UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Eric Luiz Rodrigues de França

Avaliação de Detecção de Anomalias baseado em Teoria de Resposta ao Item e Modelo de Concordância

Recife

2025

Eric Luiz Rodrigues de França

Avaliação de Detecção de Anomalias baseado em Teoria de Resposta ao Item e Modelo de Concordância

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador (a): Prof. Dr. Ricardo Bastos Cavalcante Prudêncio

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

França, Eric Luiz Rodrigues de.
Avaliação de detecção de anomalias baseado em Teoria de Resposta ao Item e Modelo de Concordância / Eric Luiz Rodrigues de França. - Recife, 2025.
113f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-graduação em Ciência da Computação, 2025.
Orientação: Bastos Cavalcante Prudêncio.
Inclui referências, apêndices e anexos.

1. Método de Detecção de Anomalias; 2. Avaliação Não Supervisionada; 3. Modelo de Concordância; 4. Teoria de Resposta ao Item. I. Prudêncio, Ricardo Bastos Cavalcante. II. Título.

UFPE-Biblioteca Central

Eric Luiz Rodrigues de França

“Avaliação de Detecção de Anomalias baseado em Teoria de Resposta ao Item e Modelo de Concordância”

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovado em: 25/02/2025.

BANCA EXAMINADORA

_____

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática / UFPE

_____

Prof. Dr. Ronnie Cley de Oliveira Alves
Instituto de Ciências Exatas e Naturais / UFPA

_____

Prof. Dr. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE
(**orientador**)

Dedico este trabalho ao meu querido Tio Digue, Dona Ivone, Niedja e Seu Francisco, que sempre farão parte de minha família. Também registro aqui, minha gratidão à todo(a)s amigos, colegas e conhecidos que, embora não cite aqui os nomes de cada um dele(a)s, tive igualmente a oportunidade de conhecê-los ainda em vida, até a presente década de 2020.

AGRADECIMENTOS

Primeiramente agradeço ao nosso Eterno, Único e Altíssimo, por proteger com seu Amor.

Agradeço aos meus pais, que me proporcionaram as condições essênciais para atingir a vida adulta. Em especial à minha mãe, por todo seu apoio, bem como incentivo na vida acadêmica.

Agradeço a minha família, base sólida que firmou minha compreensão do mundo. Em especial à minha avó garanhuense, por todo seu suporte, bem como ensinamentos das tradições e valores do passado.

Também venho agradecer ao meu professor orientador deste mestrado acadêmico, que acreditou no meu trabalho, teve compreensão e paciência na minha evolução durante este projeto. Sua orientação e seu voto de confiança, foram fundamentais para o desenvolvimento desta dissertação, de minha formação acadêmica, e também profissional.

Aos professores do Centro de Informática (Cin), de todas as disciplinas que cursei neste mestrado acadêmico, minha gratidão por compartilhar conhecimentos, experiências, e referências nas mais diversas áreas.

Aos colegas do Cin que contribuiram, direta ou indiretamente, para minha formação acadêmica. Principalmente nos primeiros semestres desta Pós, sou muito grato por todo apoio.

Ao Cin, meu profundo agradecimento por fornecer ensino de qualidade e acessível. Mesmo se reinventando ao longo dos últimos anos, continua sendo destaque e referência em todo o país, na sua área de atuação.

Por fim, agradeço à Universidade Federal de Pernambuco (UFPE), por estar presente em todas as fases da minha vida, com bons momentos e experiências. Embora venha enfrentando obstáculos ao longo do tempo, mantém-se firme fornecendo ensino superior público de qualidade. Além de impactar positivamente na comunidade e sociedade de toda uma região.

"O temor de Jeová é o princípio da sabedoria [...]"
(Provérbios 9:10)

