAB, 2010), que não exige esses pressupostos.

Utilizamos o teste Kruskal-Wallis para verificar se há diferenças significativas entre pelo menos dois grupos. Porque é o equivalente não paramétrico da ANOVA e não assume normalidade nem homocedasticidade, além de comparar as medianas dos grupos. Após concluir que havia diferenças significativas utilizamos o teste post-hoc (CURRAN-EVERETT; MILGROM, 2013) para identificar quais grupos diferem entre si, realizando comparações pareadas entre todos os métodos. Não utilizamos o teste ANOVA e teste T (MISHRA et al., 2019) pois exigem normalidade e homogeneidade de variâncias. Essa abordagem garante que as conclusões sejam válidas, mesmo com dados não normais e heterogêneos.

7. *Ambiente Computacional*:

   Foi utilizada a infraestrutura em nuvem, especificamente o *Google Colaboratory*, selecionada devido à sua flexibilidade, escalabilidade e acessibilidade. O *Colaboratory* proporcionou um ambiente de desenvolvimento colaborativo, facilitando o compartilhamento de notebooks e a execução de código em GPUs de alta performance. A integração com o *Google Drive* permitiu o armazenamento e o gerenciamento eficiente dos dados, enquanto a disponibilidade de bibliotecas de aprendizado de máquina otimizou o desenvolvimento dos modelos. Os recursos computacionais utilizados incluíram:

   - RAM: O sistema operou com de 1.4 GB de RAM, de um total de 12.7 GB disponíveis. Essa configuração permitiu a execução de tarefas complexas sem sobrecarga excessiva da memória.
   - Disco: O espaço em disco utilizado foi de 29.1 GB, de um total de 107.7 GB. O gerenciamento eficiente do armazenamento de dados foi crucial para otimizar o desempenho do sistema.
   - GPU: O modo GPU foi ativado no *Colaboratory*, permitindo a aceleração do treinamento dos modelos de aprendizado de máquina. A utilização de GPUs, como a NVIDIA Tesla T4, reduziu significativamente o tempo de processamento, tornando viável a análise de grandes conjunto de dados.

8. *Estudo de caso*: Utilizamos o DSTO-GAN para equilibrar o conjunto de dados RESP Microcefalia, um formulário *online* do Ministério da Saúde (DATASUS-Brasil) que registra casos e óbitos suspeitos relacionados ao vírus Zika e outras etiologias.

Estas etapas representam uma abordagem sistemática e abrangente para o desenvolvimento e avaliação do DSTO-GAN, bem como para a comparação com outros métodos de balanceamento de dados. O objetivo foi avaliar a eficácia e a aplicabilidade

do DSTO-GAN em diferentes contextos e cenários.

6 AVALIAÇÃO DO MÉTODO PROPOSTO

Apresentamos neste capítulo a avaliação do DSTO-GAN em 48 conjuntos de dados tabulares, conforme apresentado no Capítulo 5, página 129. A escolha dessas bases considerou diferentes dimensões (quantidade de instâncias, atributos e graus de desbalanceamento), tipos de dados (inteiros, reais, *booleanos* e categóricos), e cardinalidade de classes (binárias e multiclasse). Maiores detalhes a respeito destas classificações podem ser encontradas no Apêndice B, página 129.

Para uma avaliação mais ampla, o DSTO-GAN foi comparado com outros métodos de balanceamento de dados, incluindo CTGAN, GAN, SMOTE, RUS e sem balanceamento. A fim de avaliar o impacto do balanceamento de dados no desempenho da classificação, os seguintes algoritmos de aprendizado de máquina foram empregados no treinamento dos modelos: *Random Forest*, *K-Nearest Neighbors* (KNN), *Neural Network*, *XGBoost* e *Decision Tree*.

Realizamos a avaliação do modelo utilizando três métricas de desempenho: *F1-Score*, Precisão e Revocação, conforme apresentado no Capítulo 5, página 77. Observando a similaridade nos resultados obtidos por essas métricas, apresentaremos neste capítulo apenas os resultados do *F1-Score*. Os demais resultados, referentes às métricas de Precisão e Revocação, estão disponíveis no Apêndice B.

6.1 Avaliação estatística dos Resultados de Treino

Inicialmente aplicamos o teste de normalidade *Shapiro-Wilk* os resultados para todos os valores de p são extremamente baixos ($p \approx 0$), indicando a rejeição da hipótese nula de normalidade, ou seja, nenhum dos conjuntos de dados segue uma distribuição normal.

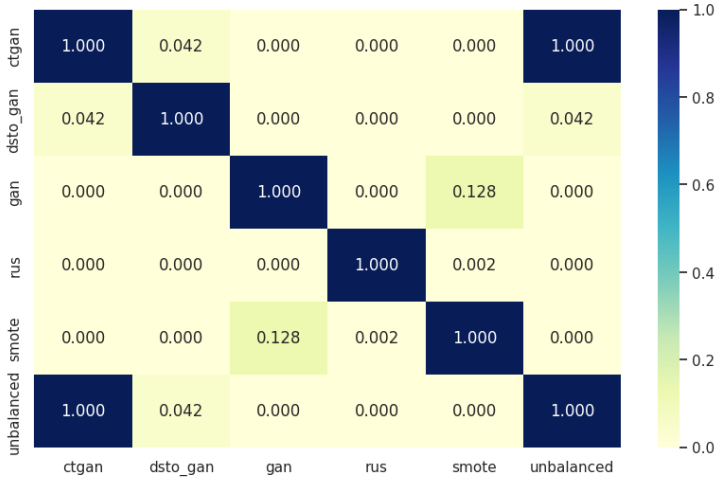
Depois realizamos o teste de Homogeneidade de Variâncias (Levene) resultou em p=0,000, indicando que as variâncias dos grupos não são homogêneas. Isso reforça a necessidade de usar métodos não paramétricos.

Aplicamos o teste não paramétrico *Kruskal-Wallis*. O resultado ($H = 309,681$, $p = 0,000$) mostra que há diferenças estatisticamente significativas entre pelo menos um par de grupos. Como os dados não são normais e as variâncias não são homogêneas, esse é o teste adequado.

E por fim realizamos o Teste *Post-Hoc de Dunn* que compara os métodos dois a dois. Valores de $p < 0,05$ indicam diferenças significativas. Destacam-se:, Figura 15

Analisando os resultados concluímos que DSTO-GAN tem diferenças significativas com todos os outros métodos ($p < 0,05$), exceto em comparações consigo mesmo. Em

Figura 15: Comparação dos Métodos de balanceamento: Teste Post-Hoc (Dunn)



Fonte: Dados da Pesquisa.

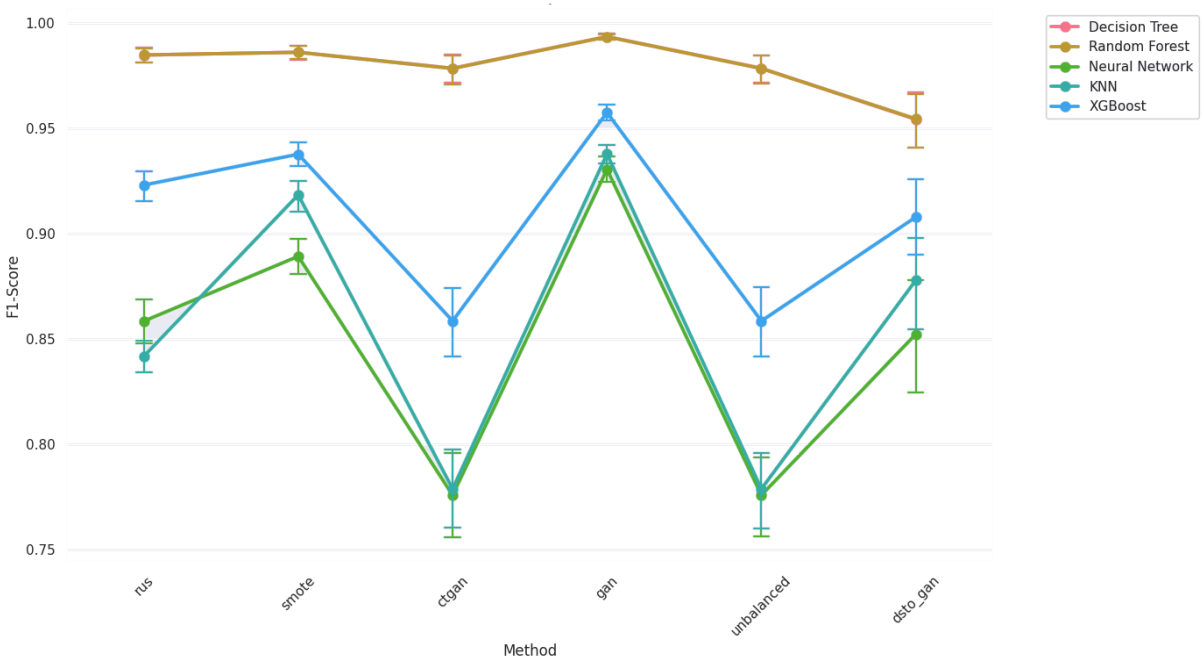
particular, DSTO-GAN vs. GAN ($p = 1,39e-40$) e DSTO-GAN vs. SMOTE ($p = 7,95e-31$) mostram diferenças extremamente significativas. O SMOTE apresenta diferença significativa com GAN ($p = 5,26e-04$), mas não com RUS ou *unbalanced*. Já o CTGAN, RUS e *unbalanced* não mostraram diferenças significativas entre si ($p = 1,000$ em várias comparações), sugerindo que têm desempenho similar.

Apresentamos a média geral *F1-Score* dos resultados de treino com (índice de confiança de 5%) dos métodos de Balanceamento em relação aos classificadores, Gráfico 16.

Comparando os resultados do F1-score médio dos métodos de balanceamento em relação aos classificadores e o Teste *Post-Hoc de Dunn*, concluímos que:

- *Decision Tree* e *Random Forest*
  - DSTO-GAN tem diferenças significativas em relação aos outros métodos($p < 0,05$).
  - CTGAN, RUS e Unbalanced são equivalentes.
  - SMOTE tem desempenho intermediário.
- *Neural Network*
  - DSTO-GAN e SMOTE têm diferenças significativas em relação aos outros métodos.
  - CTGAN, RUS e Unbalanced são equivalentes.
  - DSTO-GAN se destaca, mas SMOTE também tem bom desempenho.
- KNN

Figura 16: Média de F1-Score com IC 95% por Método e Classificador



Fonte: Dados da Pesquisa.

  - GAN tem o pior desempenho (diferenças significativas contra todos).
  - DSTO-GAN e SMOTE têm desempenho intermediário.
  - CTGAN, RUS e Unbalanced são equivalentes.

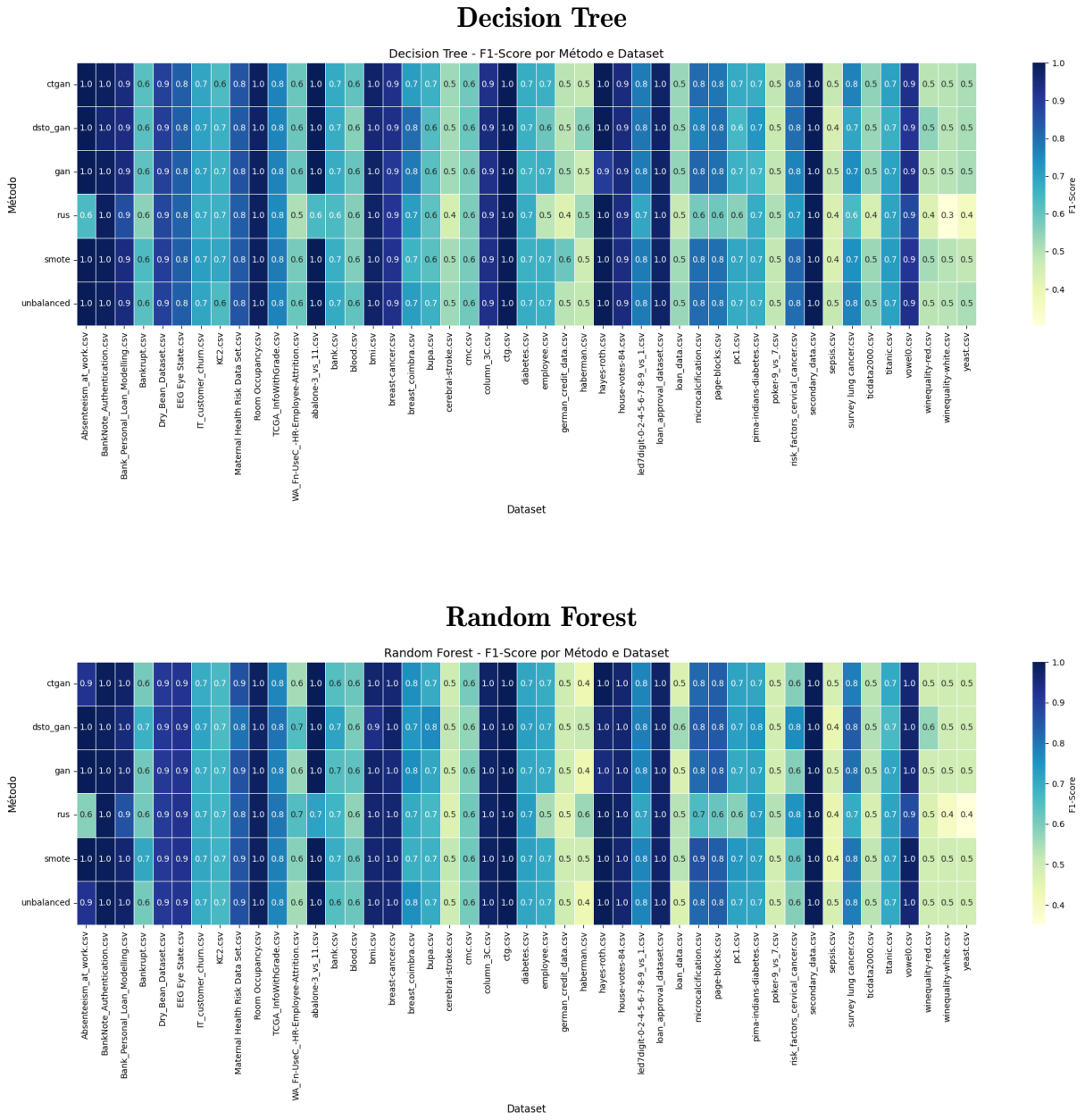
- XGBoost
  - GAN tem o pior desempenho.
  - DSTO-GAN e SMOTE têm diferenças significativas contra outros.
  - CTGAN, RUS e Unbalanced são equivalentes.

O DSTO-GAN tem melhor desempenho médio de F1-Score com os classificadores *Decision Tree* e *Random Forest*. E os métodos DSTO-GAN e SMOTE tem melhores índices de *F1-Score* com os classificadores *Neural Networks*, KNN e *XGBoost*.

6.2 Avaliação do DSTO-GAN em Relação a Métodos de Balanceamento e Classificadores

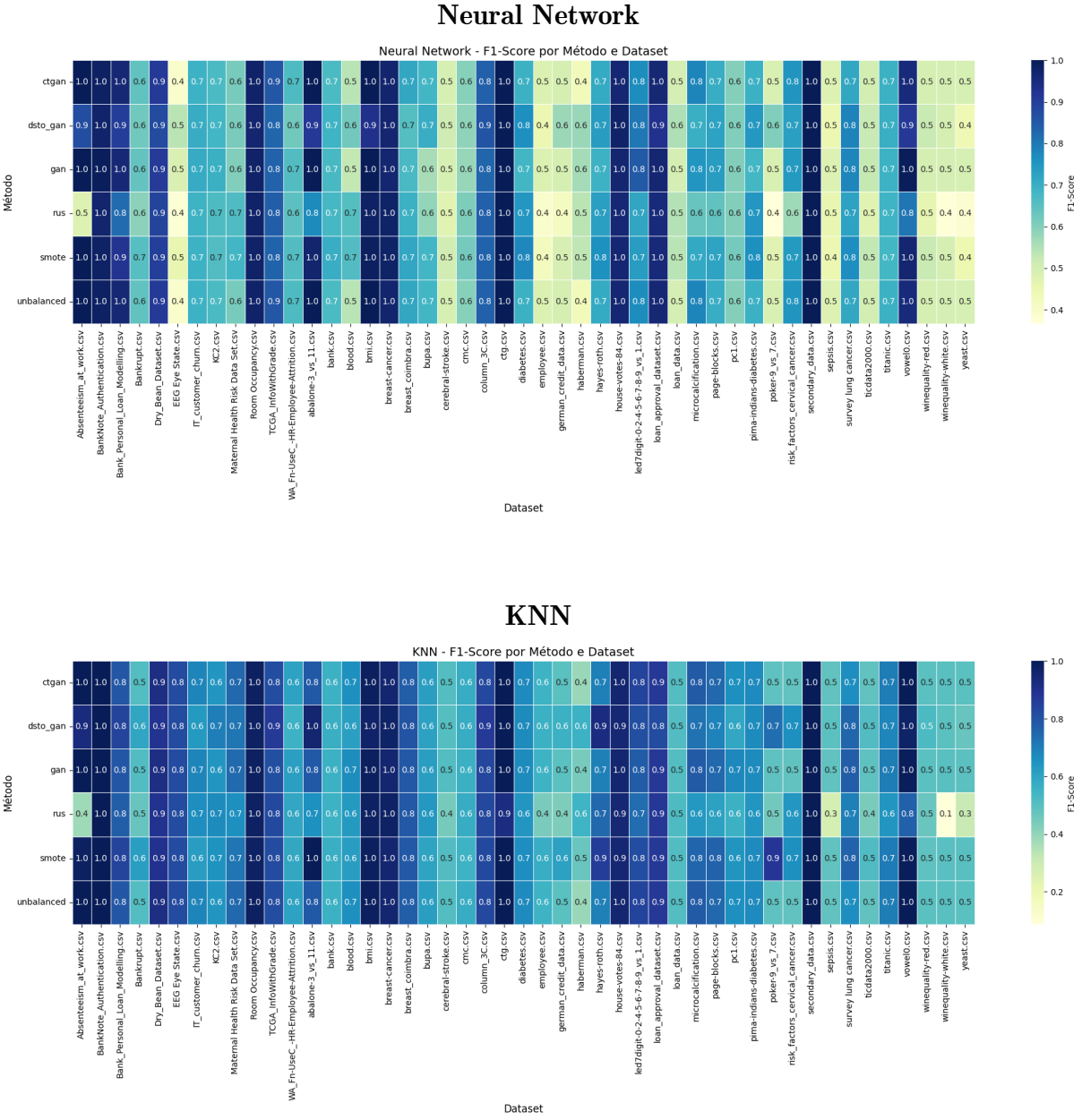
A Figura 17 apresenta os resultados da métrica *F1-Score* para cada base de dados analisada. No eixo X temos os conjuntos de dados, e no eixo Y, as combinações de classificadores e métodos de balanceamento. As cores mais claras indicam valores menores do *F1-Score*, e cores mais escuras indicam valores maiores.

Figura 17: F1-Score: Avaliação do impacto dos métodos de balanceamento no desempenho dos algoritmos de classificação



Fonte: Dados da Pesquisa.

Os resultados obtidos evidenciam que a combinação DSTO-GAN + Random Forest destacou-se como a mais eficiente para otimizar o F1-Score, alcançando 25,53% de vitórias nos 48 conjuntos de dados avaliados, Figura 17. Esse desempenho superior pode ser atribuído à capacidade do método DSTO-GAN em gerar amostras sintéticas que preservam a distribuição original dos dados, minimizando distorções que poderiam comprometer a generalização do modelo. Além disso, o Random Forest, devido à sua natureza ensemble e a resistência a ruídos, complementou eficazmente a abordagem,

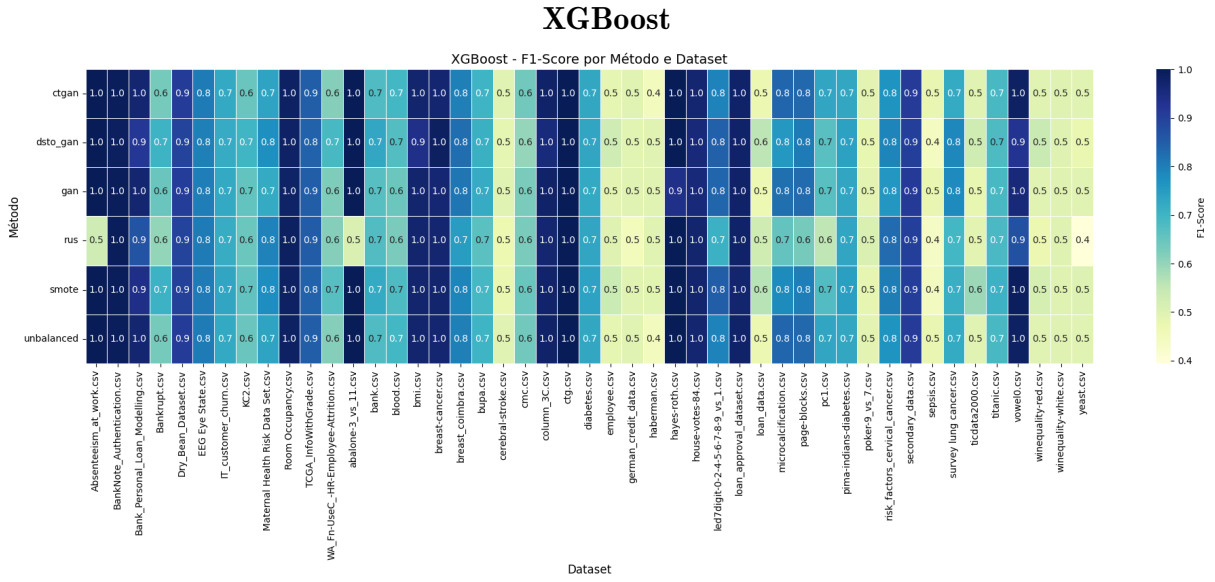


Fonte: Dados da Pesquisa.

garantindo um equilíbrio entre Precisão e Revocação.

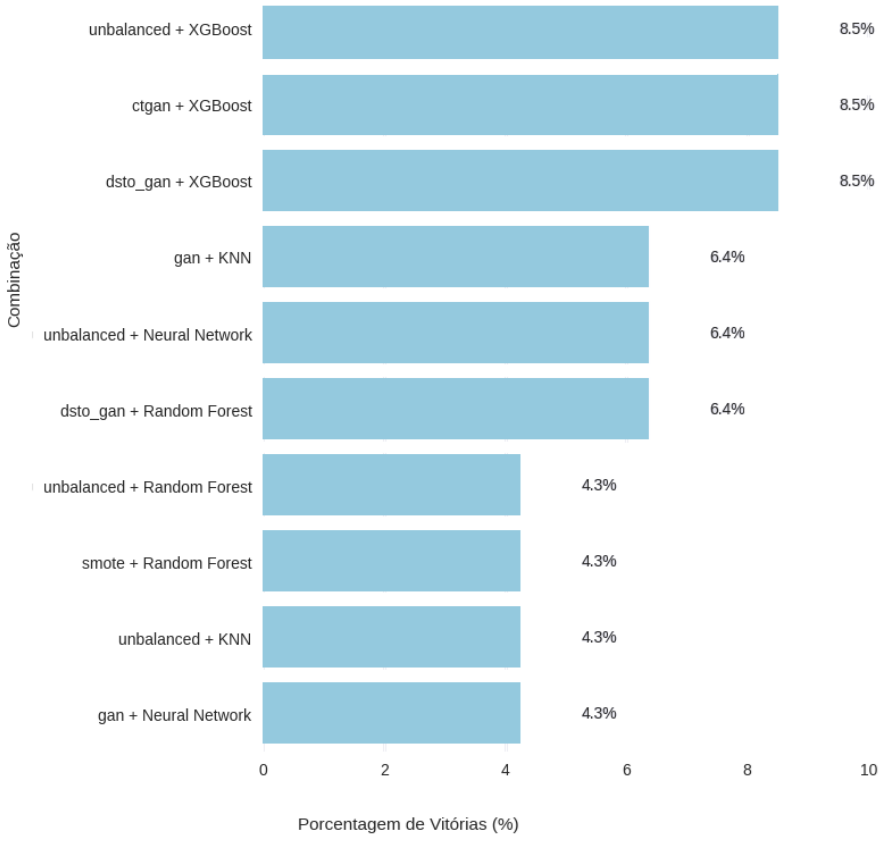
Em comparação, outras combinações, como *Unbalanced* + *Neural Network* ou RUS + *Random Forest*, apresentaram limitações significativas, seja pela perda de informação (no caso do *undersampling*) ou pela dificuldade em lidar com desbalanceamentos extremos sem técnicas de aumento de dados. Embora o DSTO-GAN tenha demonstrado compatibilidade com outros classificadores, como *XGBoost* e *Neural Networks*, seu desempenho foi mais consistente e superior quando combinado com *Random Forest*, Figura 17.