RESUMO

A avaliação de detecções de anomalias ainda permanece sendo uma tarefa complexa. Métricas de avaliação supervisionadas, como o *Precision score* ou AUC, são frequentemente utilizadas para *benchmarking*, mas não são aplicáveis em cenários reais não supervisionados, devido à falta de rótulos *Ground Truth* (GT). Ainda assim, podemos recorrer à métricas não supervisionadas para avaliação de modelos, como por exemplo o Pseudo GT, ou *Excess-Mass* (EM) e *Mass-Volume* (MV). Esses critérios de avaliação têm a vantagem de comparar modelos com base nos próprios *data points* das instâncias, sem a necessidade de dados rotulados (em classes). Contudo, possuem algumas limitações em certos cenários, inclusive quando aplicados à detecção de anomalias. Enquanto isso, a Teoria de Resposta ao Item (do inglês *Item Response Theory*) (IRT) que por muito tempo, foi inicialmente empregada para avaliar as habilidades latentes de respondentes humanos, a partir de suas respectivas respostas à itens de problemas com diferentes níveis de dificuldade. Porém nos últimos anos, a comunidade científica propôs soluções que utilizam IRT em aplicações de Inteligência Artificial (do inglês *Artificial Intelligence*) (AI), como em problemas de classificação e avaliação de algoritmos. Inclusive não mais usando um modelo dicotômico de IRT (apenas para respostas certas ou erradas), mas sim já utilizando um modelo contínuo de IRT, onde suas respostas são representadas pela probabilidade de uma predição correta. Um exemplo de implementação deste modelo contínuo é o β4-IRT, o qual facilita bastante a aplicação de IRT em AI atualmente. Portanto, neste trabalho propomos o ODAIRE (*Outlier Detection Agreement-based Item REsponses*), um *workflow* para avaliação geral de modelos de detecção de *outliers*, o qual assume que os melhores modelos sempre concordam sobre quais pares de instâncias devem ser rotulados como anomalias. Neste *workflow*, as matrizes de respostas obtidas ao se calcular a concordância entre esses modelos, são usadas para estimar a habilidade dos métodos de detecção e a dificuldade das instâncias, através da aplicação do β4-IRT. Esta inovadora abordagem proposta, representa uma alternativa para avaliar o desempenho dos métodos de detecção de anomalias em cenários não supervisionados, algo não investigado em nenhum trabalho anteriormente na literatura. Somado a isto, também surge uma alternativa para a identificação das regiões num conjunto de dados pontuais, que apresentam diferentes graus de dificuldade ou discriminação.

Palavras-chaves: Método de Detecção de Anomalias. Avaliação Não Supervisionada. Modelo de Concordância. Teoria de Resposta ao Item.

ABSTRACT

The evaluation of anomaly detection remains a complex task. Supervised evaluation metrics, such as Precision score or AUC, are often used for benchmarking, but they are not applicable in real-world unsupervised scenarios due to the lack of labels (Ground Truth – GT). Nevertheless, we can resort to unsupervised metrics for model evaluation, such as Pseudo GT, or Excess-Mass (EM) and Mass-Volume (MV). These evaluation criteria have the advantage of comparing models based on the data points of the instances themselves, without the need for labeled data (i.e., class labels). However, they have certain limitations in specific scenarios, including when applied to anomaly detection.

Meanwhile, Item Response Theory (IRT), which was originally used to assess the latent abilities of human respondents based on their answers to problem items of varying difficulty levels, has in recent years been adapted by the scientific community for use in Artificial Intelligence (AI) applications, such as classification problems and algorithm evaluation. Importantly, this no longer relies solely on the dichotomous IRT model (right or wrong answers only), but instead uses a continuous IRT model, where responses are represented by the probability of a correct prediction. One example of an implementation of this continuous model is β4-IRT, which greatly facilitates the application of IRT in AI today.

Therefore, in this work, we propose ODAIRE (Outlier Detection Agreement-based Item REsponses), a workflow for the general evaluation of outlier detection models. It assumes that the best models tend to agree on which pairs of instances should be labeled as anomalies. In this workflow, response matrices obtained by calculating agreement among these models are used to estimate the ability of the detection methods and the difficulty of the instances, through the application of β4-IRT. This innovative approach represents an alternative for evaluating the performance of anomaly detection methods in unsupervised settings — something not previously explored in the literature. Additionally, it offers a new perspective for identifying regions within a dataset that exhibit varying degrees of difficulty or discrimination.

Keywords: Anomaly Detection Method. Unsupervised Evaluation. Model Agreement. Item Response Theory.

LISTA DE FIGURAS

Figura 1 – *Pipeline* da Metodologia de Validação dos métodos de Detecção de Anomalia (do inglês *Anomaly Detection*) (AD) proposta. 39

Figura 2 – Conjuntos de dados artificiais gerados sinteticamente pela ferramenta *PyOD*. Os pontos em azul representam os *inliers*, já os *outliers* estão em vermelho. 45

Figura 3 – Gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS4*. Destacando no tamanho dos pontos, as classes positivas. 67

Figura 4 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS3* 68

Figura 5 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS12* 68

Figura 6 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS14* 69

Figura 7 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS51* 69

Figura 8 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS4* (sub-conjuntos de treino e teste). . 70

Figura 9 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS3* (sub-conjuntos de treino e teste). . 71

Figura 10 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS12* (sub-conjuntos de treino e teste). . 71

Figura 11 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS14* (sub-conjuntos de treino e teste). . 72

Figura 12 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS51* (sub-conjuntos de treino e teste). . 72

Figura 13 – Gráfico de dispersão em espaço 2D representando traço latente *Discriminação* em cada instância do *DS4* (sub-conjuntos de treino e teste). 73
Figura 14 – Conjunto de dados reais públicos chamado *Pima* (proveniente da base *ODDS*) com dimensão reduzida pela técnica *Principal Component Analysis* (PCA). Os pontos em azul representam os *inliers*, já os *outliers* estão em amarelo. 85
Figura 15 – Trecho da Matriz de predições obtidas dos Métodos de AD durante treinamento no *dataset* dados reais *Pima*. 86
Figura 16 – Trecho da Matriz de *scores* obtidos nas predições dos Métodos AD ajustados no conjunto de dados reais *Pima*. 86
Figura 17 – Vetor de Habilidades obtido pelo modelo IRT ajustado no sub-conjunto dos dados de teste do dataset *DS4* com o traço latente de todos Métodos de AD utilizados como respondentes. 87
Figura 18 – Trecho do vetor Dificuldades obtido pelo modelo IRT ajustado no dataset *DS4* com o traço latente de apenas 10 instâncias do sub-conjunto dos dados de teste. 88
Figura 19 – Trecho do vetor Discriminação obtido pelo modelo IRT ajustado no dataset *DS4* com o traço latente de apenas 10 instâncias do sub-conjunto dos dados de teste. 88
Figura 20 – Trecho da matriz de métricas de avaliação de modelos versus os Modelos AD a partir das predições obtidas no *dataset* com dados reais chamado *Pima*. 89
Figura 21 – Trecho da Matriz de ranqueamento das Habilidades obtidas pelo modelo IRT a partir das concordâncias entre predições dos Métodos AD ajustados no conjunto de dados reais *Pima*. 89
Figura 22 – Gráfico do mapa de calor representando a tabela de correlações, utilizando o coeficiente de Spearman, entre os valores de ranqueamento/postos de todas as métricas utilizadas, durante avaliação de Modelos AD ajustados ao *dataset DS4*. 91
Figura 23 – Gráfico de dispersão 2D representando média ponderada de predições dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS4* 92

Figura 24 – Trecho do gráfico de dispersão em espaço 2D, com dimensão reduzida pela técnica PCA, representando traço latente *Dificuldade* em cada instância do *dataset Breastw* (sub-conjuntos de treino e teste). 92

LISTA DE QUADROS

Quadro 1 – Configuração dos parâmetros na função geradora de dados artificiais . . . 44
Quadro 2 – Trecho da Matriz de Concondância derivada das predições obtidas no conjunto de dados *DS4*. 54
Quadro 3 – Trecho da Matriz de ranqueamento das métricas obtidas nas avaliações dos Métodos AD ajustados no conjunto de dados reais *Pima*. 57
Quadro 4 – Trecho de Matriz de correlação entre Métricas de avaliação de Métodos AD ajustados no conjunto de dados artificiais *DS4*. 58
Quadro 5 – Trecho de Matriz de correlação entre Métricas de avaliação de Métodos AD ajustados no sub-conjunto de treino de dados reais *Pima*. 59
Quadro 6 – Contagem de valores distintos obtidos após cálculos das médias aritméticas a partir das predições e das concordâncias em cada uma das instâncias do *dataset DS4*. 90
Quadro 7 – Trecho da contagem de valores distintos obtidos após cálculos das médias ponderadas (possuindo Habilidade IRT como pesos) a partir das predições e das concordâncias em cada uma das instâncias do *dataset DS4* 90