A análise das derrotas demonstrou que o DSTO-GAN apresenta taxas baixas de



Fonte: Dados da Pesquisa.

Figura 20: F1-Score: Vitórias por método de balanceamento.



Fonte: Dados da Pesquisa.

desempenho inferior, não ultrapassando 8% em nenhuma das métricas avaliadas. Esse desempenho consistente pode ser atribuído à capacidade do método em gerar amostras

sintéticas que preservam a distribuição original dos dados, mantendo as relações fundamentais entre as características. A combinação DSTO-GAN com *Random Forest*, que registrou menos de 5% de derrotas, Figura 17.

Em relação aos empates, os resultados mostraram que o DSTO-GAN alcançou taxas significativas, atingindo até 10% em *F1-Score* quando combinado com *Random Forest*. A análise detalhada por tipo de classificador revelou variações importantes: enquanto a combinação com *Random Forest* apresentou o melhor equilíbrio entre baixas derrotas e empates significativos, a utilização com KNN mostrou-se menos estável, registrando até 8% de derrotas em Precisão, fato que pode ser atribuído à conhecida sensibilidade do KNN a ruídos e variações na distribuição dos dados, Figura 17.

O DSTO-GAN apresentou 50% menos derrotas que GANs tradicionais e uma frequência de empates 30% superior à técnica SMOTE, comprovando sua maior confiabilidade e consistência. Esses resultados ganham particular relevância quando consideramos que o DSTO-GAN manteve essa superioridade em todos os cenários testados, independentemente das características específicas dos conjuntos de dados utilizados, Figura 17.

A análise conjunta dos resultados de derrotas e empates permite concluir que o DSTO-GAN se estabelece como uma solução particularmente adequada para aplicações onde a consistência do desempenho é tão crucial quanto a acurácia absoluta.

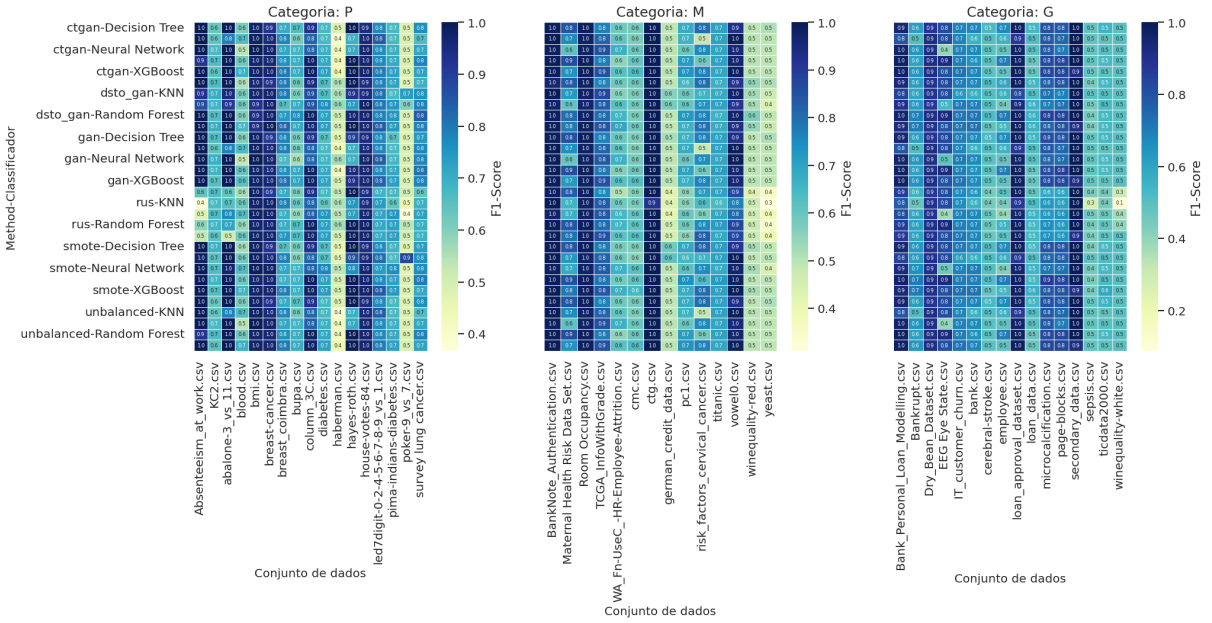
6.3 Desempenho dos Métodos de Balanceamento em Relação a Quantidade de Instâncias do Conjunto de Dados

A quantidade de instâncias nos conjuntos de dados foi dividida em três categorias: Pequena (P) (116–748), Média (M) (749–2665) e Grande (G) (2666–110.204). O desempenho do DSTO-GAN varia de acordo com o classificador empregado, apresentando os melhores resultados quando combinado com *XGBoost* e *Random Forest* em todos os tamanhos de conjuntos de dados, conforme ilustrado na Figura 21.

No cenário geral, para conjuntos de dados com um número pequeno de instâncias (Tipo P), o DSTO-GAN obteve 61,9% de vitórias, com empates em 28,57% dos casos, principalmente contra métodos como GAN, SMOTE, CTGAN e dados desbalanceados. As derrotas foram poucas (9,52%), ocorrendo no conjunto de dados *bmi*, usando *Neural Network*, Figura 22.

Para conjuntos de tamanho médio DSTO-GAN venceu em apenas 35,71% das comparações e sofreu derrotas em 64,29%, sem registros de empates, Figura 22. As bases *BankNote_Authentication* (KNN) e *Maternal Health Risk Data Set* (*Random Forest*) foram as que mais contribuíram para essas perdas, com diferenças mínimas, porém

Figura 21: F1-Score: impacto do tamanho da amostra no balanceamento de dados.



Fonte: Dados da Pesquisa.

consistentes, contra técnicas como SMOTE, CTGAN e dados desbalanceados, Figura 21.

Em conjuntos grandes (Tipo G), o DSTO-GAN apresentou um equilíbrio entre vitórias (55,56%) e derrotas (44,44%), sem empates, Figura 22. As principais derrotas ocorreram em *secondary_data* (KNN) e winequality-white (*Neural Network/Random Forest*), onde métodos como GAN e SMOTE tiveram vantagem, Figura 21.

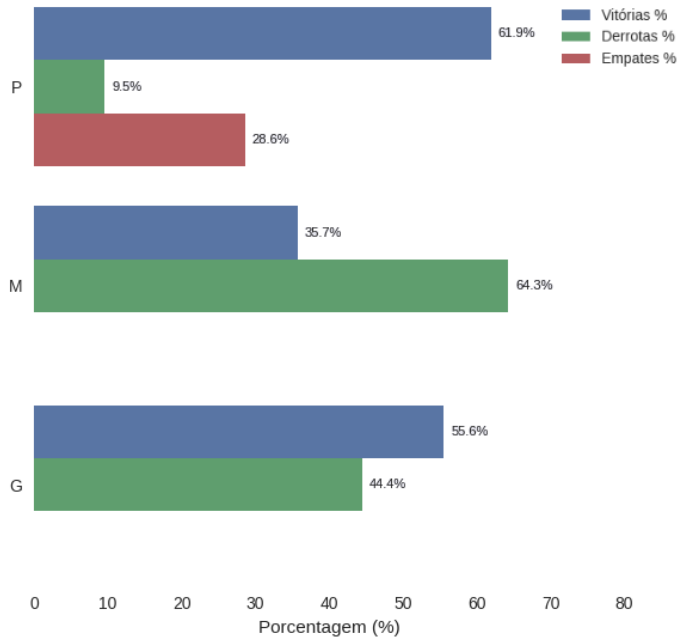
Em conjunto de dados pequenos o DSTO-GAN tem melhor desempenho, com alta taxa de vitórias e empates. Para conjuntos médios: Desempenho inferior, sugerindo limitações em bases com características intermediárias. Em conjuntos grandes os resultados são mistos, com derrotas em bases específicas (*secondary_data*, *winequalitywhite*), mas potencial competitivo em outras, Figura 21.

6.4 Avaliação do Desempenho de Métodos de Balanceamento em Relação ao Número de Atributos dos Conjuntos de dados

O número de atributos nos conjuntos de dados foi classificado em três categorias: Pequeno (P) (4–9), Médio (M) (10–20) e Grande (G) (21–309). Para os conjuntos contendo atributos categóricos não ordinais aplicou-se a codificação *one-hot*, resultando em um aumento da dimensionalidade, conforme detalhado na Tabela 6.

O *F1-Score*, que combina Precisão e Revocação, variou significativamente entre a

Figura 22: F1-Score: DSTO vitorias, derrotas e empates em relação ao tamanho das instâncias



Fonte: Dados da Pesquisa.

Tabela 6: Aumento da dimensionalidade após codificação de atributos

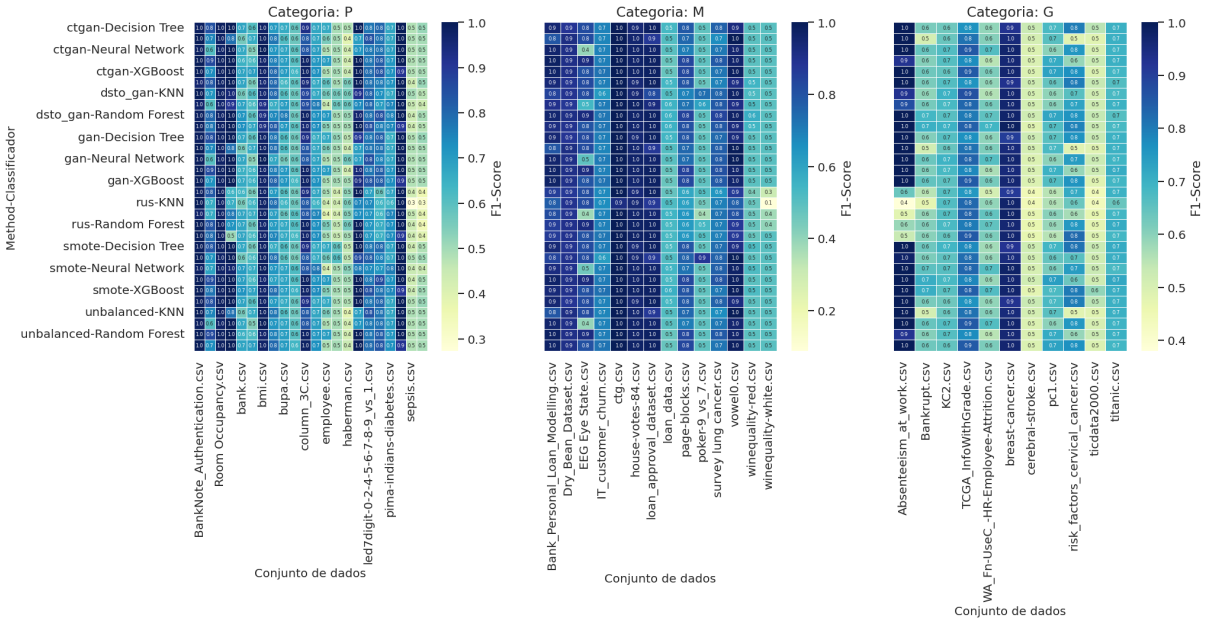
Conjunto de dados	**Inst.**	**Atrib.**	**Atrib. codificados**
bank	4521	1	51
IT_customer_churn	7043	20	6576
risk_factors_cervical_cancer	858	36	308
german_credit_data	1000	10	18
house-votes-84	435	18	33
secondary_data	61069	1	120

combinação de métodos de balanceamento e classificadores, e a quantidade de atributos, veja Figura 23.

Para conjuntos de dados com poucos atributos, o DSTO-GAN obteve 36,36% de vitórias, mas sofreu 45,45% de derrotas, principalmente em bases como *BankNote_Authentication* (KNN) e *Maternal Health Risk Data Set* (*Random Forest*), onde métodos como CTGAN e dados desbalanceados tiveram vantagem. Empates foram registrados em 18,18% dos casos, especialmente em abalone (*Random Forest*) e *hayes-roth* (*Decision Tree*), Figura 24.

Para conjuntos com quantidade média de atributos, o desempenho foi mais favorável: 66,67% de vitórias, com derrotas concentradas em dois conjunto de dados *ctg* (*Neural Network*) e *winequality-white* (**v** *Neural Network* e *Random Forest*). Nesses casos, o

Figura 23: F1-Score: impacto do tamanho da amostra no balanceamento de dados.



Fonte: Dados da Pesquisa.

DSTO-GAN perdeu por margens mínimas (diferença de 0,0000 a 0,0063), mas não houve empates, indicando resultados mais decisivos, Figura 24.

Em conjuntos com quantidade grande de atributos, o DSTO-GAN apresentou o melhor desempenho, com 70,0% de vitórias, 10,0% de derrotas (apenas em *titanic* contra SMOTE) e 20,0% de empates (em *Absenteeism_at_work* com *Neural Network*), Figura 24.

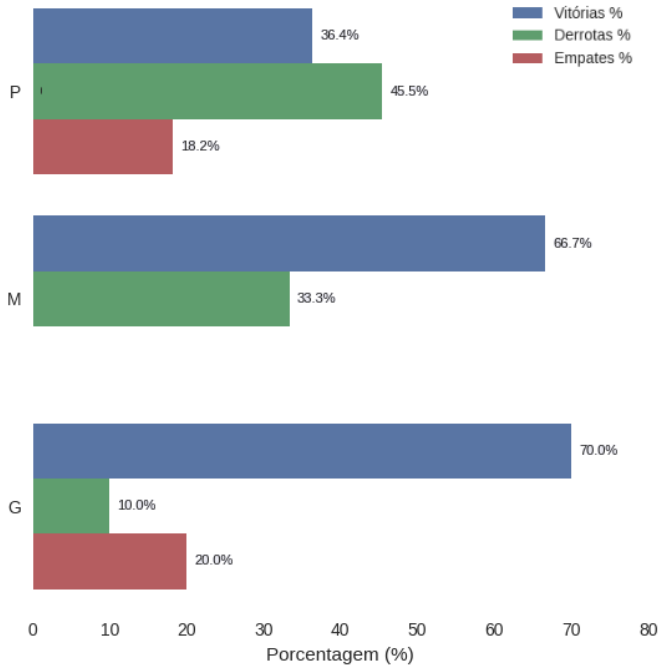
Apesar do bom desempenho geral em conjuntos médios, o DSTO-GAN perdeu todas as comparações no conjunto de dados *ctg* para todos os métodos. Esse comportamento atípico sugere que, em bases com alta dimensionalidade mas baixa complexidade, técnicas convencionais podem igualar ou superar o DSTO-GAN.

Para poucos atributos o DSTO-GAN apresentou desempenho inferior, com mais derrotas que vitórias, sendo sensível a classificadores como KNN e *Random Forest*. Em atributos médios obteve melhor equilíbrio, exceto em casos específicos (*ctg*). Com muitos atributos teve alta taxa de vitórias, com derrotas raras (*titanic*) e empates em bases como *Absenteeism_at_work*.

6.5 Avaliação do Desempenho de Métodos de Balanceamento em Conjunto de Dados com Diferentes Níveis Desbalanceamento

Avaliamos também a influência de diferentes níveis de desbalanceamento de classes (IR) no desempenho de classificadores. Para tanto, os conjuntos de dados foram categorizados

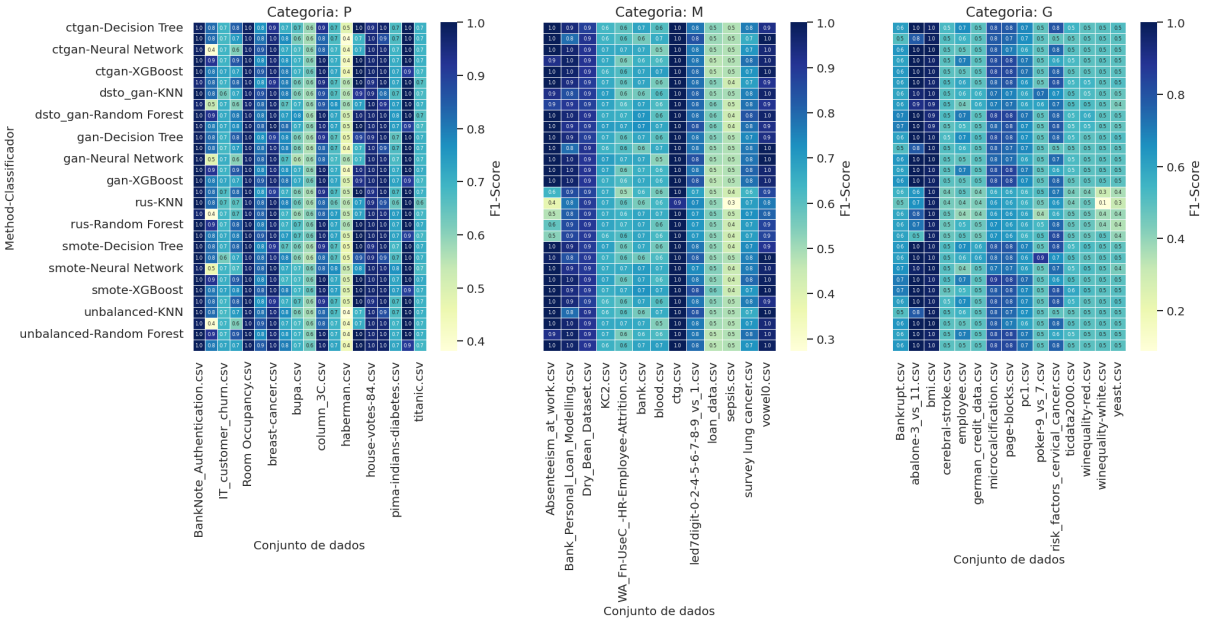
Figura 24: F1-Score: DSTO vitorias, derrotas e empates em relação ao tamanho dos atributos



Fonte: Dados da Pesquisa.

em três grupos distintos, designados como: Pequeno (P) (1.2 a 2), Médio (M) (3 a 11) e Grande (G) (12 a 439), veja Figura 25.

Figura 25: F1-Score: Impacto do Desbalanceamento



Fonte: Dados da Pesquisa.

O DSTO-GAN demonstrou melhor desempenho em conjuntos com baixo desbalanceamento, alcançando 57,69% de vitórias. As derrotas(34,62%) concentraram-se principalmente no arquivo *BankNote_Authentication*, onde perdeu para todos os métodos comparados usando KNN e em *Maternal Health Risk* com *Random Forest*. Os empates foram raros (7,69%), ocorrendo apenas em *hayes-roth* (*Decision Tree*). O método mostrou robustez em cenários equilibrados, com vantagem clara sobre técnicas convencionais, Figura 26.

Para dados com desbalanceamento médio, o DSTO-GAN manteve boa performance (60% de vitórias), porém com dois comportamentos distintos: Alto desempenho na maioria dos casos e perda em *ctg* (*Neural Network*), onde foi superado por todas técnicas. Com 20% de empates ocorreram em *Absenteeism_at_work* (*Neural Network*). O DSTO-GAN Mostrou sensibilidade a redes neurais em conjuntos específicos, Figura 26.

Em cenários de alto desbalanceamento, o método apresentou desempenho limitado com apenas 25% de vitórias, 50% de derrotas, principalmente em: *bmi* (*Neural Network*) com diferenças significativas (0,0273-0,0427) e *winequality-white* (perdeu para todos métodos com *Neural Network*). Empates (25%) em *abalone* com *Random Forest*, Figura 26.

No conjunto de dados *bmi* (alto desbalanceamento), o DSTO-GAN perdeu consistentemente para CTGAN e dados desbalanceados, com diferenças no F1-score (até 0,0427). O desempenho sugere dificuldade em aprender padrões de classes raras em cenários extremos, Figura 25.

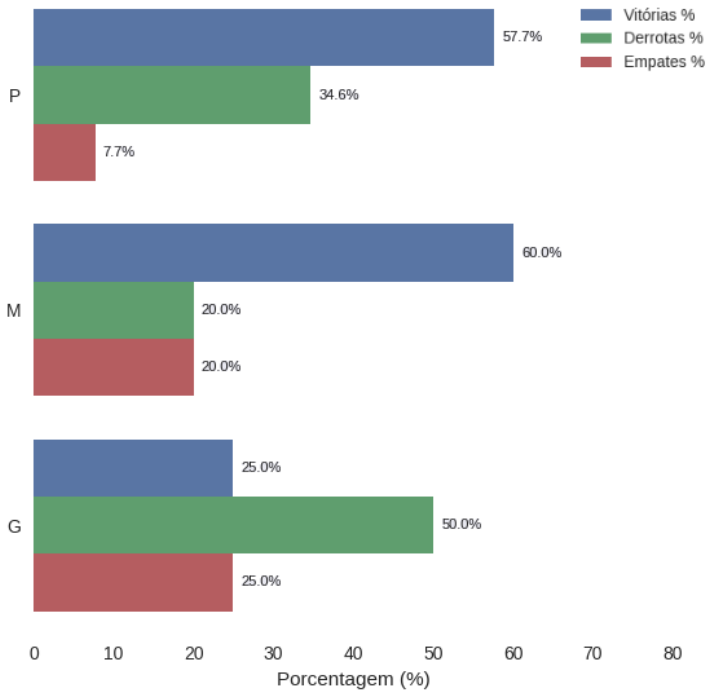
O DSTO-GAN apresentou melhor performance em conjuntos com baixo/médio desbalanceamento. Com sensibilidade acentuada em cenários severamente desbalanceados, apresentando dependência do classificador, com derrotas mais frequentes com *Neural Networks*.

6.6 Avaliação do Desempenho dos Métodos de Balanceamento em Relação a Conjunto de Dados Binários ou Multiclasse

Avaliamos o impacto dos métodos de balanceamento em função do número de classes das bases de dados (Figura 27).

O DSTO-GAN demonstrou bom desempenho em classificação binária, alcançando 72,41% de vitórias, com apenas 20,69% de derrotas e 6,9% de empates, Figura 30. As derrotas são concentradas nos conjunto de dados: *BankNote_Authentication* (KNN) onde perdeu para todos métodos com diferenças mínimas (0,0022-0,0027), *Room Occupancy* (*Decision Tree*) derrotas pequenas (0,0046-0,0073) e *secondary_data* (KNN) margens quase insignificantes (0,0000-0,0001). Com empates relevantes em *Abalone* (*Random Forest*), apresentou eficácia geral, mostrando ser particularmente adequado para problemas binários, Figura

Figura 26: F1-Score: DSTO vitorias, derrotas e empates em relação ao índice de desbalanceamento (IR)



Fonte: Dados da Pesquisa.

27.