LISTA DE TABELAS

Tabela 1 – Características dos *datasets* com dados reais utilizados. 46
Tabela 2 – Métodos de AD que foram adotados, agrupados por bibliotecas de implementação. 50
Tabela 3 – Correlação entre Habilidades (IRT) e outras Métricas de avaliação de modelos. 64
Tabela 4 – Principais conjuntos de dados (artificiais e reais), que apresentaram em média mais *scores* (métricas Área sob a curva ROC (do inglês *Area Under Curve ROC*) (AUC), *F-Measure*) na detecção de anomalias, bem como também aqueles com menor *score*. 65
Tabela 5 – Principais conjuntos de dados (artificiais e reais), que apresentaram em média mais Dificuldade (traço latente IRT), e também aqueles com menos Dificuldade. 66
Tabela 6 – Principais algoritmos de AD, que apresentaram em média mais Habilidade (traço latente IRT), e também aqueles com menos Habilidade. 66

LISTA DE ABREVIATURAS E SIGLAS

AD	Detecção de Anomalia (do inglês *Anomaly Detection*)
AI	Inteligência Artificial (do inglês *Artificial Intelligence*)
AUC	Área sob a curva ROC (do inglês *Area Under Curve ROC*)
AUC-PR	Área sob a curva *Precision-Recall* (do inglês *Area Under Curve Precision-Recall*)
CSV	Valores Separados por Vírgulas (do inglês *Comma-Separated Values*)
EM	*Excess-Mass*
GMMs	Modelos de Misturas Gaussianas (do inglês *Gaussian Mixture Models*)
GRM	Modelo de Resposta Gradual (do inglês *Graded Response Model*)
GT	*Ground Truth*
ICCs	Curvas Características de Item (do inglês *Item Characteristic Curve*)
IRT	Teoria de Resposta ao Item (do inglês *Item Response Theory*)
LOF	*Local Outlier Factor*
ML	Aprendizagem de Máquina (do inglês *Machine Learning*)
MV	*Mass-Volume*
NLP	Processamento de Linguagem Natural (do inglês *Natural Language Processing*)
OCSVM	*One-Class SVM*
ODAIRE	Detecção de Anomalias baseada em Concordância da REsposta ao Item (do inglês *Outlier Detection Agreement-based Item REsponses*)
PCA	*Principal Component Analysis*
PCM	Modelo de Crédito Parcial (do inglês *Partial Credit Model*)
ROC	Característica Operacional do Receptor (do inglês *Receiver Operating Characteristic*)
UOS	Unificando Pontuações de Anomalias (do inglês *Unifying Outlier Scores*)
XAI	Inteligência Artificial Explicável (do inglês *Explainable AI*)

LISTA DE SÍMBOLOS

γ	Letra grega Gama
$\in$	Pertence
∞	Infinito
$\geq$	Maior ou Igual
δ	Letra grega Delta
ω	Letra grega Omega
τ	Letra grega Tau
θ	Letra grega Teta
σ	Letra grega Sigma
μ	Letra grega Mi
β	Letra grega Beta
α	Letra grega Alpha