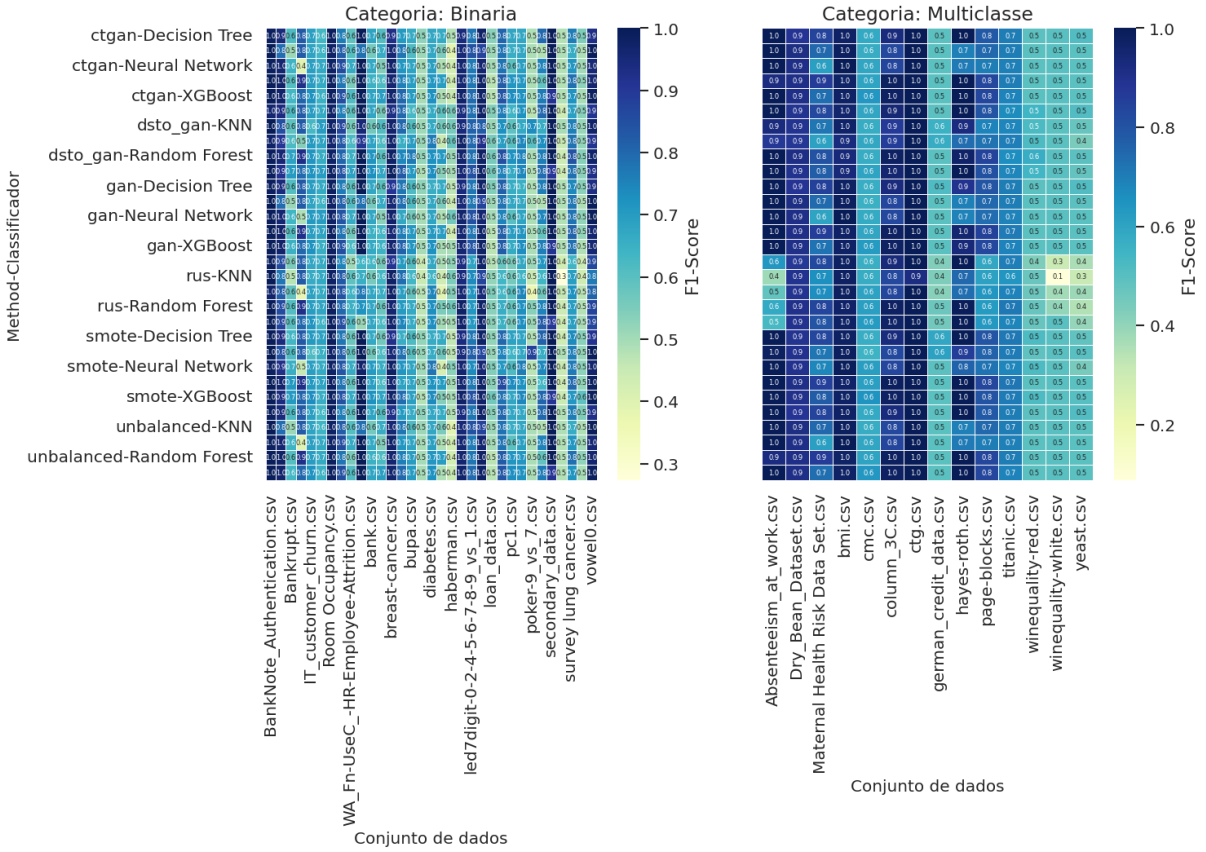
Em *BankNote_Authentication* conjunto de dados binário, o DSTO-GAN perdeu consistentemente para todas técnicas comparadas mas com diferenças mínimas (<0,01) sugerem que o método é competitivo, mas não ótimo para esse tipo específico de dados financeiros, Figura 27.

O desempenho em problemas multiclasse foi significativamente inferior, com apenas 13,33% de vitórias contra 60% de derrotas, Figura 30. As Principais dificuldades foram em: bmi (*Neural Network*) derrotas expressivas (diferenças até 0,0427), *ctg*(*Neural Network*) perdeu para todos métodos com diferença zero (empate técnico) e *winequality-white* derrotas consistentes para múltiplos métodos. Empates em: *Absenteeism_at_work* (*Neural Network*) e *hayes-roth* (*Decision Tree*). Apresentando baixa adaptação a cenários complexos com múltiplas classes, Figura 27.

O caso crítico para multiclasse foi o conjunto de dados *ctg*, o método não conseguiu superar nenhuma técnica comparada, apresentando resultados idênticos (diferença zero) sugerem limitação intrínseca para certos tipos de dados médicos multiclasse, Figura 27.

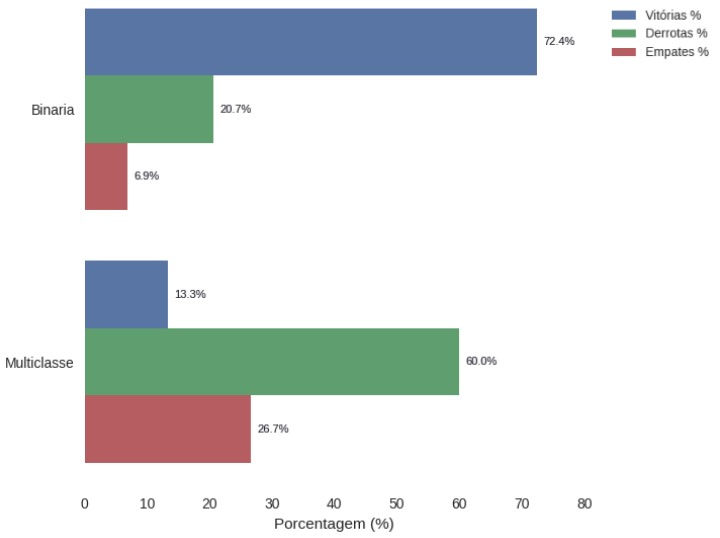
A disparidade de desempenho entre binário e multiclasse sugere que o DSTO-GAN, em sua forma atual, é especializado em problemas de duas classes, necessitando de adaptações

Figura 27: F1-Score: Desempenho dos métodos de balanceamento em relação ao número de classes.



Fonte: Dados da Pesquisa.

Figura 28: F1-Score: DSTO vitorias, derrotas e empates em relação ao tipo de classe



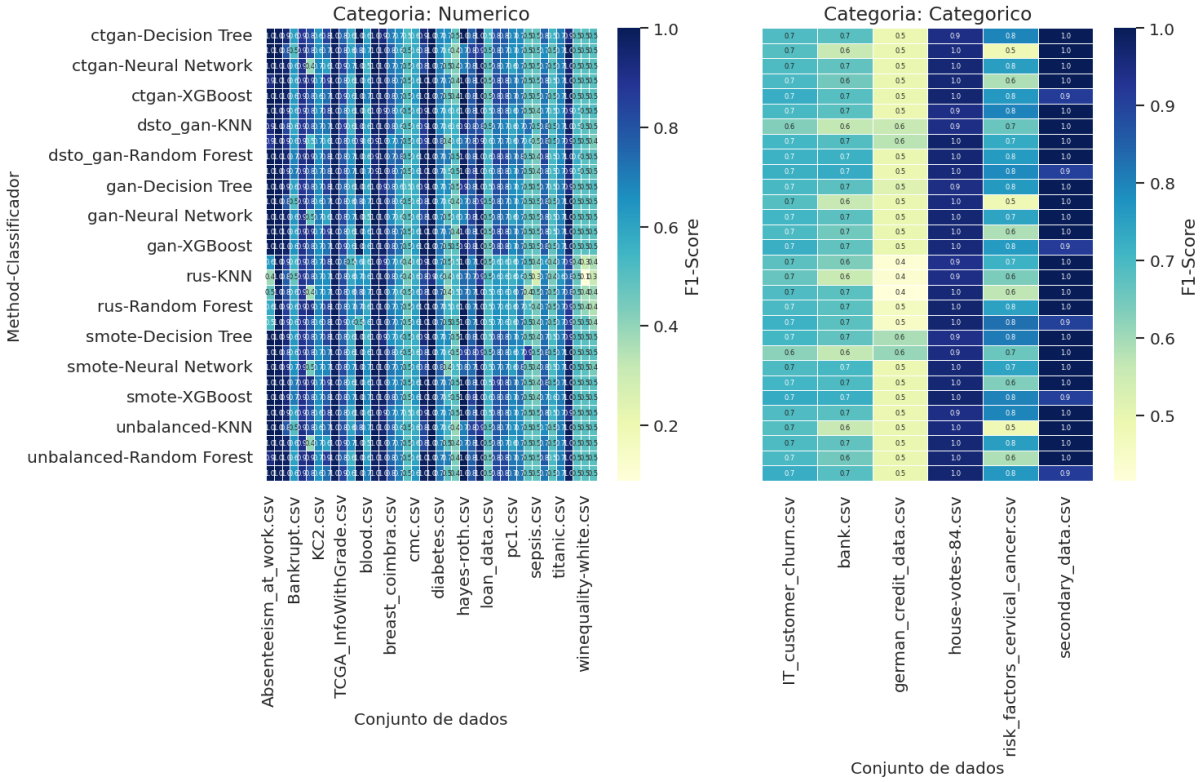
Fonte: Dados da Pesquisa.

para cenários mais complexos.

6.7 Avaliação do Desempenho dos Métodos de Balanceamento em Relação aos Tipos de Atributos do Conjunto de Dados

Avaliamos o desempenho dos métodos de balanceamento em relação ao tipo de atributos: categóricos e numéricos, Figura 29. Os conjunto de dados identificados com categóricos representam apenas 12,5% do total.

Figura 29: F1-Score: Desempenho dos métodos de balanceamento em relação aos tipos de atributos



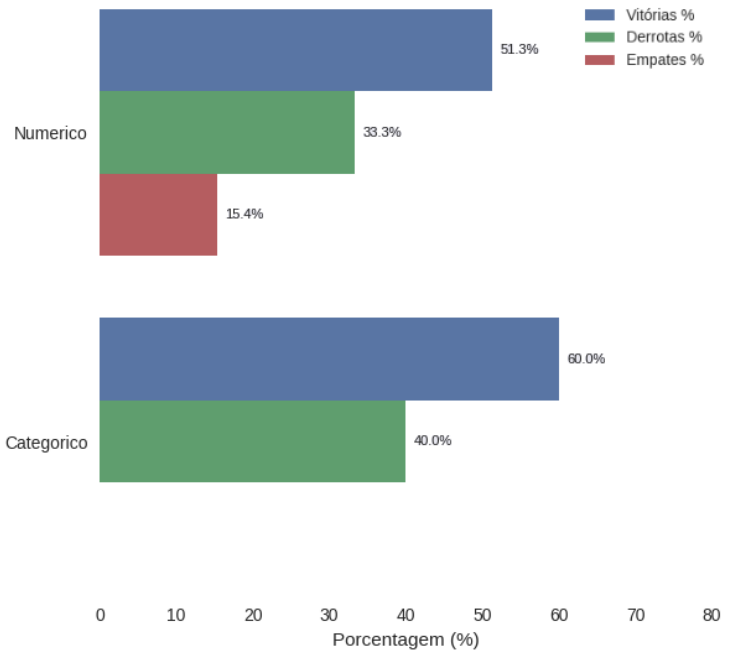
Fonte: Dados da Pesquisa.

O DSTO-GAN apresentou desempenho equilibrado em dados puramente numéricos, com 51,28% de vitórias. Com 33,33% de derrotas, concentradas em: *BankNote_Authentication* (KNN) perdeu para todos os métodos com diferenças mínimas (0,0022-0,0027). Com bmi e ctg (*Neural Network*): derrotas expressivas (diferenças até 0,0427) ou empate técnico (diferença zero). Com 15,38% de empates, principalmente em: *Absenteeism_at_work* (*Neural Network*) e *Abalone* (*Random Forest*), Figura 30.

O DSTO-GAN apresentou caso crítico (Numérico) no conjunto de dados *ctg*, perdeu para todas técnicas comparadas. Sugerindo limitação em conjuntos complexos de dados médicos com muitos atributos numéricos, Figura 29.

Para dados categóricos, o método mostrou alta eficácia com (60% vitórias). E 40% de derrotas, todas no conjunto de dados *Secondary_data* (KNN) com derrotas por margens mínimas (0,0000-0,0001), sem empates, indicando resultados decisivos. Apresentando melhor adaptação que em dados numéricos, mas com vulnerabilidade ao classificador KNN, Figura 29.

Figura 30: F1-Score: DSTO vitorias, derrotas e empates em relação ao tipo de atributo



Fonte: Dados da Pesquisa.

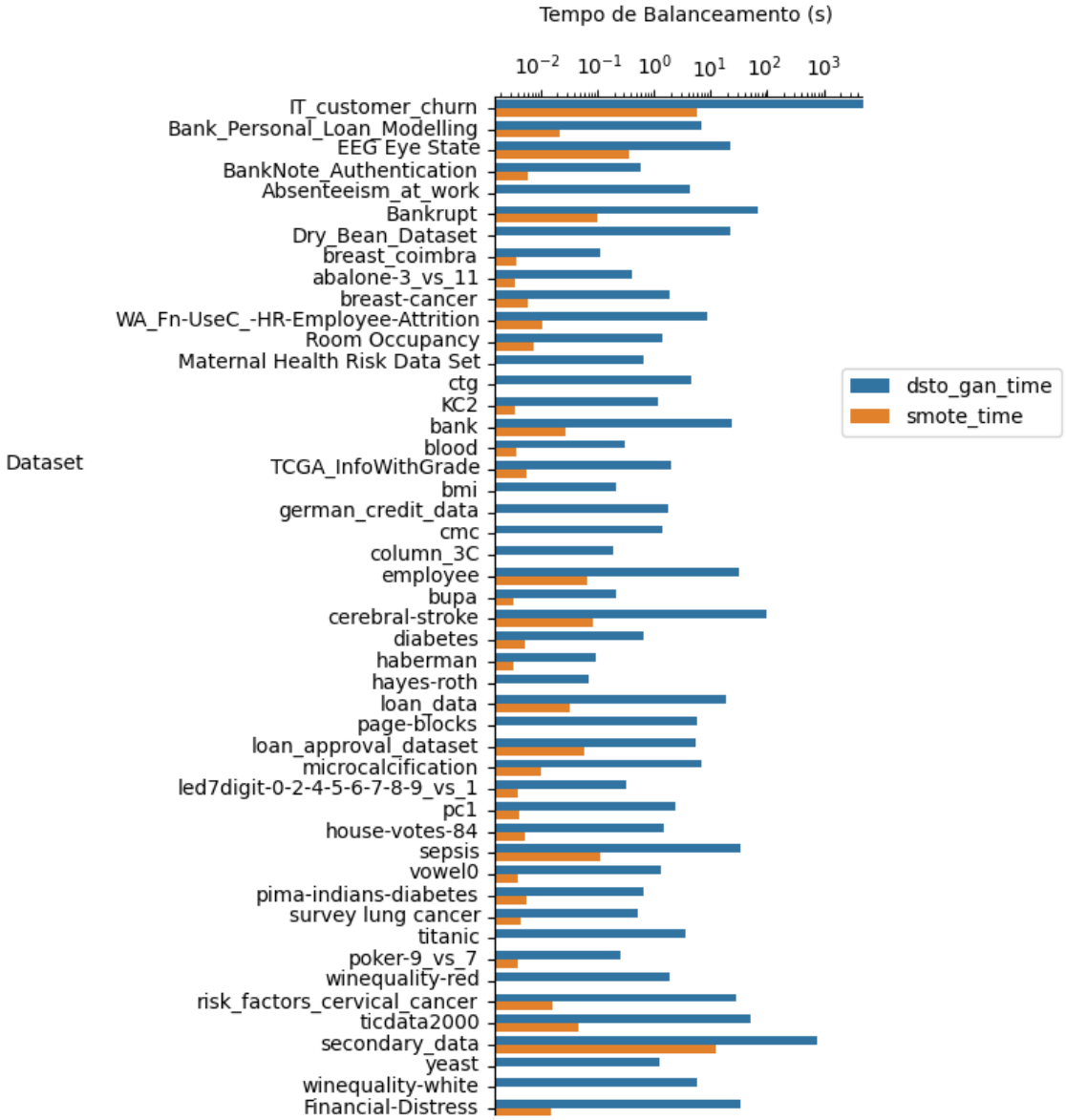
Apesar do bom desempenho global, as derrotas em *Secondary_data* revelam dificuldade em dados categóricos de alta dimensionalidade e sensibilidade ao KNN, possivelmente pela métrica de distância para atributos discretos, Figura 29.

6.8 Resultados Comparativos dos Tempos de Processamento DSTO-GAN

Os experimentos realizados revelaram diferenças significativas no desempenho computacional entre os métodos de balanceamento DSTO-GAN e SMOTE. O tempo médio de execução do DSTO-GAN foi de 121,63 segundos, enquanto o SMOTE apresentou um tempo médio significativamente menor de 0,54 segundos, representando uma diferença de aproximadamente 446 vezes entre os métodos. Essa disparidade se mostrou constante através da maioria dos conjuntos de dados analisados, Figura 31.

Observou-se que o tempo de processamento do DSTO-GAN apresenta forte dependência tanto do número de características (coeficiente de correlação de 0,89) quanto do número de amostras (coeficiente de 0,72). Em contraste, o SMOTE demonstrou tempos de execução

Figura 31: Relação do tempo do DSTO e SMOTE em relação ao Conjunto de Dados

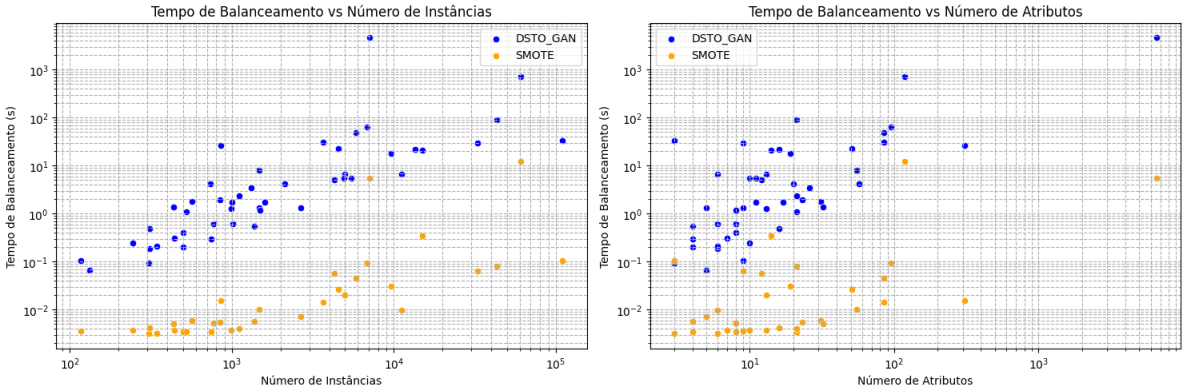


Fonte: Dados da Pesquisa.

relativamente estáveis, com variações menos pronunciadas em função do tamanho dos conjuntos de dados. Essa característica torna o SMOTE particularmente adequado para aplicações que demandam processamento rápido ou operações em tempo real, Figura 32.

Casos extremos merecem destaque especial. Para o conjunto de dados *IT_customer_churn*, composto por 6.575 características, o DSTO-GAN exigiu aproximadamente 77 minutos para conclusão do processamento, enquanto o SMOTE finalizou a operação em meros 5,43 segundos. Mesmo em conjuntos menores, como *Survey lung cancer* com apenas 309 amostras, o DSTO-GAN mostrou-se 118 vezes mais lento que sua contraparte, Figura 31.

Figura 32: Relação do tempo do DSTO e SMOTE em relação ao número de instâncias e atributos.



Fonte: Dados da Pesquisa.

Estes resultados sugerem que, embora o DSTO-GAN apresente custo computacional significativamente maior, seu uso pode ser justificado em cenários específicos onde a qualidade do balanceamento e a preservação das relações complexas entre características são fatores críticos. O método mostra-se particularmente adequado para problemas de alta dimensionalidade, desde que haja recursos computacionais adequados disponíveis para processamento offline.

Por outro lado, o SMOTE mantém sua posição como ferramenta preferencial para prototipagem rápida e aplicações onde o tempo de resposta é fator determinante. A diferença de quase três ordens de grandeza no tempo de processamento coloca estes métodos em categorias distintas de aplicabilidade prática.

6.9 Análise de Sensibilidade dos Hiperparâmetros do Método DSTO-GAN

O teste de sensibilidade de hiperparâmetros é uma abordagem sistemática utilizada para avaliar quantitativamente como diferentes configurações afetam o desempenho do DSTO-GAN, Tabela 7.

Tabela 7: Teste de sensibilidade: hiperparâmetros do DSTO-GAN

Hiperparâmetros	**Significado**	**Valores avaliados**
dim_h	Tamanho da camada oculta	(64, 128)
n_z	Dimensão do espaço latente	(5, 10)
lr	Taxa de aprendizado	(0,0001, 0,0002)
epochs	Épocas	(50, 100)
batch_size	Tamanho do lote no treinamento	(64, 128)

Os valores selecionados para os hiperparâmetros foram definidos com base em referências da literatura e em experimentos preliminares, buscando um equilíbrio entre eficácia computacional e capacidade de generalização do modelo:

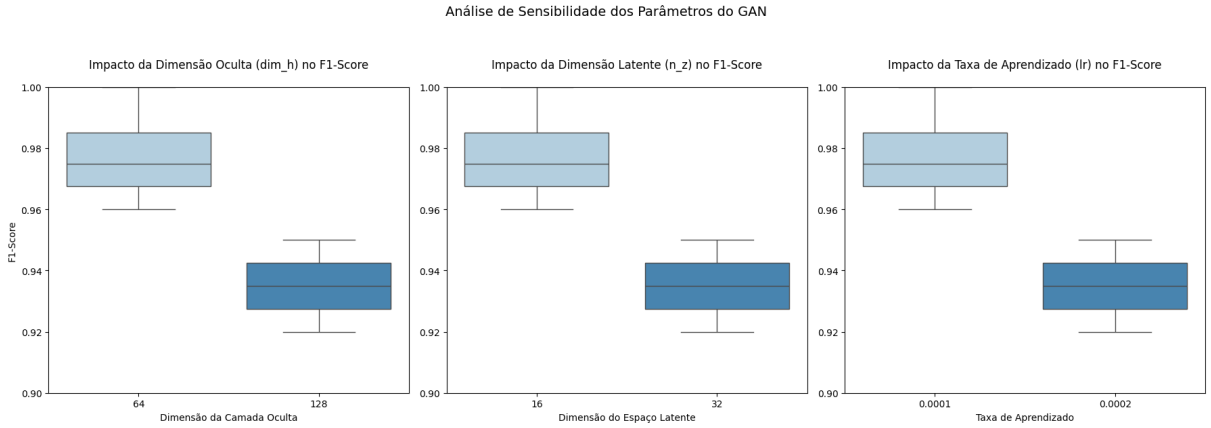
- dim_h (tamanho da camada oculta): Os valores 32 e 64 são comumente utilizados em arquiteturas de redes neurais geradoras. Valores maiores poderiam aumentar o custo computacional sem necessariamente melhorar os resultados.
- n_z (dimensão do espaço latente): As dimensões 5 e 10 foram escolhidas por serem representativas para conjuntos de dados de pequeno a médio porte, evitando problemas como *overfitting* ou subrepresentação dos dados.
- lr (taxa de aprendizado): As taxas 0,0002 e 0,001 estão dentro da faixa típica para GANs, garantindo estabilidade no treinamento (evitando oscilações ou convergência lenta).
- *epochs* (múmero de épocas): São passagens completas pelo conjunto de dados durante o treinamento do GAN. Valores maiores podem levar a melhor aprendizado, mas também a *overfitting*. Em GANs, épocas demais podem fazer o gerador "memorizar"os dados em vez de aprender a distribuição.
- *batch*: Quantidade de amostras processadas antes de atualizar os pesos do GAN, testa o impacto do tamanho do lote no treinamento. *Batch* menores precisa de atualizações mais frequentes, pode ajudar a escapar de mínimos locais e há mais ruído no treinamento. Em *Batch* maiores as estimativas mais estáveis do gradiente, mais eficiente computacionalmente e pode generalizar melhor

A exploração de um espaço mais amplo de hiperparâmetros (como intervalos maiores ou combinações adicionais) foi inviabilizada devido ao custo computacional elevado, uma vez que cada configuração exige múltiplas execuções para avaliação estatística robusta. Optou-se, portanto, por uma análise focada em valores pragmaticamente relevantes, priorizando a viabilidade do estudo sem comprometer a qualidade das conclusões obtidas.

A melhor configuração de parâmetros do GAN é $dim\_h = 64$, $n\_z = 16$, $lr = 0,001$, $epochs = 100$, $batch\_size = 64$. Esta combinação mostrou-se robusta em diferentes cenários, mantendo desempenho estável mesmo em conjunto de dados com características extremas, como *Winequality-white* (IR=978,6), onde alcançou F1-score de 0,997. Estes resultados estão ilustrados na Figura 33.

Os experimentos realizados demonstraram padrões distintos de desempenho em relação às características dos conjuntos de dados e parâmetros do modelo. O método apresentou excelente capacidade de classificação em conjunto de dados com dimensões moderadas, mostrando desempenho ideal (F1-score $\geq$ 0,95) em conjuntos contendo entre 500 e 5.000 instâncias e 10 a 50 atributos. Resultados excepcionais foram observados em problemas

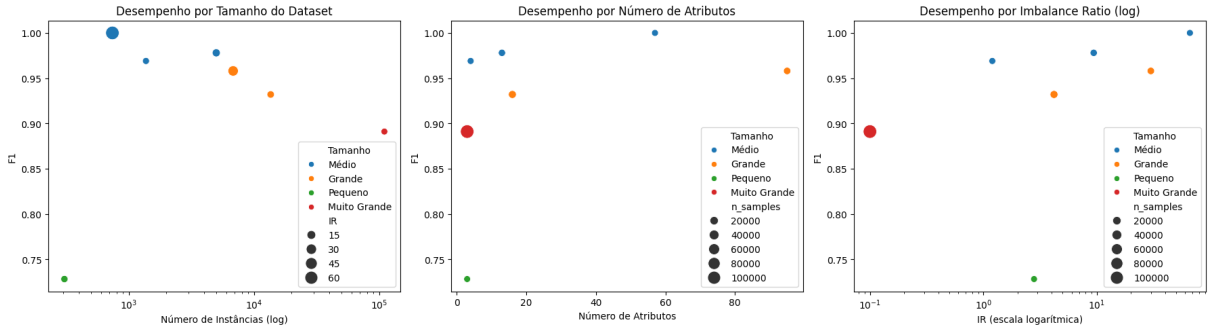
Figura 33: F1-Score: Teste de sensibilidade dos hiperparâmetros DSTO-GAN



Fonte: Dados da Pesquisa.

com alto desbalanceamento (IR $\geq$ 30), onde o método alcançou *F1-Score* (1.00) em diversos casos, como o *Absenteeism_at_work* (IR=66.3) e abalone (IR=32.5), Figura 34.

Figura 34: F1-Score: DSTO-GAN



Fonte: Dados da Pesquisa.

A Tabela 8 resume as configurações ideais para os hiperparâmetros do modelo DSTO-GAN. Os resultados indicam que o método é particularmente adequado para problemas de classificação com as seguintes características: desbalanceamento acentuado entre classes, número moderado de instâncias (500-5.000 amostras), e dimensionalidade intermediária (10-50 atributos). A robustez do método foi comprovada por sua capacidade de manter alto desempenho mesmo em cenários desafiadores, demonstrando superioridade especialmente em comparação com abordagens tradicionais para problemas com alto IR.

Tabela 8: Configurações ideais para o modelo DSTO-GAN

Fator	**Configuração Ideal**	**Justificativa**
dim_h, n_z, lr	$dim\_h = 32$, $n\_z = 5$, lr=0,001	Consistente em todos os cenários.
Número de Instâncias	$500 - 5k$	F1-Score $\geq$0,95. Menos que 500 instâncias desempenho inferior.
Número de Atributos	10–50	Abaixo de 10 ou acima de 50 prejudica o modelo.
Imbalance Ratio (IR)	IR$\geq$30	IR alto beneficia a geração sintética. IR baixo ($<$2) exige muitos dados.

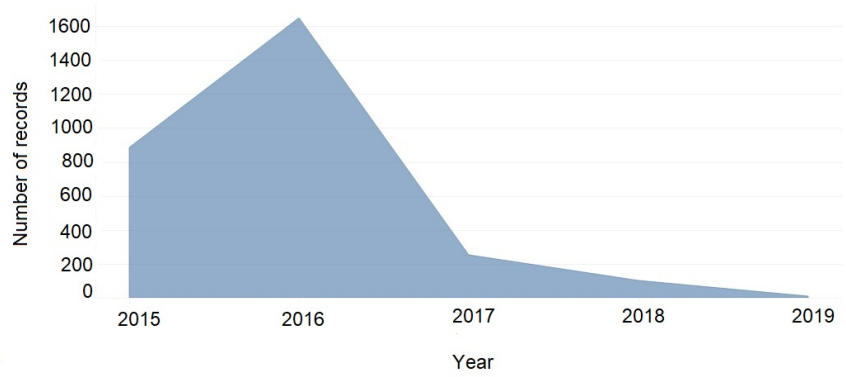
7 ESTUDO DE CASO

Para avaliar as particularidades do DSTO-GAN em conjuntos de dados reais, ele foi testado no RESP-Microcefalia. O RESP-Microcefalia é um formulário *online* desenvolvido pelo DATASUS-Brasil, instituído pelo Ministério da Saúde (MS) desde 19 de novembro de 2015. Tem a intenção de registrar os casos e óbitos suspeitos de alterações no crescimento e desenvolvimento relacionados à infecção pelo vírus Zika e outras etiologias infecciosas (BRASIL et al., 2015).

A base de dados foi fornecida pelo Ministério da Saúde (MS) e não apresenta variáveis que permitem a identificação dos indivíduos e suas famílias. Portanto, o estudo não necessitou de registro e avaliação pelo sistema de Comitês de Ética em Pesquisa, da Comissão Nacional de Ética em Pesquisa (CEP/CONEP), conforme define a Resolução do Conselho Nacional de Saúde (CNS) número 510, de 7 de abril de 2016 (BRASIL; SAúDE; SAúDE, 2019).

As notificações correspondem ao período entre 2015 e 2019. Os casos registrados no RESP com infecção congênita devido ao Zika Vírus tiveram seu pico em 2016 com mais de 1600 registros. A partir de maio de 2016, nota-se uma queda no número de casos, comportamento verificado nos anos subsequentes, conforme apresentado na Figura 35.

Figura 35: Casos confirmados de infecção congênita devido ao Zika Virus no Brasil entre 2015 e 2019



A base de dados do RESP apresenta 43 atributos, organizados em nove (9) categorias:

1. *Notificação*: Apresenta a classificação dos casos suspeitos de infecção congênita (recém nascido, criança, feto em risco, aborto espontâneo ou natimorto) e a data em que foi notificado;

2. *Dados da gestante*: idade, raça/cor e estado de residência (UF);

3. *Informações sobre o nascido vivo*: sexo, data de nascimento, peso (gramas) e comprimento (centímetros);

4. *Dados sobre a gestação e parto*: tipos de alterações congênitas, quando a alteração foi detectada (na gravidez ou após o parto), idade gestacional de detecção da microcefalia, tipo de gravidez, classificação do nascido vivo, perímetro cefálico e data da medição do perímetro cefálico. O tipo de gravidez que pode ser definido em pré-termo (com idade gestacional menor que 37 semanas de gestação), a termo (idade gestacional entre 37 e 41 semanas de gestação), pós-termo (idade gestacional maior que 42 semanas);

5. *Dados clínicos epidemiológicos da mãe*: data do início dos sintomas, o tipo de sintomas (febre, erupções cutâneas, coceira, conjuntivite, dor de cabeça e acometimento neurológico), realização de exame de STORCH[1] e resultado, resultado exame Zika, histórico de arbovírus e mal formação congênita;

6. *Informações sobre os exames de imagem*: ultrassonografia, ultrassonografia transfontanela, tomografia computadorizada e ressonância magnética;

7. *Dados sobre o estabelecimento de saúde*: município e estado;

8. *Dados sobre a evolução da doença*: óbito e data de óbito;

9. *Campos de acesso restrito ao gestor*: Classificação final do caso suspeito de alterações Congênitas e Critério de confirmação através de exames laboratoriais realizados (Zika, Dengue, Chikungunya, STORCH, outros e imagem).

Estas categorias, juntamente com os seus atributos, estão apresentados na Figura 36.

7.1 Pré-processamento da Base de Dados

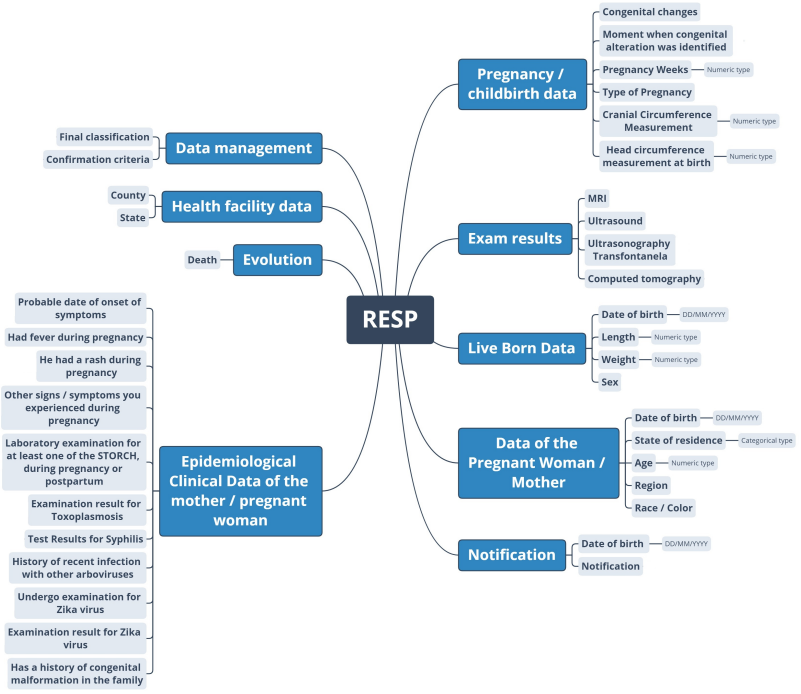
O conjunto de dados inicial compreendia 17.451 instâncias. Para o presente estudo, que visa investigar as características de alterações congênitas em neonatos associadas à infecção pelo vírus Zika, a análise foi restrita às notificações de nascidos vivos e crianças pré-existentes com microcefalia[2]. Adicionalmente, foram excluídos casos com confirmação laboratorial para sífilis ou toxoplasmose, a fim de isolar o efeito da infecção por Zika no desenvolvimento de alterações congênitas.

Ao final do processo de seleção tínhamos 10.510 instâncias, sendo 2.904 crianças

[1] O acrônimo é composto pelos patógenos mais frequentemente relacionados às infecções: bactéria Treponema Pallidum que causa a sífilis (S), o protozoário Toxoplasma Gondii que causa a toxoplasmose (TO) e os vírus da rubéola (R), citomegalovírus (C), vírus herpes simples (H) (RIBEIRO et al., 2018)

[2] A inclusão desses casos, anteriores ao surto de Zika, no Registro de Eventos em Saúde Pública (RESP) foi realizada sob orientação da Secretaria de Saúde para fins de acompanhamento longitudinal (BRASIL et al., 2015)

Figura 36: Categorias e seus respectivos atributos da base de dados do Registro de Eventos em Saúde Pública



diagnosticadas com síndrome congênita devido ao Zika Vírus e 7.606 crianças sem alteração congênita. O processo de seleção de instâncias pode ser resumido na Figura 37:

7.2 Binarização de Atributos

Todos os atributos nominais, não ordinais foram binarizados, conforme apresentado na Tabela 9.

7.3 Seleção de Atributos

A correlação é uma medida padronizada da relação entre duas variáveis e indica a força e a direção do relacionamento linear entre duas variáveis. As variáveis diâmetro encefálico[3] e o perímetro encefálico apresentam correlação muito forte. Mantemos o perímetro encefálico, pois é um critério de confirmação de microcefalia adotado pela Organização Mundial de Saúde (OMS). Também foram excluídos os atributos que estavam altamente correlacionados com o atributo de classificação, tais como os: resultados de exames de imagem, tipo de alteração congênita, quando foi detectada a alteração congênita e idade gestacional na detecção da microcefalia. Ao final do processo, a base de dados apresentava

[3]A circunferência craniana é detectada de forma intrauterina através de exames de imagem.

Figura 37: Seleção de instâncias a partir do RESP

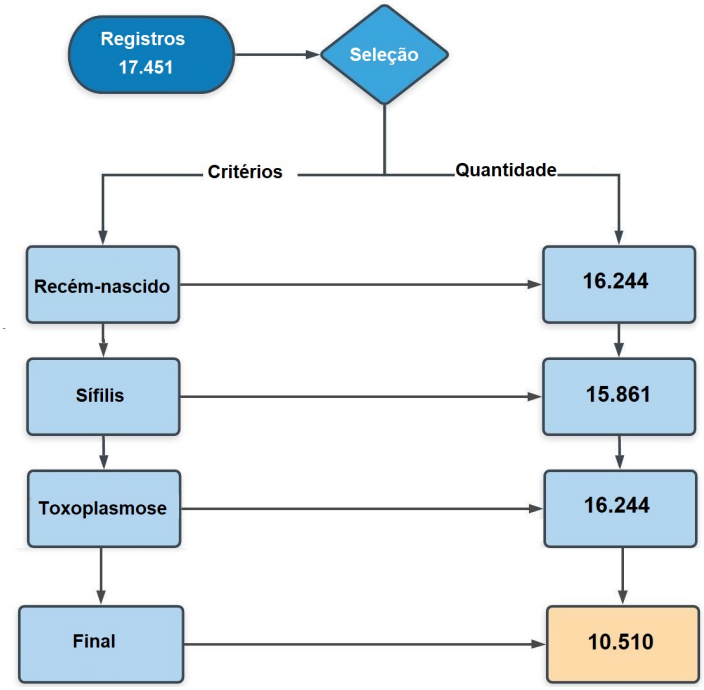


Tabela 9: RESP-Binarização de atributos

Atributo	Opções	Tipo
Sintoma	1) Prurido 2) Hiperemia conjuntival (Conjuntivite não purulenta) 3) Dor em articulação 4) Dor muscular 5) Edema em articulações 6) Cefaleia 7) Hipertrofia Ganglionar 8) Acometimento Neurológico 9) Sem sintomas	Sim Não
Critério de confirmação	1) Sem critério de confirmação 2) Clinico-epidemiologico 3)Laboratorial Zika 4) Laboratorial Dengue 5) Laboratorial Chikungunya 6) Laboratorial STORCH 7) Laboratorial outros 8) Imagem	Sim Não

38 atributos[4].

[4]Uma descrição detalhada destes atributos está disponível em: https://bitlybr.com/ZeU5ZKO

7.4 Dados Inconsistentes

Havia 95 instâncias em que a idade da gestante estava com os valores 2 e 3. Como se trata de uma idade incompatível fisiologicamente para uma concepção, excluímos esses valores e deixamos em branco.

Duas instâncias estavam com o valor do perímetro encefálico medindo 323,3 cm, o que não corresponde a um valor real. Vale ressaltar que os relatos da literatura referem média de perímetro cefálico em RN normais masculinos de 34,61 cm, com variação entre 32,14 e 37,08 cm, e em RN normais femininos média de 34,05 cm, com variação entre 31,58 e 36,52 cm (BRASIL et al., 2015). Como esses valores não correspondem a valores encontrados na literatura, excluímos esses valores, deixando-os ausentes.

7.5 Dados Ausentes

Os dados ausentes são comuns em bancos de dados na área de saúde. Por isso, o uso de métodos adequados torna-se fundamental para diminuir o impacto da ausência de informação.

A base de dados original possuía cerca de 30% de dados ausentes. Realizamos tratamento de dados ausentes através de imputação de dados por meio de média ou mediana.

7.6 Criação dos Modelos de Aprendizado

Separamos 20% de instâncias para teste (Teste na Tabela 10). O conjunto de dados restante, correspondente a 80%, foi balanceado e utilizado para criar os modelos de aprendizado. Durante o processo de treinamento, foi utilizada a validação cruzada de 10 dobras, Tabela 10.

Nossa análise comparativa incluiu cinco algoritmos de classificação distintos: *Decision Tree*, *Random Forest*, *XGBoost*, *Neural Network* (MLP) e KNN, cada um com seus hiperparâmetros otimizados conforme detalhado na Tabela 11.

Tabela 10: Separação do conjunto de dados em treino e teste

Classe	Treino	Teste
Sim	2323	581
Não	6084	1521
Total	**8407**	**2102**

Os hiperparâmetros desempenham um papel fundamental no desempenho dos modelos de aprendizado de máquina. A Tabela 11 resume as configurações principais utilizadas no modelo DSTO-GAN e nos algoritmos classificadores.

Tabela 11: Hiperparâmetros do Modelo DSTO-GAN e Classificadores

Categoria	**Hiperparâmetro**	**Valor**
DSTO-GAN	Dimensão latente (n_z)	10
	Dimensão oculta (dim_h)	64
	Taxa de aprendizado	0,0002
	Épocas	100
	Tamanho do lote	64
Classificadores	Decision Tree (max_depth)	3-15
	Random Forest (n_estimators)	10-100
	Rede Neural (alpha)	1e-4 a 1e-2
	KNN (vizinhos)	3-10
	XGBoost (learning rate)	0,01-0,3
Treinamento	Validação cruzada	10 folds
	Tamanho do teste	20%
	Otimização	BayesSearchCV

7.7 Resultados

Para avaliar o desempenho do DSTO-GAN em um contexto mais amplo, comparamos seus resultados com outros métodos de balanceamento de dados, incluindo CTGAN, GAN, SMOTE, RUS e um cenário sem balanceamento. Para investigar o impacto dessas técnicas no desempenho da classificação, utilizamos os seguintes algoritmos de aprendizado de máquina no treinamento dos modelos: *Random Forest*, *K-Nearest Neighbors* (KNN), *Neural Network*, *XGBoost* e *Decision Tree*.

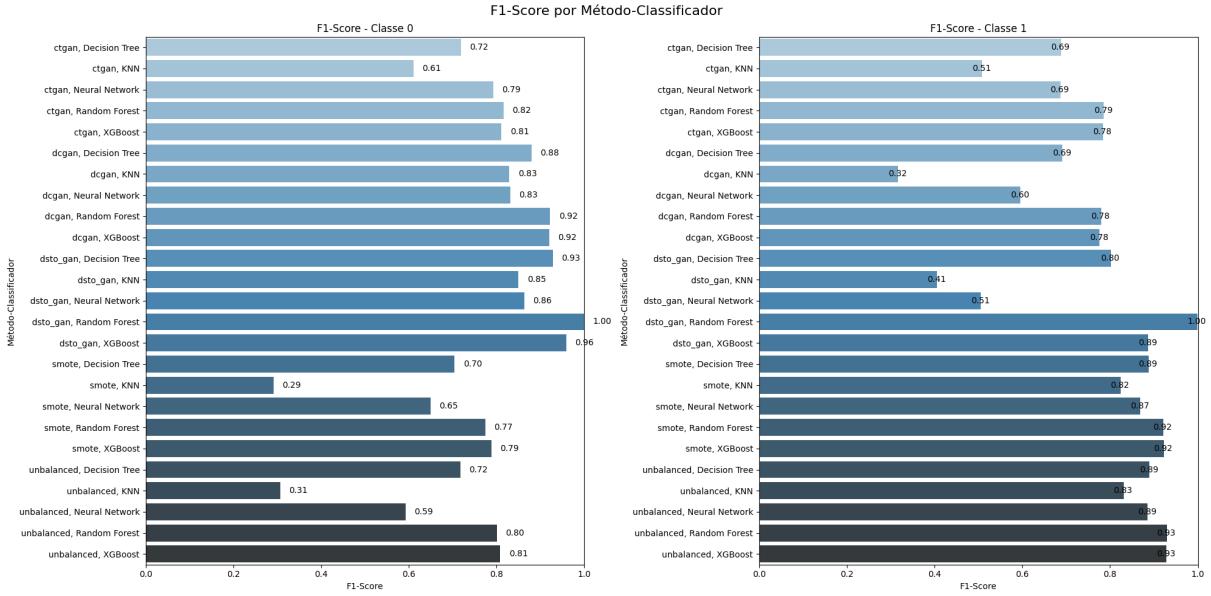
A Figura 38 apresenta o *F1-Score* para diferentes métodos de classificação em duas classes (0 e 1). A classe zero (0) corresponde a sem síndrome congênita e classe um (1) corresponde com síndrome congênita. O *F1-Score* é uma métrica que combina precisão e Revocação, sendo útil para avaliar o desempenho de modelos de classificação, especialmente em conjuntos de dados desbalanceados.

A Figura 39 apresenta a precisão para diferentes métodos de classificação em duas classes (0 e 1). A precisão é uma métrica que indica a proporção de previsões positivas que estão corretas.

A Figura 40 apresenta o Revocação para diferentes métodos de classificação em duas classes (0 e 1). O Revocação é uma métrica que indica a proporção de casos positivos reais que foram corretamente identificados pelo modelo.

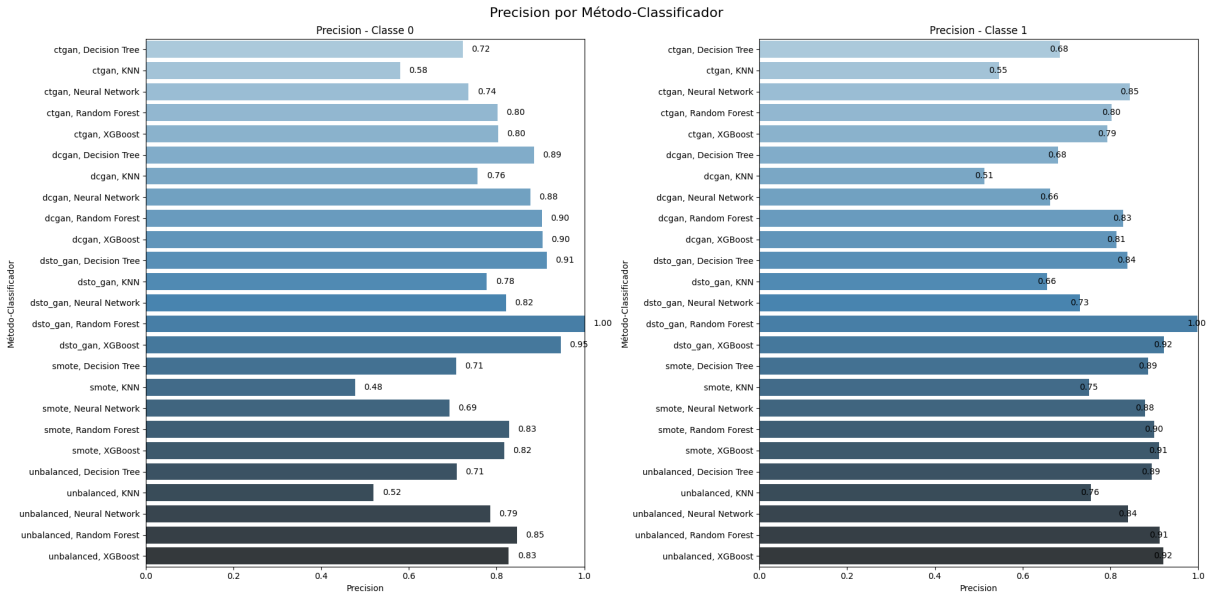
Entre os métodos avaliados, o DSTO-GAN demonstra o melhor desempenho geral, alcançando F1-Score, precisão e Revocação próximos ou iguais a 1.0 em classificadores como *Decision Tree* e *Random Forest*, indicando alta eficácia tanto na classificação da classe majoritária quanto da minoritária. O GAN também se mostra bastante eficiente,

Figura 38: RESP Microcefalia: F1-Score para diferentes combinações de métodos de balanceamento e classificadores



Fonte: Dados da Pesquisa.

Figura 39: RESP Microcefalia: Precisão para diferentes combinações de métodos de balanceamento e classificadores

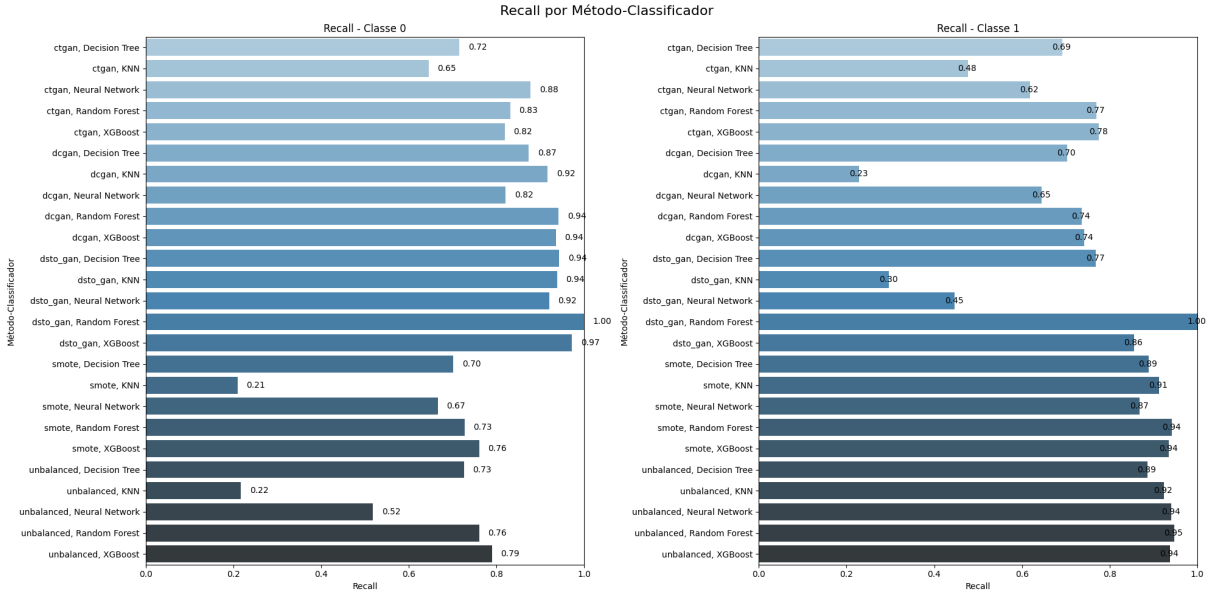


Fonte: Dados da Pesquisa.

especialmente para a classe majoritária, embora com resultados ligeiramente inferiores aos do DSTO-GAN.