SUMÁRIO

Lista de tabelas 13

1 INTRODUÇÃO 19

1.1 PROBLEMA E MOTIVAÇÃO 19

1.2 PROPOSTA DA DISSERTAÇÃO 21

1.3 ESTRUTURA DA DISSERTAÇÃO 22

2 REVISÃO DA LITERATURA E TRABALHOS RELACIONADOS . 23

2.1 AVALIAÇÃO DE MÉTODOS DE DETECÇÃO DE ANOMALIAS 23

2.2 MÉTRICAS DE AVALIAÇÃO DOS MÉTODOS DA DETECÇÃO DE ANOMALIAS 25

2.2.1 Avaliação supervisionada da Detecção de anomalias 25

2.2.1.1 Acurácia 25

2.2.1.2 Precisão 26

2.2.1.3 Cobertura 26

2.2.1.4 F1-Score 27

2.2.1.5 Área sob a curva ROC 27

2.2.1.6 Área sob a curva Precision-Recall 28

2.2.1.7 Log loss 28

2.2.2 Avaliação não-supervisionada da Detecção de anomalias 29

2.2.2.1 Pseudo-Ground Truth 29

2.2.2.2 Excess-Mass e Mass-Volume 30

2.3 TEORIA DE RESPOSTA AO ITEM 32

2.3.1 TRI em Psicometria 32

2.3.1.1 Modelos de Resposta Dicotômica 33

2.3.1.2 Modelos de Resposta Politômica 34

2.3.1.3 Modelos de Resposta Contínua 35

2.3.2 TRI em Inteligencia Artificial 37

3 AVALIAÇÃO DA DETECÇÃO DE ANOMALIAS BASEADA EM TEORIA DE RESPOSTA AO ITEM E EM MODELO DE CONCORDÂNCIA 39

3.1 MÓDULO DE DETECÇÃO DE ANOMALIAS 40

3.2 MÓDULO DE CONCORDÂNCIA 40
3.3 MÓDULO IRT 42
3.4 OUTROS MÓDULOS 42
4 EXPERIMENTOS REALIZADOS 43
4.1 CONJUNTOS DE DADOS 43
4.1.1 Dados Sintéticos 43
4.1.2 Dados Reais 44
4.1.3 Pré-processamento dos dados 46
4.1.4 Características dos dados 47
4.1.5 Características das anomalias 48
4.2 ALGORITMOS DE AD 49
4.2.1 Parâmetros dos Modelos 51
4.2.2 Treinamento dos Modelos 52
4.2.3 Predições no conjunto de testes 53
4.3 MÓDULO DE CONCORDÂNCIA 53
4.4 MODELO IRT 54
4.5 *BASELINES* 55
4.5.1 Métricas de avaliação 56
4.5.2 Ranqueamentos de Métodos AD 57
4.5.3 Correlação das Métricas de Métodos AD 57
4.5.4 Comparação de combinações da Matriz de Predições, Concordâncias, com o Traço latente de IRT 59
4.5.5 Ranqueamento de Traço latente de IRT 61
4.5.6 Correlação das Métricas de avaliação e Traço latente de IRT 62
5 RESULTADOS DOS EXPERIMENTOS 64
5.1 CORRELAÇÃO ENTRE Detecção de Anomalias baseada em Concordância da REsposta ao Item (do inglês *Outlier Detection Agreement-based Item REsponses*) (ODAIRE) E OUTRAS MÉTRICAS 64
5.2 HABILIDADES ESTIMADAS 67
5.3 DIFICULDADES ESTIMADAS 70
5.4 DISCRIMINAÇÕES ESTIMADAS 73
5.5 PONTOS NEGATIVOS 74
5.5.1 Vieses nos dados 75

5.5.2 Vieses nos algoritmos 75
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS 77
6.1 TRABALHOS FUTUROS 77
REFERÊNCIAS 80
APÊNDICE A – REPRESENTAÇÃO GRÁFICA BIDIMENSIONAL DOS DATASETS 85
APÊNDICE B – VISUALIZAÇÃO TABULAR DOS DADOS GERADOS POR MÓDULOS DO ODAIRE OU EM SEUS EXPERIMENTOS 86
APÊNDICE C – VISUALIZAÇÃO GRÁFICA BIDIMENSIONAL DOS DATASETS E SEUS TRAÇOS LATENTES IRT . 92
ANEXO A – DADOS REAIS PÚBLICOS UTILIZADOS 93
ANEXO B – MÉTODOS DE DETECÇÃO DE ANOMALIAS UTILIZADOS 99

1 INTRODUÇÃO

Nesse capítulo de introdução, será apresentado o contexto geral onde a presente dissertação está inserida. Que no caso trata-se do problema em avaliar métodos de AD, em cenários de aprendizagem não-supervisionada, com a adoção de Modelos de Concordância e de IRT. Na Seção 1.1, além do contexto geral serão apresentados o Problema de pesquisa e a Motivação. Em seguida na Seção 1.2, a Proposta da dissertação com seus Objetivo geral e específicos serão apresentados. Assim como um breve resumo do trabalho que foi desenvolvido. Finalmente, na Seção 1.3 será apresentada a estrutura desta dissertação.