Por outro lado, CTGAN e SMOTE apresentam desempenho inconsistente: enquanto

Figura 40: RESP Microcefalia: Revocação para diferentes combinações de métodos de balanceamento e classificadores



Fonte: Dados da Pesquisa.

algoritmos como *Random Forest* e *XGBoost* obtêm resultados aceitáveis, outros, como KNN, têm performance inferior. Já os dados não balanceados revelam as maiores limitações, com baixa eficiência, principalmente na detecção da classe minoritária.

Os resultados evidenciam modelos de balanceamento generativos, especialmente o DSTO-GAN, superam significativamente o desbalanceamento tradicional, garantindo maior equilíbrio e acurácia na classificação. Enquanto o GAN também se destaca, métodos como CTGAN e SMOTE dependem mais do classificador utilizado.

8 CONCLUSÃO

Este trabalho teve como objetivo principal investigar a aplicação de métodos generativos, especificamente Redes Adversárias Generativas (GANs) e *Autoencoders Variacionais* (VAEs), no balanceamento de conjuntos de dados tabulares, com foco na mitigação do problema de classes desequilibradas. A partir das hipóteses propostas e dos objetivos estabelecidos, foram conduzidos experimentos em um conjunto diversificado de 48 bases de dados, abrangendo diferentes níveis de desbalanceamento, dimensionalidade e tipos de atributos (numéricos e categóricos). Os resultados obtidos permitiram validar parcialmente as hipóteses iniciais e oferecer informações sobre o desempenho e as limitações do método DSTO-GAN, bem como sobre a influência de fatores como o índice de desbalanceamento (IR), a natureza dos dados e a escolha do classificador.

8.1 Respostas às Questões de Pesquisa

As questões de pesquisa propostas neste trabalho foram fundamentais para orientar a investigação e a análise dos resultados, buscando explorar e aprimorar o uso de modelos generativos no balanceamento de conjuntos de dados tabulares. A partir dos experimentos realizados e dos resultados obtidos, foi possível responder a essas questões. A seguir, são apresentadas as respostas a cada uma das questões de pesquisa, com base nas evidências coletadas e nas análises conduzidas ao longo deste estudo.

QP1 *Modificações e adaptações em modelos generativos*: Para melhorar o balanceamento de conjuntos de dados tabulares usando modelos generativos, como GANs, VAEs ou métodos baseados em difusão, são necessárias modificações e adaptações específicas que garantam a geração de dados sintéticos realistas, equilibrados e que preservem a estrutura original dos dados. Abaixo estão as principais modificações e estratégias realizadas na pesquisa:

- Arquitetura do Híbrida: GANs puras podem ter dificuldade em aprender distribuições tabulares complexas. Modificamos a arquitetura usando um *encoder* variacional (VAE) para mapear dados tabulares em um espaço latente estruturado. O *decoder* gera amostras sintéticas a partir desse espaço, enquanto o discriminador avalia sua qualidade.

- Geração de Dados Balanceados: O DSTO-GAN Calcula o número de amostras necessárias para cada classe minoritária (*n_to_sample*). E Gera dados sintéticos para as classes sub-representadas usando o procedimento G_SM1, que utiliza o *decoder* para criar novas amostras a partir de ruído Gaussiano.

- Treinamento Adversarial: O discriminador é atualizado com dados reais

e falsos. O gerador (*encoder* + *decoder*) é otimizado para enganar o discriminador (além de minimizar o erro de reconstrução MSE).

QP2 *Comparação com outras abordagens de balanceamento*: A segunda questão de pesquisa investigou como o desempenho de métodos generativos se compara com outras abordagens de balanceamento em termos de eficácia no balanceamento de classes e na acurácia de modelos de classificação. Os resultados demonstraram que o DSTO-GAN mostrou-se eficaz em cenários de desbalanceamento moderado e dados numéricos, superando abordagens tradicionais quando combinado com classificadores robustos como Random Forest. Por outro lado, em situações mais desafiadoras (desbalanceamento extremo, alta dimensionalidade ou muitos atributos categóricos, seu desempenho foi inferior ou equivalente a outras técnicas, indicando que sua eficácia depende fortemente do contexto de aplicação.

QP3 *Configurações e funcionalidades para otimização*:

A terceira questão de pesquisa investigou as configurações e funcionalidades que otimizam o DSTO-GAN, considerando desempenho, estabilidade e eficiência computacional. Para isso, realizou-se uma análise sistemática de hiperparâmetros, avaliando como diferentes combinações impactam a qualidade da geração de dados sintéticos. O estudo focou em três parâmetros-chave: a dimensão da camada oculta (dim_h), o tamanho do espaço latente (n_z) e a taxa de aprendizado (lr). Os valores testados para dim_h (32 e 64) foram selecionados com base em arquiteturas similares de GANs, priorizando um equilíbrio entre capacidade de modelagem e custo computacional. Verificou-se que redes mais profundas não melhoraram significativamente os resultados, apenas aumentando o tempo de treinamento. Quanto ao espaço latente (n_z), as dimensões 5 e 10 foram escolhidas por serem suficientes para capturar a variabilidade dos dados sem introduzir ruído ou complexidade desnecessária. A taxa de aprendizado (lr) foi analisada em dois valores típicos para GANs (0,0002 e 0,001), garantindo que o modelo convergisse de forma estável, sem oscilações ou estagnação. A configuração ótima identificada, dim_h=32, n_z=5, lr=0,001, mostrou-se robusta em diferentes cenários, alcançando alto F1-Score mesmo em conjuntos de dados desbalanceados. Além da seleção de hiperparâmetros, o estudo determinou as condições ideais para aplicação do DSTO-GAN, como conjuntos de dados com pelo menos 5.000 amostras e dimensionalidade moderada (10 a 100 atributos). O modelo também demonstrou maior eficácia em situações de alto desbalanceamento (IR >= 10), onde técnicas convencionais tendem a falhar. Essas otimizações implícitas reforçam a adaptabilidade do método a problemas reais, mantendo eficiência computacional sem sacrificar o desempenho.

QP4 *Desempenho em conjuntos numéricos e categóricos*:

A quarta questão de pesquisa investigou em quais tipos de conjuntos de dados - numéricos ou categóricos - os métodos baseados em modelos generativos apresentam melhor desempenho comparado a outras técnicas de balanceamento. Os resultados revelaram diferenças significativas no comportamento do DSTO-GAN conforme a natureza dos dados. Para atributos categóricos, o DSTO-GAN demonstrou clara superioridade sobre os demais métodos, mostrando-se particularmente eficaz na preservação da estrutura discreta característica desses dados. Essa vantagem foi consistente na maioria dos casos analisados, com exceção de poucas situações onde obteve desempenho equivalente. Quando aplicado a dados numéricos, o método manteve um bom desempenho, porém com resultados mais equilibrados em relação às técnicas convencionais. Em diversos casos, métodos como SMOTE e GAN alcançaram eficácia comparável, especialmente quando devidamente ajustados para distribuições contínuas específicas. Um ponto de atenção foi identificado no conjunto *BankNote_Authentication*, onde o DSTO-GAN enfrentou dificuldades particulares. Este caso aponta para limitações em cenários com alta dimensionalidade ou distribuições numéricas específicas (ex.: dados médicos com padrões complexos).

8.2 Validação das Hipóteses

As hipóteses deste trabalho foram planejadas para orientar a pesquisa sobre a aplicação de modelos generativos, especificamente o DSTO-GAN, no balanceamento de conjuntos de dados tabulares. A seguir, são apresentados os resultados que confirmam, refutam ou contextualizam cada uma das hipóteses, oferecendo uma compreensão mais clara sobre o desempenho e as limitações do método DSTO-GAN em diferentes cenários.

H1 *É possível adaptar algoritmos baseados em modelos generativos para conjuntos de dados tabulares, preservando sua eficácia em contextos de dados heterogêneos e complexos*:

A hipótese H1 foi parcialmente confirmada. Os resultados obtidos revelam um cenário promissor, porém com desafios específicos que exigem considerações cuidadosas. Os experimentos demonstraram que o DSTO-GAN pode ser efetivamente adaptado para dados tabulares, mostrando desempenho particularmente forte em duas situações-chave: para conjuntos com atributos categóricos e em problemas de classificação binária. No entanto, a adaptação revelou limitações importantes em cenários específicos dados com alta dimensionalidade, conjuntos severamente desbalanceados e casos que combinam atributos numéricos complexos com classificadores específicos (como redes neurais).

H2 *Métodos de balanceamento de dados baseados em modelos generativos podem superar*

outras abordagens de balanceamento em termos de eficácia no balanceamento de classes e precisão de modelos de classificação:
A hipótese H2 foi confirmada em cenários específicos. O DSTO-GAN se mostram particularmente eficaze para: conjuntos categóricos ou mistos, em problemas de classificação binária e em cenários com desbalanceamento moderado. Para outros contextos, como dados puramente numéricos de alta dimensionalidade ou problemas multiclasse extremos, abordagens tradicionais ou híbridas podem ser mais adequadas.

H3 *Métodos de balanceamento baseados em modelos generativos tendem a apresentar melhores resultados em conjuntos de dados predominantemente numéricos quando comparados a outras abordagens de balanceamento*:
A hipótese H3 não foi confirmada. Embora o DSTO-GAN tenha apresentado desempenho competitivo neste tipo de dados, sua vantagem não foi decisiva quando comparada a outras abordagens. Em diversos casos, técnicas convencionais como SMOTE (com ajustes adequados) e GAN alcançaram resultados equivalentes, especialmente quando aplicadas a distribuições numéricas específicas. O contraste mais significativo surgiu na avaliação de dados categóricos, onde o DSTO-GAN mostrou desempenho claramente superior à maioria das alternativas. Essa diferença se deve principalmente à capacidade intrínseca dos modelos generativos de preservar adequadamente a estrutura discreta característica desses dados - uma limitação conhecida dos métodos tradicionais de balanceamento.

H4 *O índice de desbalanceamento (IR) pode ter uma influência direta no desempenho de algoritmos baseados em modelos generativos. A hipótese é que quanto maior o desbalanceamento, melhor o desempenho desses métodos em ajustar a distribuição das classes*:
Os resultados refutam a hipótese inicial de que maior desbalanceamento levaria a melhor desempenho. Na realidade O DSTO-GAN mostrou-se mais eficaz em cenários de desbalanceamento leve a moderado, para desbalanceamentos extremos (IR grande), o método enfrentou dificuldades significativas. A relação entre IR e desempenho segue uma curva não monotônica, com pico de eficiência em níveis intermediários.

8.3 Contribuições e Limitações

Este estudo contribuiu para demonstrar que métodos generativos podem ser eficazes no balanceamento de dados tabulares, especialmente em cenários de desbalanceamento extremo (tanto muito baixo quanto muito alto) e em dados categóricos, enquanto em conjuntos numéricos ou com desbalanceamento moderado outras técnicas podem ser

igualmente adequadas. O trabalho também estabeleceu relações críticas entre a eficácia do método e a escolha do classificador, com melhores resultados em combinação com Random Forest e XGBoost.

Em conjuntos de dados particularmente grandes e complexos, o DSTO-GAN enfrentou dificuldades em: manter a consistência na geração de instâncias sintéticas, preservar relações complexas entre múltiplos atributos e lidar com a combinação de alta dimensionalidade e desbalanceamento extremos. Estes desafios apontam para limitações arquiteturais na manipulação de espaços de características muito amplos e heterogêneos.

8.4 Recomendações para Trabalhos Futuros

Com base nos resultados obtidos, sugere-se as seguintes direções para pesquisas futuras:

- Investigar técnicas de pré-processamento mais robustas, como, por exemplo, a codificação eficiente de variáveis categóricas.
- Realizar uma otimização mais abrangente dos classificadores, especialmente para KNN e *MLP*, incluindo parâmetros críticos como métricas de distância, funções de ativação e regularização.
- Incorporar o Coeficiente Kappa de Cohen como parte da avaliação, juntamente com as métricas já adotadas, permitindo uma análise ainda mais abrangente, considerando não apenas os erros de classificação, mas também a concordância do modelo além do esperado aleatoriamente.
- Estudo de Casos Multiclasse: O desempenho do DSTO-GAN em problemas multiclasse foi inferior ao observado em problemas binários. Pesquisas futuras podem focar no desenvolvimento de estratégias específicas para lidar com a complexidade inerente a conjuntos de dados multiclasse.
- Análise de Custo-Benefício: Uma análise detalhada do custo computacional versus o ganho em desempenho pode ajudar a determinar em quais cenários o uso do DSTO-GAN é justificável, especialmente em aplicações que exigem processamento em tempo real ou com recursos limitados.

8.5 Considerações Finais

As conclusões deste trabalho evidenciam que os métodos generativos, em particular o DSTO-GAN, representam uma contribuição significativa para o campo de balanceamento de dados, embora seu desempenho varie consideravelmente conforme o contexto de aplicação. Os resultados demonstram que essas técnicas são particularmente eficazes para conjuntos categóricos e problemas binários, onde superam abordagens tradicionais

na preservação da estrutura dos dados e na melhoria da performance dos classificadores. No entanto, as limitações identificadas em cenários de desbalanceamento extremo e dados multiclasse indicam que ainda há espaço para avanços metodológicos.

O DSTO-GAN apresentou melhor desempenho, particularmente em conjuntos de dados com dimensões moderadas (500-5.000 instâncias e 10-50 atributos), onde alcançou F1-score $\geq$ 0,95. O método mostrou-se especialmente eficaz em cenários com alto desbalanceamento (IR $\geq$ 30), com (F1-score = 1.00) em diversos casos, como evidenciado nos conjuntos*Absenteeism_at_work* (IR=66.3) e *Abalone* (IR=32.5).

Embora o DSTO-GAN ofereça vantagens em termos de qualidade de balanceamento, seu custo computacional o torna menos adequado para aplicações que exigem processamento em tempo real ou para prototipagem rápida. O SMOTE, por sua vez, mantém-se como alternativa eficiente para cenários onde o tempo de processamento é fator crítico, apresentando desempenho temporal consistente independentemente do tamanho do conjunto de dados.

A seleção do método de balanceamento ideal deve considerar um equilíbrio entre as necessidades específicas de cada aplicação, os recursos computacionais disponíveis e a importância relativa da qualidade do balanceamento versus o tempo de processamento.

Este estudo reforça a importância de uma abordagem contextualizada no tratamento de dados desbalanceados, onde a seleção da técnica ideal deve considerar não apenas o método de balanceamento, mas também as características intrínsecas dos dados e os objetivos específicos da modelagem. A combinação de técnicas, incluindo possíveis híbridos entre métodos generativos e convencionais, mostra-se promissora para superar as limitações identificadas.

Por fim, os resultados obtidos abrem caminho para pesquisas futuras que explorem otimizações arquiteturais, aplicações em novos domínios e análises de custo-benefício mais aprofundadas. A evolução contínua desses métodos poderá, assim, consolidar os modelos generativos como ferramentas indispensáveis no arsenal de aprendizado de máquina para problemas desbalanceados, desde que aplicados com critério e adaptados às particularidades de cada cenário.

REFERÊNCIAS

ABDULGANIYU, O. H. et al. Xidintfl-vae: Xgboost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder. The Journal of Supercomputing, Springer, v. 81, n. 1, p. 1–38, 2025.

AGGARWAL, C. C.; ZHAI, C. A Survey of Text Classification Algorithms. Boston, MA: Springer US, 2012. 163–222 p. Disponível em: <https://doi.org/10.1007/978-1-4614-3223-4_6>.$ISBN978 - 1 - 4614 - 3223 - 4$.

ALMHAITHAWI, D.; JAFAR, A.; ALJNIDI, M. Example-dependent cost-sensitive credit cards fraud detection using smote and bayes minimum risk. SN Applied Sciences, Springer, v. 2, p. 1–12, 2020.

ANSHELEVICH, D.; KATZ, G. Synthetic tabular data generation using a vae-gan architecture. Available at SSRN 4902016, 2024.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein GAN. 2017. Disponível em: <https://arxiv.org/abs/1701.07875>.

BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders. Machine learning for data science handbook: data mining and knowledge discovery handbook, Springer, p. 353–374, 2023.

BARUA, S. et al. Mwmote-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans. Knowl. Data Eng., v. 26, n. 2, p. 405–425, 2014. Disponível em: <https://ieeexplore.ieee.org/document/6361394>.

BATISTA, G. E. A. P. A.; PRATI, R.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor., v. 6, p. 20–29, 2004.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl., Association for Computing Machinery, New York, NY, USA, v. 6, n. 1, p. 20–29, jun 2004. ISSN 1931-0145. Disponível em: <https://doi.org/10.1145/1007730.1007735>.

BELENKO, V. et al. Evaluation of gan applicability for intrusion detection in self-organizing networks of cyber physical systems. In: IEEE. 2018 International Russian Automation Conference (RusAutoCon). [S.l.], 2018. p. 1–7.

BRASIL; SAúDE, M. da; SAúDE, C. N. de. Resolução Nº 510, de 07 de abril de 2016. jul 2019. Disponível em: <http://conselho.saude.gov.br/resolucoes/2016/Reso510.pdf>.

BRASIL et al. Protocolo de vigilância e resposta à ocorrência de microcefalia e/ou alterações do sistema nervoso central (SNC): emergência de saúde pública de importância internacional. [S.l.]: Ministério da Saúde, 2015.

BREIMAN, L. Random forests. Machine Learning, v. 45, p. 5–32, 10 2001.

CHAWLA, N.; JAPKOWICZ, N.; KOLCZ, A. Workshop learning from imbalanced data sets ii. In: Proceedings of international conference on machine learning. [S.l.: s.n.], 2003.

CHAWLA, N. V. et al. Smote: Synthetic minority over-sampling technique. Jornal de pesquisa de inteligência artificial, v. 16, p. 321–357, 2002.

CHAWLA, N. V. et al. Smoteboost: Improving prediction of the minority class in boosting. In: SPRINGER. Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7. [S.l.], 2003. p. 107–119.

CHEN, M.-Y.; CHIANG, H.-S.; HUANG, W.-K. Efficient generative adversarial networks for imbalanced traffic collision datasets. IEEE Transactions on Intelligent Transportation Systems, v. 23, n. 10, p. 19864–19873, 2022.

CHEN, T. Xgboost: extreme gradient boosting. R package version 0.4-2, v. 1, n. 4, 2015.

CHUANG, P.-J.; HUANG, P.-Y. B-vae: a new dataset balancing approach using batched variational autoencoders to enhance network intrusion detection. The Journal of Supercomputing, Springer, v. 79, n. 12, p. 13262–13286, 2023.

COVER, T.; HART, P. Nearest neighbor pattern classification. IEEE transactions on information theory, IEEE, v. 13, n. 1, p. 21–27, 1967.

CURRAN-EVERETT, D.; MILGROM, H. Post-hoc data analysis: benefits and limitations. Current opinion in allergy and clinical immunology, LWW, v. 13, n. 3, p. 223–224, 2013.

DABLAIN, D.; KRAWCZYK, B.; CHAWLA, N. V. Deepsmote: Fusing deep learning and smote for imbalanced data. arXiv, 2021.

DABLAIN, D. A.; CHAWLA, N. V. Towards understanding how data augmentation works with imbalanced data. arXiv preprint arXiv:2304.05895, 2023.

DIEZ-OLIVAN, A. et al. Kernel-based support vector machines for automated health status assessment in monitoring sensor data. Int J Adv Manuf Technol, 2018.

DING, H. et al. Rvgan-tl: A generative adversarial networks and transfer learning-based hybrid approach for imbalanced data classification. Information Sciences, v. 629, p. 184–203, 2023. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025523001615>.

DONG, T. et al. Towards Fast Network Intrusion Detection based on Efficiency-preserving Federated Learning. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). [S.l.: s.n.], 2021. p. 468–475.

DOU, Q. et al. Domain generalization via model-agnostic learning of semantic features. Advances in neural information processing systems, v. 32, 2019.

DOUZAS, G.; BACAO, F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications, v. 91, p. 464 – 471, 2018. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S095741 34 7417306346>.

DUBEY, R. et al. Analysis of sampling techniques for imbalanced data: An n= 648 adni study. NeuroImage, Elsevier, v. 87, p. 220–241, 2014.

ENGELMANN, J.; LESSMANN, S. Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications, Elsevier, v. 174, p. 114582, 2021.

FAN, M. et al. Cluster-based generative adversarial network imbalanced data generation method. In: 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS). [S.l.: s.n.], 2021. p. 547–552.

FERNÁNDEZ, A. Learning from imbalanced data sets. [S.l.]: Springer, 2018.

FERNANDEZ, A. et al. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res., AI Access Foundation, v. 61, p. 863–905, abr. 2018.

FIORE, U. et al. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, Elsevier, v. 479, p. 448–455, 2019.

GARCÍA, S.; HERRERA, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary computation, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 17, n. 3, p. 275–306, 2009.

GHOSH, K. et al. The class imbalance problem in deep learning. Mach. Learn., Springer Science and Business Media LLC, v. 113, n. 7, p. 4845–4901, jul. 2024.

GONZÁLEZ-ESTRADA, E.; COSMES, W. Shapiro–wilk test for skew-normal distributions based on data transformations. Journal of Statistical Computation and Simulation, Taylor &Francis, v. 89, n. 17, p. 3258–3272, 2019.

GOODFELLOW, I. J. NIPS 2016 tutorial: Generative adversarial networks. CoRR, abs/1701.00160, 2017. Disponível em: <http://arxiv.org/abs/1701.00160>.

GOODFELLOW, I. J. et al. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 2672–2680.

GULRAJANI, I. et al. Improved training of wasserstein gans. CoRR, abs/1704.00028, 2017. Disponível em: <http://arxiv.org/abs/1704.00028>.

GUO, H.; VIKTOR, H. L. Learning from imbalanced data sets with boosting and data generation. SIGKDD Explor., Association for Computing Machinery (ACM), v. 6, n. 1, p. 30–39, jun. 2004.

GUPTA, A. et al. Class-weighted evaluation metrics for imbalanced data classification. openreview.net, 2020.

HABIBI, O.; CHEMMAKHA, M.; LAZAAR, M. Imbalanced tabular data modelization using ctgan and machine learning to improve iot botnet attacks detection. Eng. Appl. Artif. Intell., Pergamon Press, Inc., USA, v. 118, n. C, feb 2023. ISSN 0952-1976. Disponível em: <https://doi.org/10.1016/j.engappai.2022.105669>.

Haibo He et al. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). [S.l.: s.n.], 2008. p. 1322–1328.

HAN, J.; KAMBER, M.; PEI, J. Data Mining: Concepts and Techniques. 3. ed. Oxford, England: Morgan Kaufmann, 2011. (The Morgan Kaufmann Series in Data Management Systems).

HAN, J.; KAMBER, M.; PEI, J. Data mining: concepts and techniques. Norwel, MA, USA: Elsevier, 2012.

HART, P. E. The condensed nearest neighbor rule. IEEE Transaction on Information Theory, may 1968.

HE, H.; GARCIA, E. A. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, v. 21, n. 9, p. 1263–1284, 2009.

HEJAZI, S.; PACKIANATHER, M.; LIU, Y. A novel approach using wgan-gp and conditional wgan-gp for generating artificial thermal images of induction motor faults. Procedia Computer Science, v. 225, p. 3681–3691, 2023. ISSN 1877-0509. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050923015211>. 27th International Conference on Knowledge Based and Intelligent Information and Engineering Sytems (KES 2023).

HU, A. et al. Tablegan-mca: Evaluating membership collisions of gan-synthesized tabular data releasing. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery, 2021. (CCS '21), p. 2096–2112. ISBN 9781450384544. Disponível em: <https://doi.org/10.1145/3460120.3485251>.

HUANG, H. et al. Synthetic data for enhanced privacy: A vae-gan approach against membership inference attacks. Knowledge-Based Systems, Elsevier, v. 309, p. 112899, 2025.

ISLAM, M. A. et al. An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. Journal of Information Security and Applications, v. 78, p. 103618, 2023. ISSN 2214-2126. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214212623002028>.

JAFARIGOL, E.; TRAFALIS, T. A review of machine learning techniques in imbalanced data and future trends. arXiv preprint arXiv:2310.07917, 2023.

JAIN, D. K. et al. An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. Information Processing &Management, Elsevier, v. 59, n. 1, p. 102758, 2022.

JAPKOWICZ, N. et al. Learning from imbalanced data sets: a comparison of various strategies. In: AAAI PRESS MENLO PARK. AAAI workshop on learning from imbalanced data sets. [S.l.], 2000. v. 68, p. 10–15.

JEDRZEJOWICZ, J.; JEDRZEJOWICZ, P. Gep-based classifier for mining imbalanced data. Expert Systems with Applications, v. 164, p. 114058, 2021. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417420308204>.

JEONG, J.; JEONG, H.; KIM, H.-J. Bamtgan: A balanced augmentation technique for tabular data. In: 2023 9th International Conference on Applied System Innovation (ICASI). [S.l.: s.n.], 2023. p. 205–207.

JOHNSON, J. M.; KHOSHGOFTAAR, T. M. Survey on deep learning with class imbalance. Journal of big data, Springer, v. 6, n. 1, p. 1–54, 2019.

JORDON, J.; YOON, J.; SCHAAR, M. V. D. Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations. [S.l.: s.n.], 2018.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, v. 30, 2017.

KHADKA, K. et al. Synthetic data generation using combinatorial testing and variational autoencoder. In: IEEE. 2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). [S.l.], 2023. p. 228–236.

KHAN, A. A.; CHAUDHARI, O.; CHANDRA, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. Expert Systems with Applications, v. 244, p. 122778, 2024. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417423032803>.

KIM, J. et al. Oct-gan: Neural ode-based conditional tabular gans. In: Proceedings of the Web Conference 2021. [S.l.: s.n.], 2021. p. 1506–1515.

KIM, J.; LEE, C.; PARK, N. Stasy: Score-based tabular data synthesis. arXiv preprint arXiv:2210.04018, 2022.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. In: BENGIO, Y.; LECUN, Y. (Ed.). 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. [s.n.], 2014. Disponível em: <http://arxiv.org/abs/1312.6114>.

KOTELNIKOV, A. et al. Tabddpm: Modelling tabular data with diffusion models. In: PMLR. International Conference on Machine Learning. [S.l.], 2023. p. 17564–17579.

KOZIARSKI, M.; WOZNIAK, M.; KRAWCZYKZ, B. Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise. CoRR, abs/2004.03406, 2020. Disponível em: <https://arxiv.org/abs/2004.03406>.

KRAMER, O.; KRAMER, O. Scikit-learn. Machine learning for evolution strategies, Springer, p. 45–53, 2016.

KRAWCZYK, B. Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell., Springer Science and Business Media LLC, v. 5, n. 4, p. 221–232, nov. 2016.

KRAWCZYK, B.; KOZIARSKI, M.; WOZNIAK, M. Radial-based oversampling for multiclass imbalanced data classification. IEEE Transactions on Neural Networks and Learning Systems, v. 31, p. 2818 – 2831, 08 2020.

KRSTAJIC, D. et al. Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of cheminformatics, v. 6, p. 10, 03 2014.

KUBAT, M. Addressing the curse of imbalanced training sets: One-sided selection. Fourteenth International Conference on Machine Learning, 06 2000.

KUBAT, M.; MATWIN, S. et al. Addressing the curse of imbalanced training sets: one-sided selection. In: CITESEER. Icml. [S.l.], 1997. v. 97, n. 1, p. 179.

LAURIKKALA, J. Improving identification of difficult small classes by balancing class distribution. In: HEIDELBERG, S.-V. (Ed.). Conference on Artificial Intelligence in Medicine in Europe. AIME 2001: Artificial Intelligence in Medicine. [S.l.: s.n.], 2001. p. 63–66.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: IEEE. Proceedings of 2010 IEEE international symposium on circuits and systems. [S.l.], 2010. p. 253–256.

LEE, C.; KIM, J.; PARK, N. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In: PMLR. International Conference on Machine Learning. [S.l.], 2023. p. 18940–18956.

LEE, J.; PARK, K. Gan-based imbalanced data intrusion detection system. Personal and Ubiquitous Computing, Springer, v. 25, n. 1, p. 121–128, 2021.

LEI, K. et al. Generative adversarial fusion network for class imbalance credit scoring. Neural Computing and Applications, Springer, v. 32, p. 8451–8462, 2020.

LI, J. et al. Taegan: Generating synthetic tabular data for data augmentation. arXiv preprint arXiv:2410.01933, 2024.

LI, Z.; KAMNITSAS, K.; GLOCKER, B. Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation. In: SPRINGER. Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22. [S.l.], 2019. p. 402–410.

LIU, D. et al. Diwift: Discovering instance-wise influential features for tabular data. In: Proceedings of the ACM Web Conference 2023. New York, NY, USA: Association for Computing Machinery, 2023. (WWW '23), p. 1673–1682. ISBN 9781450394161. Disponível em: <https://doi.org/10.1145/3543507.3583382>.

LIU, X.-Y.; WU, J.; ZHOU, Z.-H. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), IEEE, v. 39, n. 2, p. 539–550, 2008.

LOEZER, L. et al. Cost-sensitive learning for imbalanced data streams. In: Proceedings of the 35th annual ACM symposium on applied computing. [S.l.: s.n.], 2020. p. 498–504.

MACI, A. et al. Unbalanced web phishing classification through deep reinforcement learning. Computers, MDPI, v. 12, n. 6, p. 118, 2023.

MAHINNEZHAD, S. et al. Data Augmentation and Class Imbalance Compensation Using CTGAN to Improve Gas Detection Systems. In: 2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). [S.l.: s.n.], 2024. p. 1–6.

MANI, I.; ZHANG, I. knn approach to unbalanced data distributions: a case study involving information extraction. In: ICML. Proceedings of workshop on learning from imbalanced datasets. [S.l.], 2003. v. 126, n. 1, p. 1–7.

MCKIGHT, P. E.; NAJAB, J. Kruskal-wallis test. The corsini encyclopedia of psychology, Wiley Online Library, p. 1–1, 2010.

MEASE, D.; WYNER, A. J.; BUJA, A. Boosted classification trees and class probability/quantile estimation. Journal of Machine Learning Research, v. 8, n. 3, 2007.

MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

MISHRA, P. et al. Application of student's t test, analysis of variance and covariance. Annals of Cardiac Anesthesia, v. 22, n. 4, p. 407–411, 2019.

MOHAMMADPOUR, S. I.; KHEDMATI, M.; ZADA, M. J. H. Classification of truck-involved crash severity: Dealing with missing, imbalanced, and high dimensional safety data. PLoS one, Public Library of Science San Francisco, CA USA, v. 18, n. 3, p. e0281901, 2023.

MRABET, H. et al. A survey of iot security based on a layered architecture of sensing and data analysis. Sensors, MDPI, v. 20, n. 13, p. 3625, 2020.

MULLICK, S. S.; DATTA, S.; DAS, S. Generative adversarial minority superampling. CoRR, abs / 1903.09730, 2019. Disponível em: <http://arxiv.org/abs/1903.09730>.

NDICHU, S. et al. Security-Alert Screening with Oversampling Based on Conditional Generative Adversarial Networks. In: 2022 17th Asia Joint Conference on Information Security (AsiaJCIS). [S.l.: s.n.], 2022. p. 1–7.

NEUMANN, J. V.; MORGENSTERN, O. Theory of games and economic behavior. Princeton, NJ, US: Princeton University Press, 1944. xviii, 625-xviii, 625 p. (Theory of games and economic behavior.).

NGUYEN, S. et al. Effects of resampling techniques on imbalanced data classification: A new under-resampling method. In: Advances in Business and Management Forecasting. [S.l.]: Emerald Publishing Limited, 2021. p. 51–70.

NICKERSON, A.; JAPKOWICZ, N.; MILIOS, E. Using unsupervised learning to guide re-sampling in imbalanced data sets. Proceedings of the Eighth International Workshop on AI and Statitsics, 01 2001.

NORDSTOKKE, D. W.; ZUMBO, B. D. A new nonparametric levene test for equal variances. Psicológica, Universitat de València, v. 31, n. 2, p. 401–430, 2010.

ORESKI, G. Synthesizing credit data using autoencoders and generative adversarial networks. Knowledge-Based Systems, v. 274, p. 110646, 2023. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705123003969>.

PARFENOV, D. et al. Investigation of the Security of ML-models in IoT Networks from Adversarial Attacks. In: 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). [S.l.: s.n.], 2023. p. 229–232.

PARK, N. et al. Data synthesis based on generative adversarial networks. arXiv preprint arXiv:1806.03384, 2018.

PERERA, P.; OZA, P.; PATEL, V. M. One-Class Classification: A Survey. 2021. Disponível em: <https://arxiv.org/abs/2101.03064>.

PRATI, R.; BATISTA, G.; MONARD, M.-C. Class imbalances versus class overlapping: An analysis of a learning system behavior. In: . [S.l.: s.n.], 2004. p. 312–321.

PROKHORENKOVA, L. et al. Catboost: unbiased boosting with categorical features. Advances in neural information processing systems, v. 31, 2018.

QUINLAN, J. R. C4. 5: programs for machine learning. [S.l.]: Elsevier, 2014.

REZVANI, S.; WANG, X. A broad review on class imbalance learning techniques. Applied Soft Computing, v. 143, p. 110415, 2023. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494623004337>.

RIBEIRO, I. G. et al. Microcefalia no Piauí, Brasil: estudo descritivo durante a epidemia do vírus Zika, 2015-2016. Epidemiologia e Serviço de Saúde, scielo, v. 27, 00 2018. ISSN 2237-9622. Disponível em: <https://www.scielo.br/j/ress/a/mW3drcZMG3ZwJ5TTWWCcRnm/?lang=pt>.

RODRIGUEZ-ALMEIDA, A. J. et al. Synthetic Patient Data Generation and Evaluation in Disease Prediction Using Small and Imbalanced Datasets. IEEE Journal of Biomedical and Health Informatics, v. 27, n. 6, p. 2670–2680, 2023.

ROSS, T.-Y.; DOLLÁR, G. Focal loss for dense object detection. In: proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2017. p. 2980–2988.

RUAN, Y.-P. et al. Condition-transforming variational autoencoder for conversation response generation. In: IEEE. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.], 2019. p. 7215–7219.

SCHULTZ, K. et al. Convgen: A convex space learning approach for deep-generative oversampling and imbalanced classification of small tabular datasets. Pattern

Recognition, v. 147, p. 110138, 2024. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S003132032300835X>.

SCHöLKOPF, B. et al. Estimating the support of a high-dimensional distribution. Neural Computation, 2001.

SEIFFERT, C. et al. Rusboost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, v. 40, n. 1, p. 185–197, 2010.

SERAJ, A. et al. Chapter 5 - cross-validation. In: ESLAMIAN, S.; ESLAMIAN, F. (Ed.). Handbook of Hydroinformatics. Elsevier, 2023. p. 89–105. ISBN 978-0-12-821285-1. Disponível em: <https://www.sciencedirect.com/science/article/pii/B978012821285100021X>.

SHAFQAT, W.; BYUN, Y.-C. A Hybrid GAN-Based Approach to Solve Imbalanced Data Problem in Recommendation Systems. IEEE Access, v. 10, p. 11036–11047, 2022.

SHANNON, C. E. XXII. programming a computer for playing chess. Lond. Edinb. Dublin Philos. Mag. J. Sci., Informa UK Limited, v. 41, n. 314, p. 256–275, mar. 1950.

SOLEIMANI, M. et al. Imbalanced multiclass medical data classification based on learning automata and neural network. EAI Endorsed Transactions on AI and Robotics, v. 2, 2023.

ST, L.; WOLD, S. et al. Analysis of variance (anova). Chemometrics and intelligent laboratory systems, Elsevier, v. 6, n. 4, p. 259–272, 1989.

SUH, N. et al. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. arXiv preprint arXiv:2310.15479, 2023.

SUN, S.; WANG, T.; CHU, F. A multi-learner neural network approach to wind turbine fault diagnosis with imbalanced data. Renewable energy, Elsevier, v. 208, p. 420–430, 2023.

SUN, Z. et al. Undersampling method based on minority class density for imbalanced data. Expert Systems with Applications, v. 249, p. 123328, 2024. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417424001933>.

TAREKEGN, A. N.; GIACOBINI, M.; MICHALAK, K. A review of methods for imbalanced multi-label classification. Pattern Recognition, v. 118, p. 107965, 2021. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320321001527>.

TOMEK, I. Two modifications of cnn. IEEE Transactions on Systems, Man, and Cybernetics, SMC-6, n. 11, p. 769–772, 1976.

UDDIN, N. et al. An ensemble machine learning based bank loan approval predictions system with a smart application. International Journal of Cognitive Computing in Engineering, v. 4, p. 327–339, 2023. ISSN 2666-3074. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666307423000293>.

VAROQUAUX, G. et al. Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. NeuroImage, Elsevier, v. 145, p. 166–179, 2017.

VELARDE, G. et al. Tree boosting methods for balanced and imbalanced classification and their robustness over time in risk assessment. Intelligent Systems with Applications, v. 22, p. 200354, 2024. ISSN 2667-3053. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667305324000309>.

VU, L.; BUI, C. T.; NGUYEN, Q. U. A deep learning based method for handling imbalanced problem in network traffic classification. In: Proceedings of the 8th international symposium on information and communication technology. [S.l.: s.n.], 2017. p. 333–339.

WANG, A. X. et al. Challenges and opportunities of generative models on tabular data. Applied Soft Computing, p. 112223, 2024. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494624009979>.

WANG, J.; PEREZ, L. et al. The effectiveness of data augmentation in image classification using deep learning. Convolutional Neural Networks Vis. Recognit, v. 11, n. 2017, p. 1–8, 2017.

WANG, W. et al. Malware traffic classification using convolutional neural network for representation learning. In: IEEE. 2017 International conference on information networking (ICOIN). [S.l.], 2017. p. 712–717.

WANG, Z. et al. Flowgan: Unbalanced network encrypted traffic identification method based on gan. In: IEEE. 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). [S.l.], 2019. p. 975–983.

WEN, Y. et al. Improving molecular machine learning through adaptive subsampling with active learning††electronic supplementary information (esi) available. see doi: https://doi.org/10.1039/d3dd00037k. Digital Discovery, v. 2, n. 4, p. 1134–1142, 2023. ISSN 2635-098X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2635098X23000281>.

Wilson, D. L. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, SMC-2, n. 3, p. 408–421, 1972.

XIAO, J. et al. A novel deep ensemble model for imbalanced credit scoring in internet finance. International Journal of Forecasting, v. 40, n. 1, p. 348–372, 2024. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207023000353>.

XU, L. et al. Modeling tabular data using conditional gan. In: WALLACH, H. et al. (Ed.). Advances in Neural Information Processing Systems. Curran Associates, Inc., 2019. v. 32. Disponível em: $<https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf>$.

XU, L. et al. Modeling tabular data using conditional gan. arxiv 2019. arXiv preprint arXiv:1907.00503, v. 1, 2019.

XU, L.; VEERAMACHANENI, K. Synthesizing tabular data using generative adversarial networks. arXiv preprint arXiv: 1811.11264, 2018.

XU, Z. et al. Class-weighted classification: Trade-offs and robust approaches. In: PMLR. International conference on machine learning. [S.l.], 2020. p. 10544–10554.

YACOUBY, R.; AXMAN, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In: Proceedings of the first workshop on evaluation and comparison of NLP systems. [S.l.: s.n.], 2020. p. 79–91.

YANG, X. et al. Amdo: An over-sampling technique for multi-class imbalanced problems. IEEE Transactions on Knowledge and Data Engineering, v. 30, p. 1672–1685, 2018.

YEN, S.-J.; LEE, Y.-S. Cluster-based under-sampling approaches for imbalanced data distributions. Expert Systems with Applications, Elsevier, v. 36, n. 3, p. 5718–5727, 2009.

YILMAZ, I.; MASUM, R.; SIRAJ, A. Addressing imbalanced data problem with generative adversarial network for intrusion detection. In: IEEE. 2020 IEEE 21st international conference on information reuse and integration for data science (IRI). [S.l.], 2020. p. 25–30.

YUAN, Y. et al. Review of resampling techniques for the treatment of imbalanced industrial data classification in equipment condition monitoring. Engineering Applications of Artificial Intelligence, Elsevier, v. 126, p. 106911, 2023.

ZHANG, L.; ZHANG, D. Evolutionary cost-sensitive extreme learning machine. IEEE Transactions on Neural Networks and Learning Systems, v. 28, n. 12, p. 3045–3060, 2017.

ZHAO, Z. et al. Ctab-gan: Effective table data synthesizing. In: PMLR. Asian Conference on Machine Learning. [S.l.], 2021. p. 97–112.

ZHAO, Z. et al. Ctab-gan+: Enhancing tabular data synthesis. Frontiers in big Data, Frontiers Media SA, v. 6, p. 1296508, 2024.

ZHOU, Z.-H.; FENG, J. Deep forest. National science review, Oxford University Press, v. 6, n. 1, p. 74–86, 2019.

ZHU, H. et al. Nus: Noisy-sample-removed undersampling scheme for imbalanced classification and application to credit card fraud detection. IEEE Transactions on Computational Social Systems, IEEE, 2023.

ZHU, R.; GUO, Y.; XUE, J.-H. Adjusting the imbalance ratio by the dimensionality of imbalanced data. Pattern Recognition Letters, v. 133, p. 217–223, 2020. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167865520300829>.

ZHU, T.; LIN, Y.; LIU, Y. Synthetic minority oversampling technique for multiclass imbalance problems. Pattern Recognition, v. 72, p. 327–340, 2017. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320317302947>.

APÊNDICE A – CARACTERÍSTICAS DOS CONJUNTOS DE DADOS

A Tabela A apresenta as 48 (quarenta e oito) bases de dados com suas respectivas características.

Tabela A: Descrição das bases de dados em relação a quantidade e tipos de instâncias, atributos, classes e IR. Inst= Número de Instâncias, TamInst= Classificação da base quanto ao número de Instâncias (P, M, G), Atributos = Número de Atributos, TamAtrib= Classificação da base quanto ao número de atributos (P, M, G), I=Atributo Inteiro, R=Atributo Real, B=Atributo booleano e C=Categórico, Class = Numérica ou Categórica, Tipo = B (classe binária) e M (multiclasse), IR = Proporção de desbalanceamento e TamIR = Classificação da base quanto ao IR (P, M, G)

						Tipos de Atributos							
N	Base de dados	Inst	TamInst	Atrib	TamAtrib	I	R	B	C	Class	Tipo	IR	TamIR
1	abalone	502	P	9	P	2	7	0	0	N	B	32,47	G
2	Absenteeism	740	P	21	G	20	0	0	1	N	M	9,55	M
3	bank	4521	G	17	G	7	0	0	10	C	B	7,68	M
4	bank_Personal_Loan_Modelling	5000	G	14	M	13	1	0	0	N	B	9,42	M
5	BankNote	1372	M	5	P	1	4	0	0	N	B	1,25	P
6	Bankrupt	6819	G	96	G	1	95	0	0	N	B	30,00	G
7	blood	748	P	5	P	5	0	0	0	N	B	3,20	M
8	bmi	500	P	4	P	3	0	0	1	N	M	15,23	G
9	breast_coimbra	116	P	10	M	3	7	0	0	N	B	1,23	P
10	breast-cancer	569	P	32	G	1	30	0	1	N	B	1,68	P
11	bupa	345	P	7	P	6	1	0	0	N	B	1,38	P
12	cerebral-stroke	43400	G	12	M	4	3	0	5	N	B	54,43	G
13	cmc	1473	M	10	M	10	0	0	0	N	M	1,89	P
14	column_3C	310	P	7	P	0	6	0	1	N	M	2,50	M
15	ctg	2126	M	21	G	21	0	0	0	N	M	10,92	M
16	diabetes	768	P	9	P	1	8	0	0	N	B	1,87	P
17	Dry_Bean	13611	G	17	G	3	14	0	0	N	M	6,79	M
18	EEG Eye State	14980	G	15	M	1	14	0	0	N	B	1,23	P
19	employee	32769	G	10	M	10	0	0	0	N	B	16,27	G
20	financial-distress	3672	M	86	G	5	81	0	0	N	M	7,70	M
21	german_credit	1000	M	10	M	5	0	0	5	C	M	28,08	G
22	haberman	306	P	4	P	4	0	0	0	N	B	2,78	M
23	hayes-roth	132	P	6	P	6	0	0	0	N	M	1,70	P
24	house-votes-84	435	P	17	G	0	0	0	17	C	B	1,59	P
25	IT_customer_churn	7043	G	20	G	2	1	0	17	C	B	2,77	M
26	KC2	522	P	22	G	4	17	0	1	N	B	3,88	M
27	led7digit	443	P	8	P	1	7	0	0	N	B	10,97	M
28	loan_approval	4269	G	13	M	13	0	0	0	N	B	1,65	P
29	loan_data	9578	G	14	M	7	6	0	1	N	B	5,25	M
30	Maternal Health Risk	1014	M	7	P	4	2	0	1	N	M	1,49	P
31	microcalcification	11183	G	7	P	0	6	0	1	N	B	42,01	G
32	page-blocks	5473	G	11	M	7	4	0	0	N	M	175,46	G
33	pc1	1109	M	22	G	4	17	1	0	N	B	13,40	M
34	pima-indians-diabetes	768	P	9	P	7	2	0	0	N	B	1,87	P
35	poker-9_vs_7	244	P	11	M	1	10	0	0	N	B	29,50	G
36	risk_factors_cervical_cancer	858	M	36	G	10	0	0	26	C	B	14,60	G
37	Room Occupancy	2665	M	6	P	1	5	0	0	N	B	1,74	P
38	secondary_data	61069	G	21	G	0	3	0	18	C	B	1,25	P
39	sepsis	110204	G	4	P	4	0	0	0	N	B	12,60	M
40	survey lung cancer	309	P	16	M	14	0	0	2	N	B	6,92	M
41	TCGA_InfoWithGrade	839	M	24	G	23	1	0	0	N	B	1,38	P
42	ticdata2000	5822	G	86	G	86	0	0	0	N	B	15,73	G
43	titanic	1309	M	8	P	6	0	0	2	N	M	2,56	M
44	vowel0	988	M	14	M	1	13	0	0	N	B	9,98	M
45	WA-HR-Employee-Attrition	1470	M	35	G	26	0	0	9	N	B	5,20	M
46	winequality-red	1599	M	12	M	1	11	0	0	N	M	68,10	G
47	winequality-white	4898	G	12	M	1	10	0	1	N	M	439,60	G
48	yeast	1484	M	9	P	0	8	0	1	N	M	92,60	G

*Esta tabela está ordenada, de forma crescente, pelo nome dos conjuntos de dados.

Realizamos uma análise detalhada das características dos conjuntos de dados utilizados para avaliação do DSTO-GAN, apresentados na Tabela A. Para uma compreensão mais aprofundada da distribuição dessas características, foram elaboradas as Figuras 41 (a), (b)

e (c), que consistem em histogramas com curvas de densidade para três métricas principais: número de atributos, número de instâncias e índice de desbalanceamento (IR).

Essas figuras permitem visualizar a variabilidade e a concentração dos valores para cada métrica, fornecendo informações sobre a heterogeneidade dos conjuntos de dados selecionados. A análise dessas distribuições é fundamental para contextualizar o desempenho do DSTO-GAN em diferentes cenários, destacando sua capacidade de lidar com conjuntos de dados de variados tamanhos, complexidades e níveis de desbalanceamento.