1.1 PROBLEMA E MOTIVAÇÃO

Porquê a escolha do tema detecção de anomalias (AD) nesta dissertação? Basicamente se deve à sua ampla relevância prática e aos desafios, ainda presentes, na avaliação de algoritmos desta área, especialmente em cenários não supervisionados. Em aplicações como segurança da informação, saúde, finanças e manutenção preditiva, a identificação de comportamentos anômalos é essencial, mas frequentemente ocorre na ausência de rótulos confiáveis (*ground truth*), o que inviabiliza o uso de métricas tradicionais supervisionadas. Além disso, as métricas agregadas geralmente utilizadas, como AUC ou *F1-Score*, ocultam a variabilidade entre instâncias e não capturam a real dificuldade que certos dados impõem aos algoritmos. Essa limitação abre espaço para abordagens mais inteligentes e detalhadas de avaliação. Assim, a dissertação propõe o uso inovador da Teoria de Resposta ao Item (IRT), combinada com modelos de concordância, para oferecer uma solução que permite estimar, simultaneamente, a habilidade dos detectores de *outliers*, e a dificuldade das instâncias, sem a necessidade de rótulos. Essa abordagem contribui para preencher uma lacuna importante na literatura e oferece novas perspectivas para a validação de modelos em tarefas não supervisionadas.

A validação de métodos de AD (IVANOVSKA et al., 2021) é bastante relevante para a seleção dos algoritmos mais adequados à determinado conjunto de dados, revelando assim instâncias discrepantes significativas numa base de dados. As abordagens existentes para validação de detecção de valores anômalos podem ser amplamente divididas em duas categorias: validação supervisionada e não supervisionada. Medidas de validação supervisionadas são usadas para comparar as instâncias discrepantes retornadas por um algoritmo com uma anomalia de dados

conhecida, como classes supervisionadas. Apesar de amplamente utilizadas em experimentos de *benchmark*, as medidas supervisionadas não são aplicáveis quando a verdade anotada (GT) sobre os dados não está disponível, o que é muito comum em aplicações práticas. Já a validação não supervisionada mede a qualidade da anormalidade das instâncias, geralmente baseada em uma estratégia que considera apenas pseudo-rótulos e estatísticas de densidade de distribuição das anomalias retornadas. As medidas de validação não supervisionadas não precisam de nenhuma informação externa de GT, no entanto, apresentam outras limitações, como não poderem ser bem estimadas em grandes dimensões ou um viés de incerteza comum para imitar o GT.

Independentemente da abordagem, seja ela não supervisionada ou supervisionada, uma dimensão que normalmente é negligenciada na validação da detecção de anomalias é considerar a qualidade de um modelo de detecção de anomalias para situações ou instâncias particulares em um conjunto de dados. Por exemplo, a pontuação Pseudo GT é calculada para cada instância em um conjunto de dados, mas para medir o desempenho do modelo, é calculada a média entre as instâncias, sem levar em conta que algumas instâncias podem ser mais difíceis do que outras. Essas medidas simples de validação agregada escondem o fato de que alguns modelos são muito bons para detectar valores discrepantes de itens fáceis, mas não são robustos para instâncias de anomalias em fronteiras de anomalias de grupo, valores discrepantes de distribuição de densidade ou outras situações desafiadoras. A análise de desempenho em nível de instância tem sido mais comum em cenários de Aprendizagem de Máquina (do inglês *Machine Learning*) (ML) supervisionado (BURNELL et al., 2023), mas não é profundamente investigada para detecção de anomalias.

Aqui propomos uma nova solução para validação de detecção de outliers, ODAIRE, que se baseia em modelos de IRT. (EMBRETSON; REISE, 2013). A IRT tem sido amplamente adotada em psicometria para medir as habilidades cognitivas de respondentes humanos (por exemplo, estudantes) com base em suas respostas a itens de teste (por exemplo, questões de exames) com diferentes níveis de dificuldade. Altas habilidades são atribuídas aos respondentes que tendem a responder corretamente aos itens de maior dificuldade, mantendo respostas robustas para os itens fáceis. Na IRT, habilidades e dificuldades são definidas como parâmetros de Curvas Características de Item (do inglês *Item Characteristic Curve*) (ICCs), que modelam a probabilidade de resposta correta para um item. Esses parâmetros são geralmente estimados para maximizar a probabilidade de respostas observadas em um teste.