Figura 41: Histograma - Quantidade de atributos, instância e IR

Fonte: Dados da Pesquisa.

A Figura 41(a) mostra um histograma da quantidade de atributos, revelando uma distribuição assimétrica com uma concentração significativa de instâncias com poucos atributos. A maioria dos dados possui menos de 50 atributos, com um pico entre 0 e 50. Há uma cauda longa para a direita, indicando a presença de algumas instâncias com um número muito maior de atributos, chegando a 300. A distribuição sugere que a maioria dos conjuntos de dados neste estudo possui uma quantidade relativamente baixa de atributos.

A Figura 41(b) exibe um histograma da quantidade de instâncias, mostrando uma distribuição também assimétrica, com a maioria dos dados concentrados em faixas menores. A maior parte dos dados possui menos de 20.000 instâncias, com um pico entre 0 e 20.000. Assim como no gráfico anterior, há uma cauda longa para a direita, com alguns casos isolados com mais de 80.000 instâncias.

A Figura 42(a) revela uma distribuição aproximadamente uniforme entre a quantidade de instâncias com as categorias P, M e G, com as seguintes porcentagens, respectivamente, 35,42%, 33,33% e 31,25%.

A Figura 41(c) ilustra a distribuição do grau de desbalanceamento (*Imbalance Ratio*-IR), mostrando uma assimetria semelhante aos gráficos anteriores. A maioria dos dados

apresenta um IR baixo, indicando um certo equilíbrio entre as classes. O pico está entre 1.2 e 11 de IR, e há uma cauda longa para a direita, com alguns casos atingindo um IR superior a 400. A distribuição sugere que a maioria dos conjuntos de dados não possui um desbalanceamento extremo entre as classes. No entanto, a presença de alguns casos com IR muito alto indica um desbalanceamento significativo, o que pode ser um desafio para algoritmos de aprendizado de máquina, pois estes tendem a ser enviesados para a classe majoritária, prejudicando a capacidade de generalização do modelo para a classe minoritária, que geralmente é a de maior interesse. Além disso, métodos de balanceamento, como *oversampling* ou *undersampling*, podem se tornar complexos e menos eficazes em cenários de IR extremo, exigindo técnicas mais sofisticadas ou combinadas, como SMOTE ou ADASYN, para lidar com a disparidade entre as classes sem comprometer a qualidade dos dados.

Para uma análise mais estruturada, as características dos conjuntos de dados — instâncias, atributos e índice de desbalanceamento (IR) — foram classificadas em três categorias: pequeno (P), médio (M) e grande (G). Essa classificação foi baseada no cálculo dos percentis, que dividem os dados em três grupos de tamanho equivalente. A seguir, são apresentados os intervalos definidos para cada categoria:

Instâncias: P (116 a 748), M (749 a 2665) e G (2666 a 110204)

Atributos: P (4 a 9), M (10 a 20) e G (21 a 309)

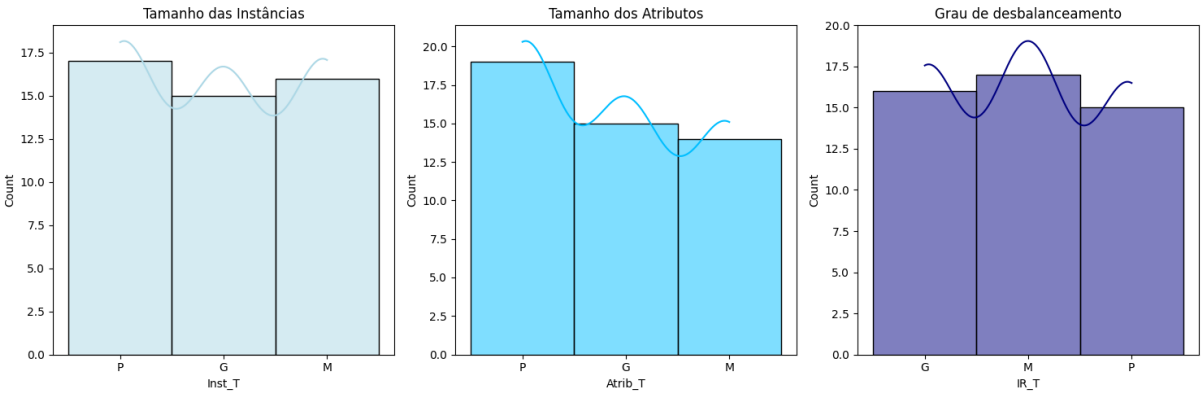
IR: P (1.2 a 2), M (3 a 11) e G (12 a 439)

Em relação a quantidade de atributos, a Figura 42(b) mostra a seguinte distribuição: (P) com 39,58%, (M) 29,17% e (G) com 31,25%. A categoria P possui a maior contagem de atributos, seguida pela categoria G e, por fim, a categoria M com a menor contagem. Isso indica que o conjunto de dados possui uma variedade de atributos, com alguns casos apresentando um número significativamente maior do que outros.

A Figura 42(c) mostra a distribuição do IR: (P) 31,25%, (M) 35,42% e (G) 33,33%. A categoria G apresenta o maior grau de desbalanceamento, seguida pela categoria M e, por fim, a categoria P com o menor grau de desbalanceamento. Isso sugere que o conjunto de dados possui um desequilíbrio entre as classes, com algumas classes sendo muito mais frequentes do que outras.

Além da classificação da base quanto ao número de atributos, instâncias e IR, foi realizada também uma classificação quanto às características dos atributos, se são nominais ou numéricos, além do tipo de classificação, se a classificação é binária ou multiclasse. Neste sentido, utilizamos o seguinte protocolo de classificação, se pelo menos (metade + 1) dos atributos for de um tipo de dados, a *base de dados* será desse tipo. Por exemplo: se a base de dados possui 30 atributos, e 17 atributos são nominais, então é classificada

Figura 42: Distribuição Tamanho - Atributos, Instâncias e IR



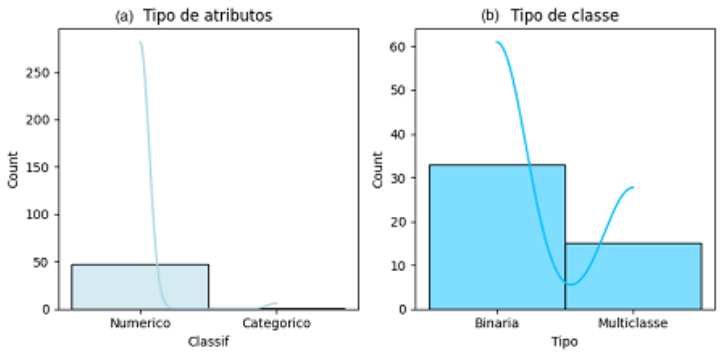
Fonte: Dados da Pesquisa.

como categórica. Os detalhes das bases de dados são fornecidos na Tabela A, tais como a quantidade de instâncias, atributos, os tipos Inteiro (I), Real (R), Booleano (B) e Categóricos (C). A base mista teria a mesma quantidade da atributos categóricos e numéricos, nos nossos conjunto de dados não tivemos nenhuma base mista.

Em relação ao tipo de dados, 97,96% dos conjuntos de dados são numéricos e 2,04% são categóricos, veja Figura 43(a). A predominância de atributos numéricos sugere que algoritmos como Árvores de Decisão e Random Forest podem ser adequados para modelar os dados.

A maioria dos conjuntos de dados possui classes binárias 69,39%, o que é comum em problemas de classificação e 30,61% são multiclasse (M), conforme Figura 43(b). A presença de classificação multiclasse indica que há conjuntos de dados com problemas de classificação mais complexos.

Figura 43: Tipo de atributos e tipo de classe



Fonte: Dados de pesquisa.

APÊNDICE B – AVALIAÇÃO DO MÉTODO PROPOSTO (PRECISÃO E REVOCAÇÃO)

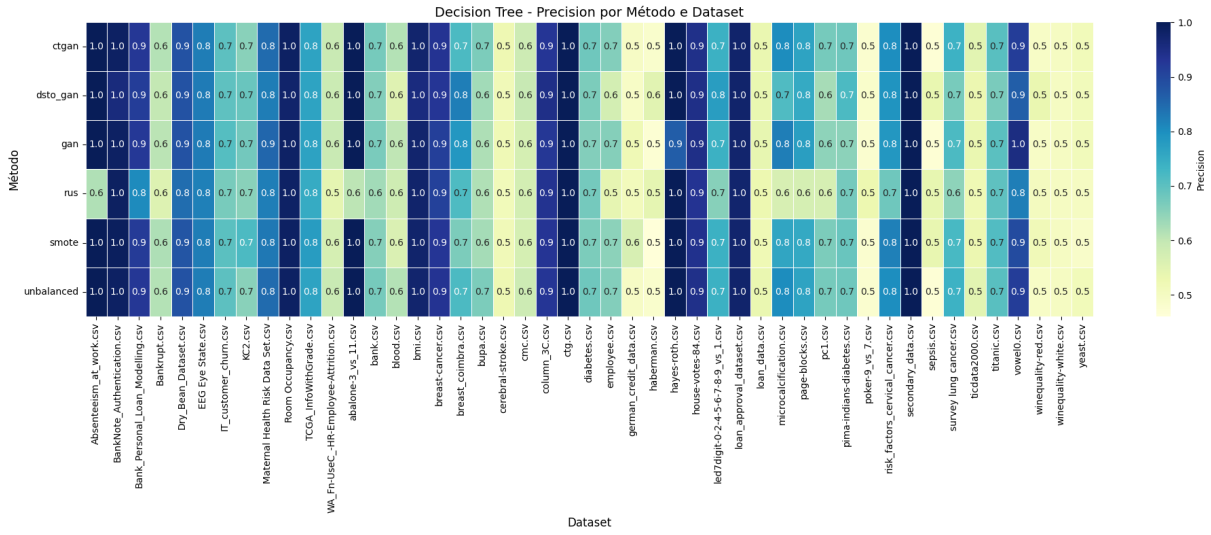
Este capítulo apresenta a avaliação do DSTO-GAN quanto às métricas de Precisão e Revocação, complementando os resultados da Capítulo 6.

B.1 Avaliação do DSTO-GAN em Relação a Métodos de Balanceamento e Classificadores

.

A Figura 44 ilustra o desempenho da Precisão para diversas combinações de conjuntos de dados, algoritmos de classificação e técnicas de balanceamento.

Figura 44: Precisão: Desempenho dos métodos de balanceamento e classificadores



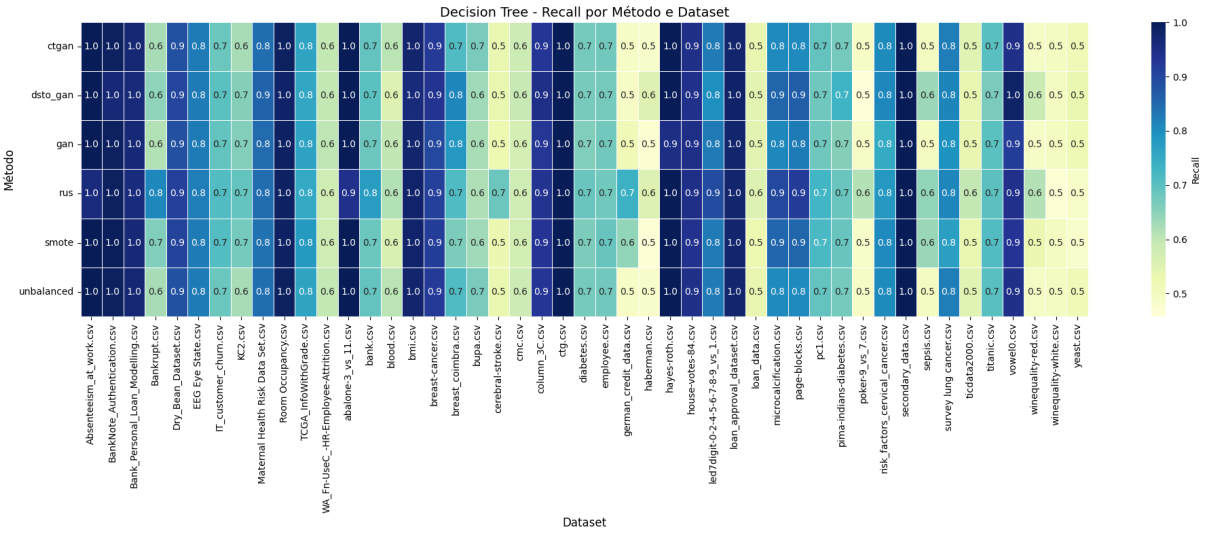
Fonte: Dados da Pesquisa.

A Figura 45 exibe o desempenho do Revocação para diversas combinações de conjuntos de dados, algoritmos de classificação e técnicas de balanceamento.

B.2 Desempenho dos Métodos de Balanceamento em Relação a Quantidade de Instâncias do Conjunto de Dados

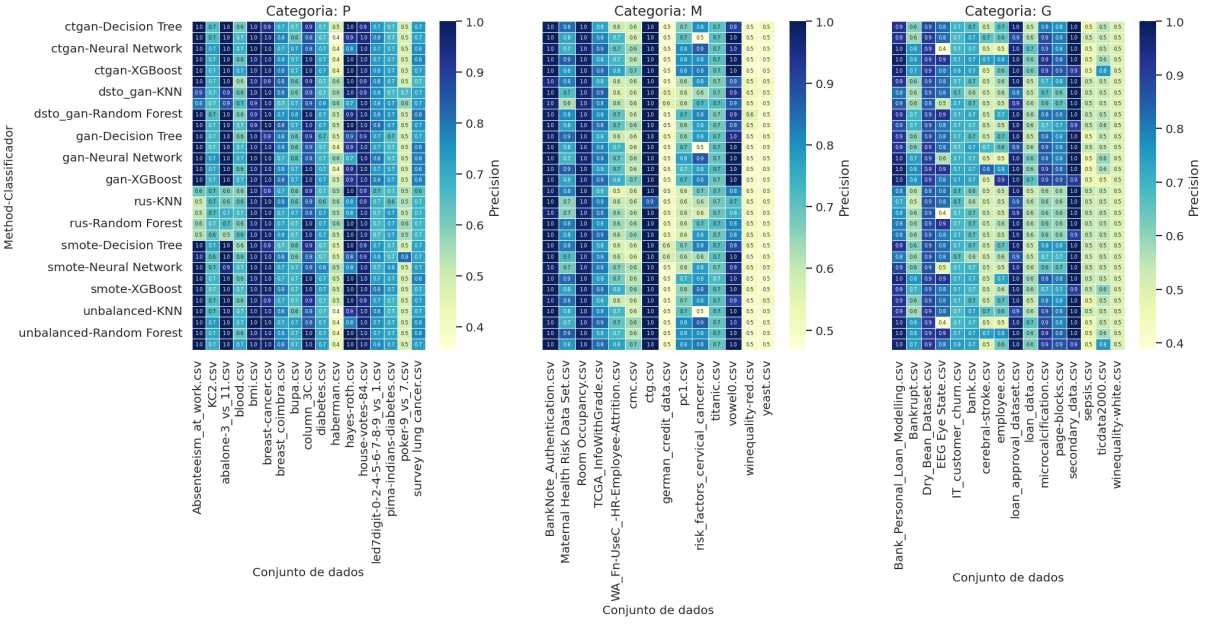
A precisão foi consistentemente alta em conjuntos pequenos e médios, especialmente com *Random Forest* e *XGBoost*. No entanto, em conjuntos como *winequality-red* e *yeast*, a precisão caiu devido ao desbalanceamento extremo, Figura 46.

Figura 45: Revocação: Desempenho dos Métodos de Balanceamento e Classificadores



Fonte: Dados da Pesquisa.

Figura 46: Precisão e Tamanho da Amostra no Balanceamento de Dados



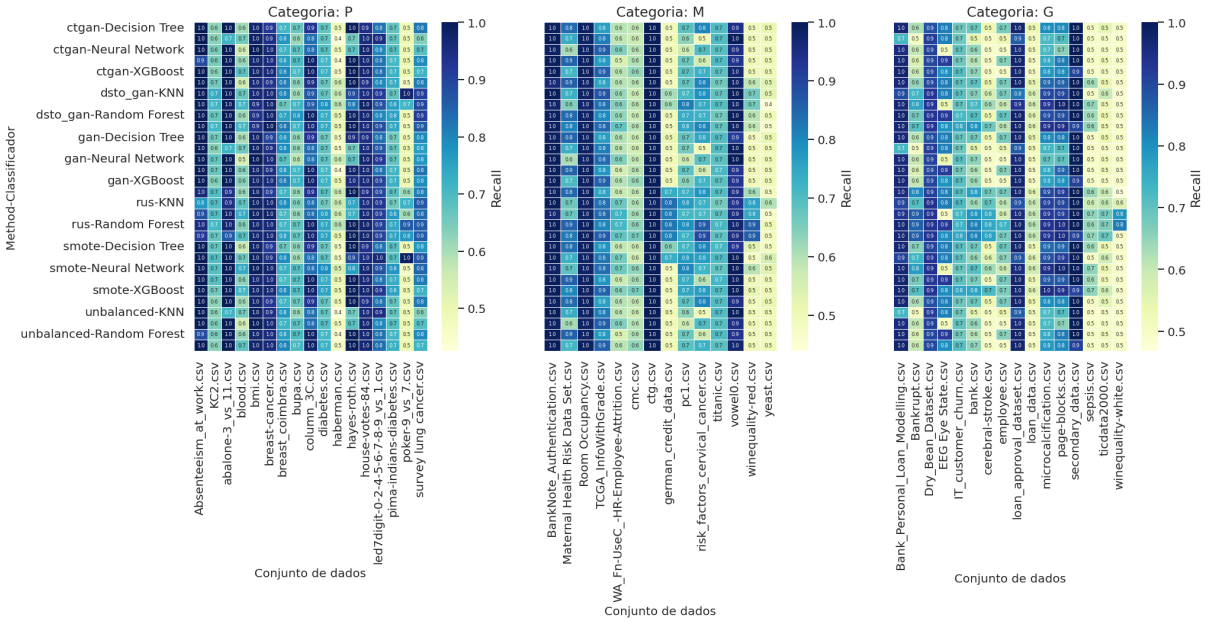
Fonte: Dados da Pesquisa.

B.3 Avaliação do Desempenho de Métodos de Balanceamento em Relação a Dimensionalidade dos Conjuntos de Dados

A precisão, que mede a acurácia das previsões positivas, também variou conforme o classificador e o tamanho dos atributos, Figura 23.

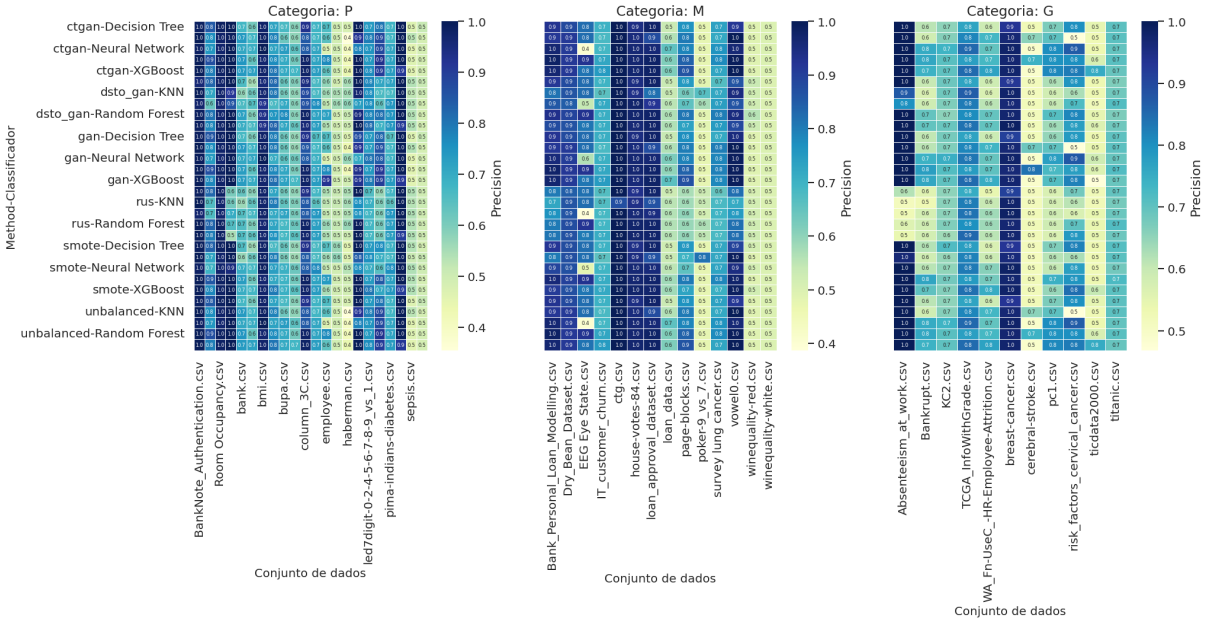
O Revocação, que mede a capacidade de identificar corretamente as classes positivas,

Figura 47: Revocação: Impacto do Tamanho da Amostra no Balanceamento de Dados



Fonte: Dados da Pesquisa.

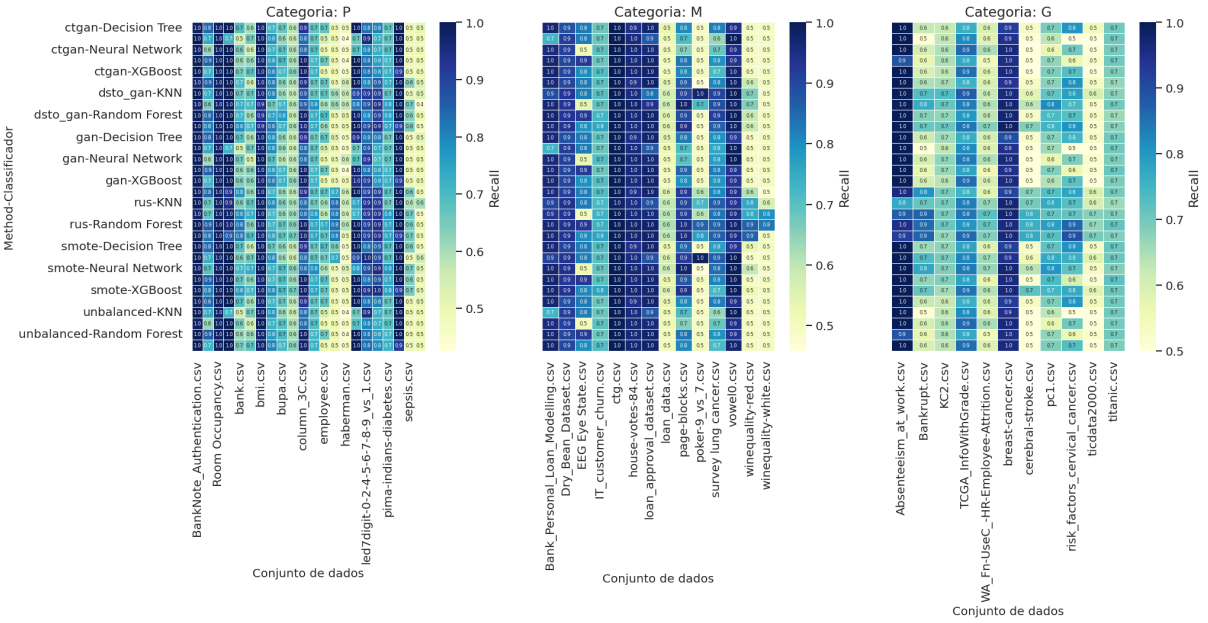
Figura 48: Precisão: Impacto da Dimensionalidade no Balanceamento de Dados



Fonte: Dados da Pesquisa.

seguiu padrão semelhante às outras métricas, Figura 49:

Figura 49: Revocação: Impacto da Dimensionalidade no Balanceamento de Dados



Fonte: Dados da Pesquisa.

B.4 Avaliação do Desempenho de Métodos de Balanceamento em Conjunto de Dados com Diferentes Níveis Desbalanceamento

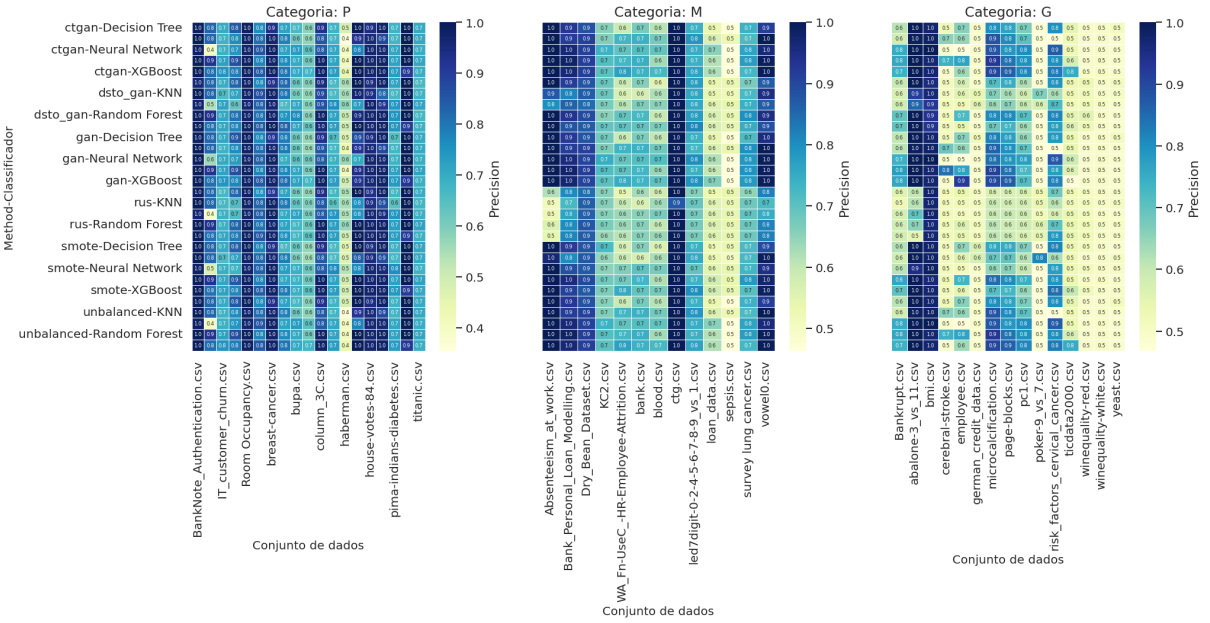
Avaliamos a influência de diferentes níveis de desbalanceamento de classes (IR) no desempenho de classificadores, em conjunto com a aplicação de métodos de balanceamento. Para tanto, os conjuntos de dados foram categorizados em três grupos distintos, designados como P, M e G, de acordo com seus respectivos níveis de IR.

A análise do desempenho dos classificadores foi realizada através da avaliação das métricas Precisão (Figura 50) e Revocação (Figura 51).

B.5 Avaliação do Desempenho dos Métodos de Balanceamento em Relação a Conjunto de Dados Binários ou Multiclasse

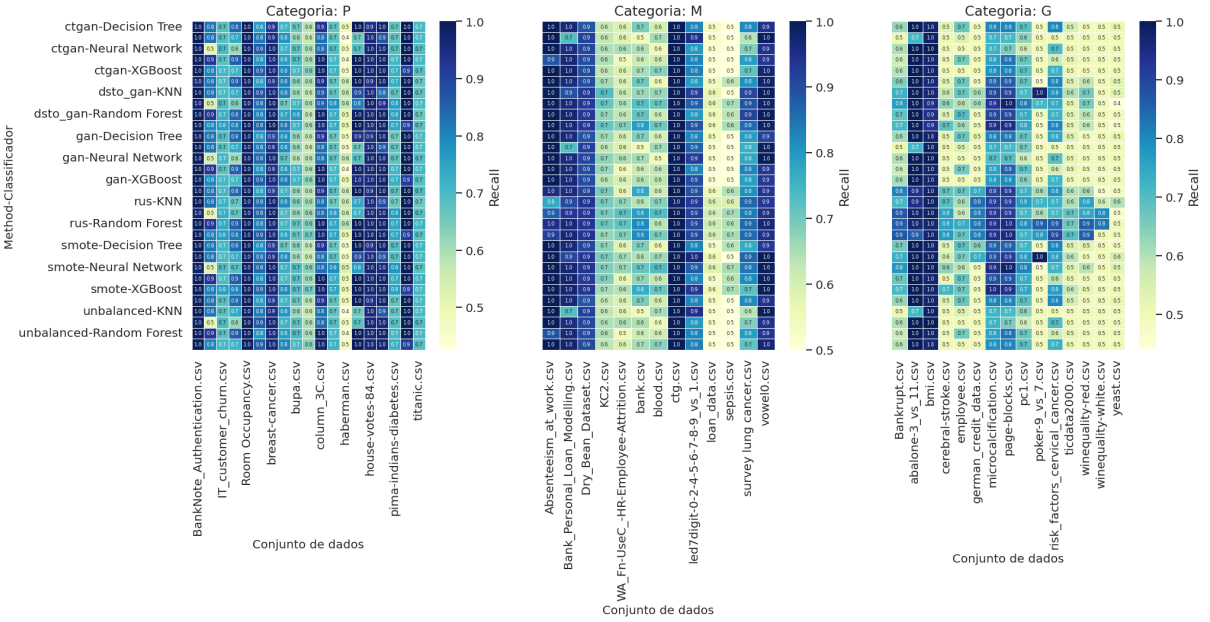
Avaliamos o impacto de diferentes métodos de balanceamento (CTGAN, DCGAN, DSTO-GAN, RUS, SMOTE) combinados com classificadores (*Decision Tree*, KNN, *Neural Network*, *Random Forest* e *XGBoost*) no desempenho de modelos de classificação para o atributo "Tipo" em duas categorias: Binária e Multiclasse. As métricas de avaliação utilizadas são Precisão (Figura 52) e Revocação (Figura 53). O foco principal desta análise é o método DSTO-GAN.

Figura 50: Precisão: Impacto do Desbalanceamento no Balanceamento de Dados



Fonte: Dados da Pesquisa.

Figura 51: Revocação: Impacto do Desbalanceamento no Balanceamento de Dados

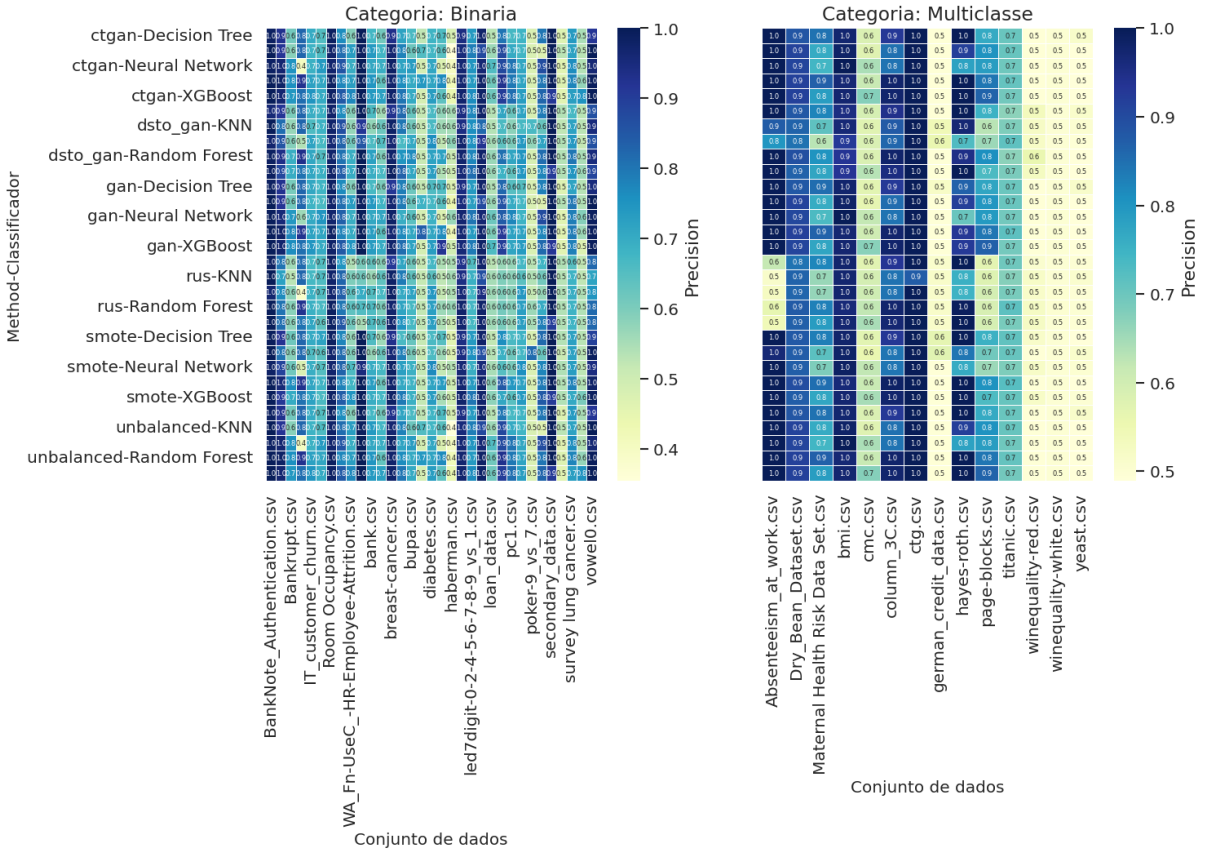


Fonte: Dados da Pesquisa.

B.6 Avaliação do Desempenho dos Métodos de Balanceamento em Relação aos Tipos de Atributos do Conjunto de Dados

Avaliamos o desempenho dos métodos de balanceamento em relação ao tipo de atributos: categóricos e numéricos, Figuras 54 e 55. Os conjunto de dados identificados com

Figura 52: Precisão: desempenho dos métodos de balanceamento em relação aos tipos de classe



Fonte: Dados da Pesquisa.

categóricos representam apenas 2% do total.

B.7 Análise de Sensibilidade dos Hiperparâmetros do Método DSTO-GAN

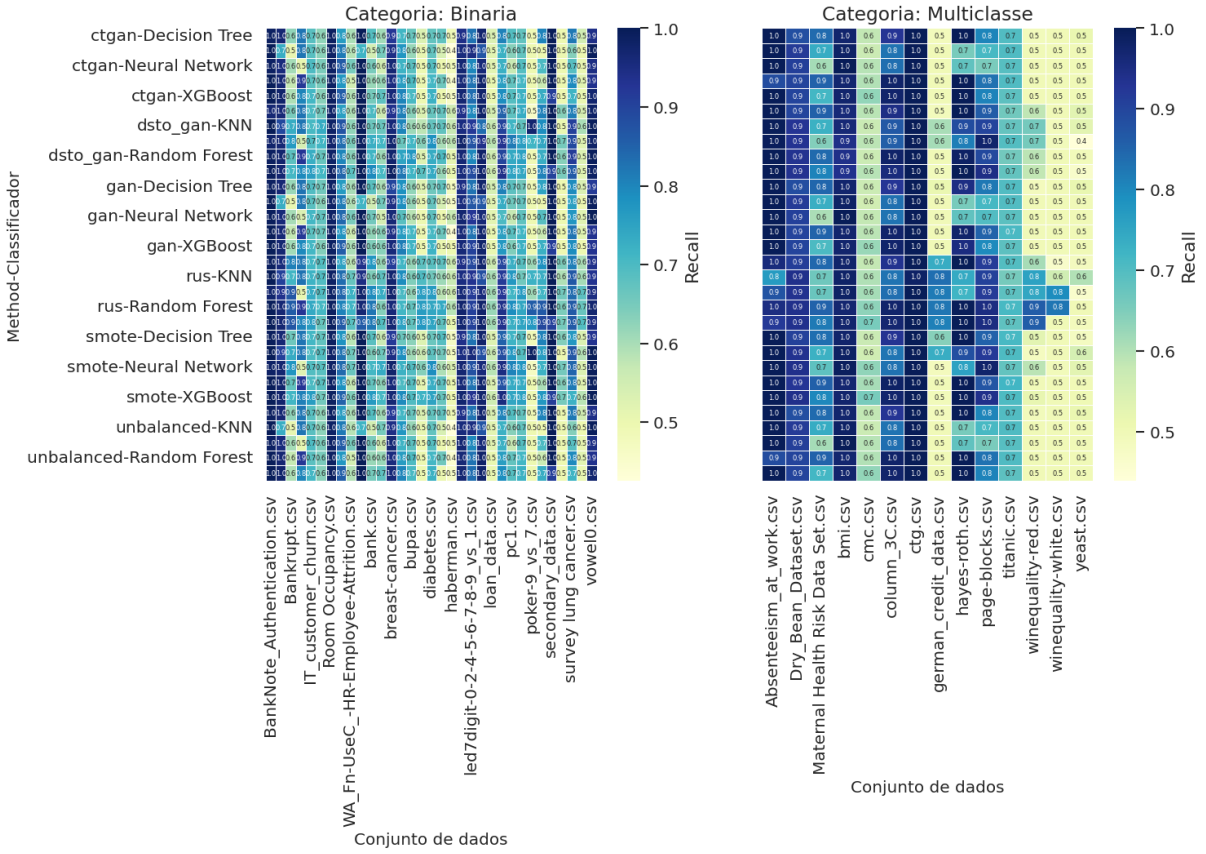
O teste de sensibilidade de hiperparâmetros constitui uma metodologia sistemática, aplicada na análise quantitativa que avalia o impacto das variações nas configurações de hiperparâmetros sobre o desempenho de modelos de aprendizado de máquina. Avaliamos os seguintes hiperparâmetros do método DSTO-GAN, Tabela 13.

Tabela 13: Teste de sensibilidade: Hiperparâmetros do DSTO-GAN

Hiperparâmetros	**Significado**	**Valores Testados**
dim_h	Tamanho da camada oculta	(32, 64)
n_z	Dimensão do espaço latente	(5, 10)
lr	Taxa de aprendizado	(0.0002, 0.001)

A melhor Configuração de Parâmetros do GAN é $dim\_h = 32$, $n\_z = 5$, $lr = 0.001$. Esta combinação aparece consistentemente em todos os melhores resultados,

Figura 53: F1-Score: desempenho dos métodos de balanceamento em relação aos tipos de classe



Fonte: Dados da Pesquisa.

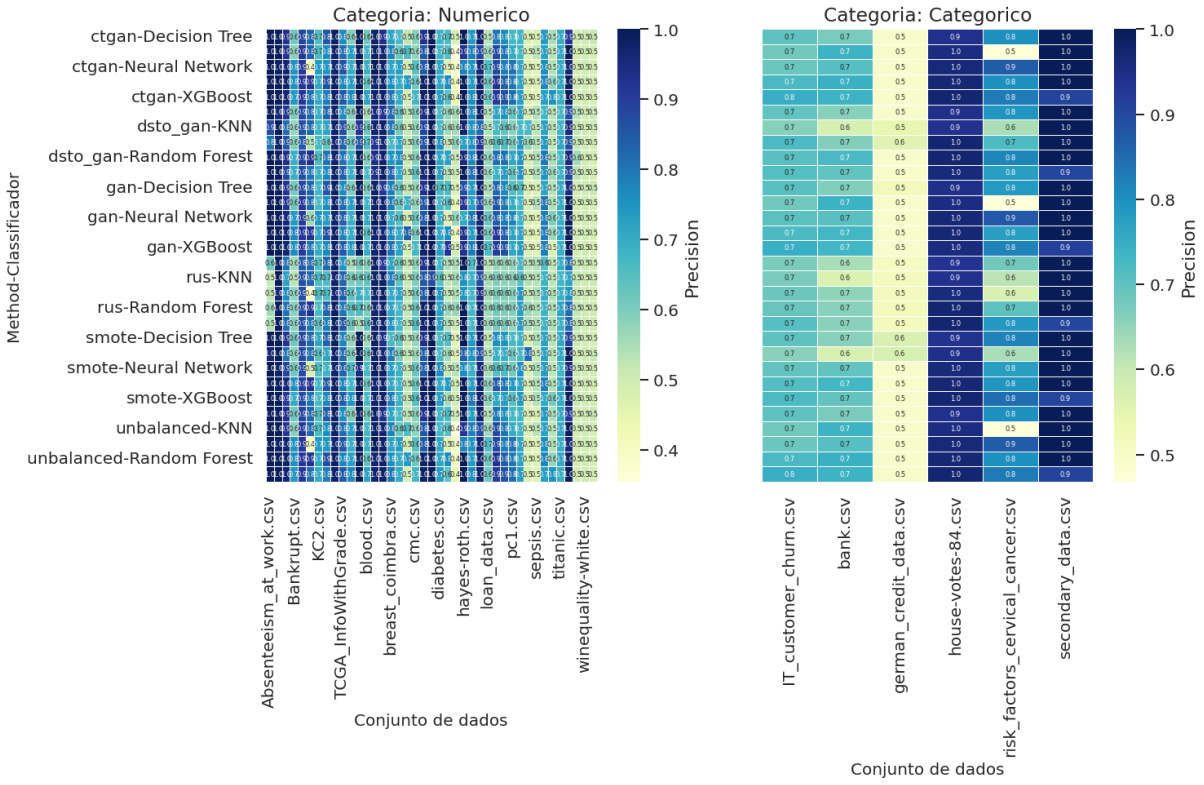
independentemente do tamanho do conjunto de dados ou do classificador, Figuras 56 e 57.

Valores menores de $dim\_h = 32$ e $n\_z = 5$ parecem ser suficientes para capturar os padrões dos dados sem causar *overfitting*, Figuras 56 e 57.

A taxa de aprendizado ($lr = 0.001$) mostrou-se eficaz para convergência estável, sendo pequena o suficiente para evitar oscilações, mas grande o suficiente para treinamento eficiente, Figuras 56 e 57.

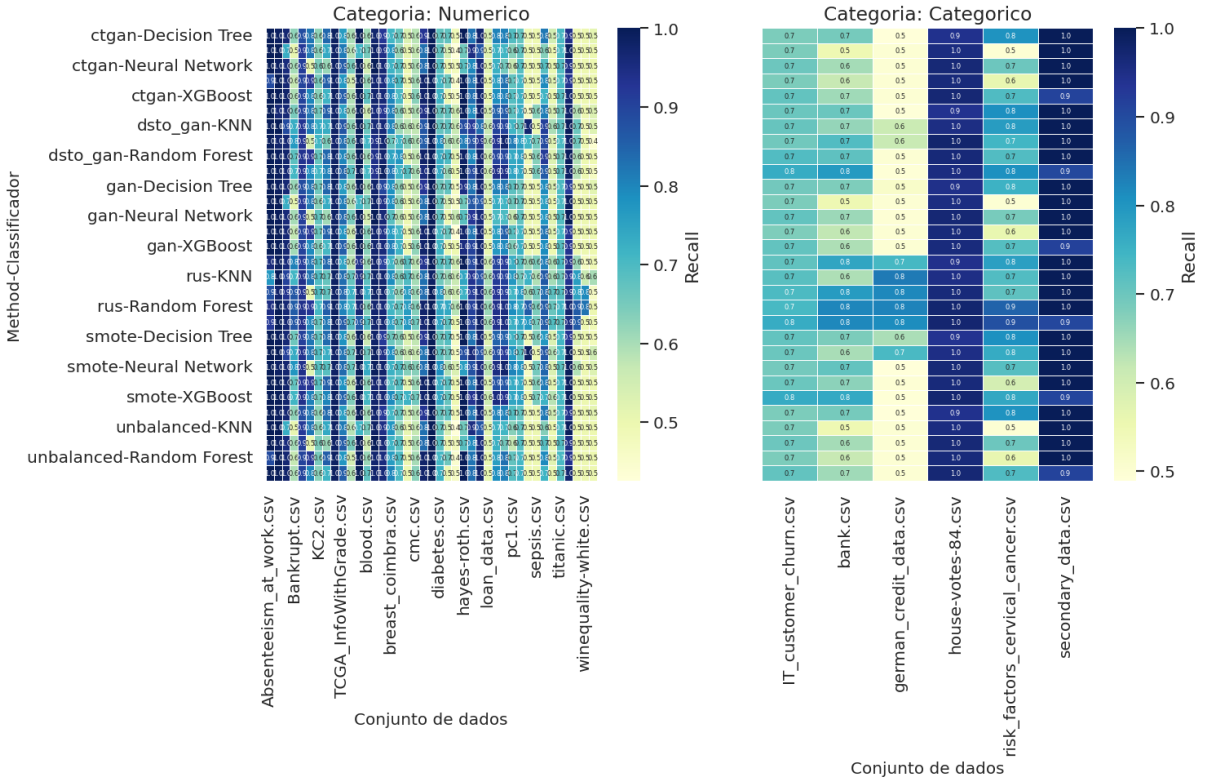
Configurações mais complexas (maiores valores de dim_h e n_z) não melhoraram o desempenho, apenas aumentariam o custo computacional, Figuras 56 e 57.

Figura 54: Precisão: desempenho dos métodos de balanceamento em relação aos tipos de atributos



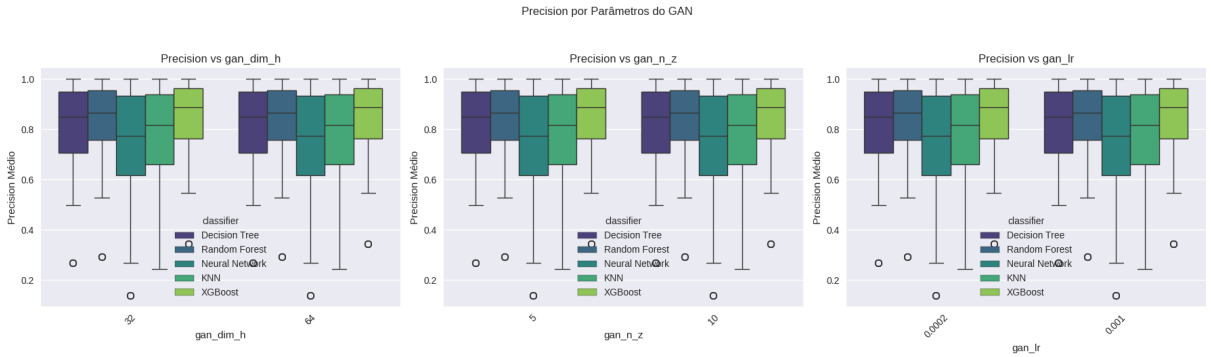
Fonte: Dados da Pesquisa.

Figura 55: Revocação: desempenho dos métodos de balanceamento em relação aos tipos de atributos



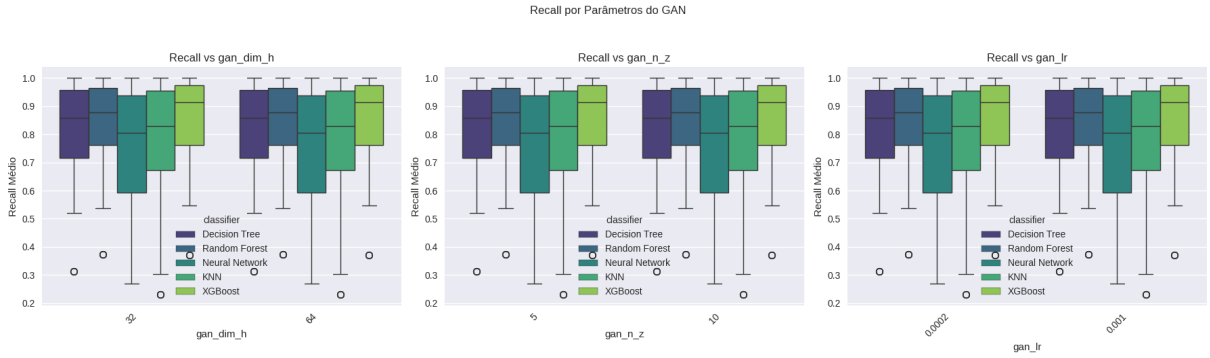
Fonte: Dados da Pesquisa.

Figura 56: Precisão: Teste de sensibilidade dos Hiperparâmetros DSTO-GAN



Fonte: Dados da Pesquisa.

Figura 57: Revocação: Teste de sensibilidade dos Hiperparâmetros DSTO-GAN



Fonte: Dados da Pesquisa.