1.2 PROPOSTA DA DISSERTAÇÃO

Recentemente, os modelos IRT foram adaptados para avaliar sistemas de AI (MARTÍNEZ-PLUMED et al., 2016), especificamente para avaliação de algoritmos em tarefas supervisionadas como classificação. No entanto, o uso da IRT para avaliação de detecção de anomalias não é simples devido à falta de uma verdade básica ou de uma resposta correta. Portanto, na solução proposta, a IRT é utilizada para modelar a probabilidade de concordância entre métodos ao detectar *outliers* em um determinado conjunto de dados. Seguindo a nomenclatura da IRT, cada respondente é um modelo de AD, enquanto cada item é uma instância em um conjunto de dados a ser atribuído ou não como *outlier*. Cada resposta, por sua vez, está associada a um único modelo e instância, indicando o quanto o modelo concorda com outros modelos ao atribuir aquela instância. Com base numa matriz de resposta produzida por um conjunto de modelos num conjunto de dados, o ODAIRE estima simultaneamente duas informações distintas: (1) a capacidade de cada modelo de detecção de *outliers*; e (2) a dificuldade de cada instância. Valores de capacidade mais altos são retornados para modelos de detecção de valores discrepantes que concordam sobre como atribuir valores discrepantes para as instâncias mais difíceis. Num duplo sentido, uma instância é considerada difícil quando apenas os bons modelos concordam sobre como atribuir valores discrepantes a essa instância. Consequentemente, bons modelos são aqueles que alcançam um consenso mais alto, dando mais importância às instâncias que são mais difíceis de atribuir adequadamente como *outliers*.

Nos experimentos realizados, o ODAIRE foi utilizado para avaliar 23 modelos diferentes de AD produzidos por diferentes famílias de algoritmos. No ODAIRE, o modelo β4-IRT (FERREIRA-JUNIOR et al., 2023) foi utilizado para estimar habilidades e dificuldades. Os experimentos foram realizados com 20 conjuntos de dados com características diferentes (local, global, dependência e anomalias agrupadas). Os resultados revelaram uma alta correlação entre habilidade e 10 medidas de validação diferentes (AUC, Área sob a curva *Precision-Recall* (do inglês *Area Under Curve Precision-Recall*) (AUC-PR), *Precision*, *Recall*, *F-Measure*, Acurácia, *Log-loss*, Pseudo GT, EM e MV), onde sete são avaliações supervisionadas e três não supervisionadas. Este resultado indica que a capacidade foi realmente útil para identificar os modelos que produziram boa classificação dos dados para anomalia ou não. As contribuições desta dissertação podem ser então resumidas da seguinte forma:

- Uma nova abordagem para validação de AD, baseada em IRT e Modelo de Concordância,

que abre novas ideias para avaliação de algoritmos em outras tarefas não supervisionadas ou semi-supervisionadas, quando o GT não é totalmente conhecida;

- ODAIRE pode ser adotado não apenas para validação de detecção de anomalias, mas também para estimar dificuldade de instância, quantificando e revelando áreas incertas em um conjunto de dados;
- Os experimentos revelaram significativa correlação entre a capacidade do modelo e as medidas usuais de validação de detecção de anomalias. Isto é bastante promissor, uma vez que a solução proposta não depende de informações supervisionadas para validação de detecção de *outliers*.

1.3 ESTRUTURA DA DISSERTAÇÃO

A dissertação está organizada em seis capítulos, como se segue: Capítulo 1 introduz o problema e a motivação deste trabalho, além da proposta desta dissertação para mitigar tal problema; Capítulo 2 fornecemos uma breve revisão sobre validação em modelos de AD, seguida por uma visão geral dos conceitos e trabalhos relacionados à IRT; Capítulo 3 descrevemos a ODAIRE, a solução que é proposta nesta dissertação; Capítulo 4 descrevemos detalhadamente a configuração e implementação necessária, para que fossem realizados os experimentos com a solução proposta neste trabalho; Capítulo 5 apresentamos os resultados obtidos nos experimentos aqui realizados; Capítulo 6 concluímos o trabalho com as considerações finais e possíveis trabalhos futuros.

2 REVISÃO DA LITERATURA E TRABALHOS RELACIONADOS

Neste capítulo, apresentamos brevemente o tema da avaliação de algoritmos de AD, com o foco em apresentar os desafios da avaliação e discutir brevemente as diferentes técnicas e métricas da literatura. Em seguida, revisamos os conceitos de IRT com seus modelos e alguns usos de IRT em problemas de ML.

2.1 AVALIAÇÃO DE MÉTODOS DE DETECÇÃO DE ANOMALIAS

Há trabalhos anteriores já existentes na literatura, que abordam o tema relacionado à avaliação dos métodos de AD. Como por exemplo no trabalho de (IVANOVSKA et al., 2021), a detecção de *outliers* em conjuntos de dados complexos representa um desafio recorrente na área de Visão Computacional. Especialmente em contextos industriais, como o controle de qualidade automatizado. Nesses cenários, a ocorrência de amostras anômalas é tipicamente rara, resultando em bases de dados altamente desbalanceadas. Destacam que, embora as abordagens baseadas em aprendizado profundo, como Redes Adversárias Generativas (GANs), tenham avançado significativamente na detecção de anomalias, a forma como esses modelos são avaliados ainda apresenta limitações consideráveis.

Tradicionalmente, a avaliação do desempenho de algoritmos de detecção de anomalias utiliza métricas como a AUC (Área sob a Curva ROC) e a AP (Precisão Média). No entanto, os autores argumentam que essas métricas são inadequadas para conjuntos de dados com forte desequilíbrio entre classes, como é comum em aplicações do mundo real. Isso ocorre porque métricas globais não capturam com precisão a capacidade do modelo de operar eficientemente em pontos específicos da curva ROC, pontos esses que são cruciais para aplicações industriais, onde é necessário detectar a maioria das anomalias sem gerar um número excessivo de falsos positivos.

Com o intuito de suprir essa limitação, (IVANOVSKA et al., 2021) propõem uma métrica alternativa denominada "$\%TNX\%TP$". Essa métrica avalia a proporção de verdadeiros negativos ($\%TN$) que o modelo é capaz de identificar ao alcançar uma taxa fixa de verdadeiros positivos ($X\%TP$), permitindo assim mensurar de forma mais realista o impacto do modelo na redução da carga de inspeção manual. Além disso, a métrica é flexível, permitindo que o valor de X seja ajustado conforme os requisitos do cliente ou da aplicação.

Os experimentos conduzidos pelos autores incluíram quatro arquiteturas baseadas em GANs (GANomaly, skip-GANomaly, f-AnoGAN e OCGAN) aplicadas a conjuntos de dados balanceados (MNIST, Fashion-MNIST e CIFAR-10) e desbalanceados (imagens de satélite e de defeitos em aço). Os resultados evidenciaram que, nos cenários desbalanceados, a AUC e a AP apresentam baixa correlação com a métrica *%TNX%TP*, podendo levar a conclusões equivocadas quanto à eficácia dos modelos. Dessa forma, os autores defendem que a avaliação de algoritmos de detecção de anomalias deve considerar métricas mais específicas e sensíveis ao desequilíbrio dos dados, especialmente em aplicações industriais, onde o desempenho prático e a economia de recursos são fatores prioritários.

Outro trabalho que vale a pena destacar, é o estudo feito por (GOLDSTEIN; UCHIDA, 2016), onde realizaram uma das mais abrangentes avaliações comparativas de algoritmos de detecção de *outliers* não supervisionados, aplicados a dados multivariados. Os autores destacam que, embora a detecção de anomalias, também conhecida como detecção de outliers, seja amplamente utilizada em diversas aplicações práticas, como segurança da informação, detecção de fraudes e análise médica, ainda há uma carência significativa de avaliações sistemáticas que possibilitem a comparação justa entre os diversos métodos propostos na literatura. Para enfrentar essa lacuna, o estudo avaliou 19 algoritmos representativos de diferentes abordagens, como métodos baseados em vizinhança, agrupamento, modelos estatísticos e técnicas baseadas em SVM e análise de componentes principais.

O trabalho diferencia os tipos de anomalias comumente encontrados em aplicações reais, incluindo anomalias globais, locais, coletivas e contextuais. Essa distinção é importante, pois algoritmos diferentes apresentam desempenhos distintos dependendo da natureza das anomalias presentes no conjunto de dados. A pesquisa também salienta a importância da normalização adequada e da preparação dos dados para transformar problemas complexos (como detecção de anomalias coletivas) em tarefas tratáveis por algoritmos de detecção de anomalias pontuais.

Para conduzir a avaliação, (GOLDSTEIN; UCHIDA, 2016) utilizaram 10 conjuntos de dados multivariados de diferentes domínios, cobrindo uma ampla variedade de tamanhos, dimensionalidades e proporções de anomalias. Cada algoritmo foi avaliado com base na área sob a curva ROC (AUC), uma métrica amplamente aceita na literatura por considerar a ordenação das instâncias em termos de anomalia. Foram exploradas diferentes configurações de parâmetros, como o número de vizinhos k, com o objetivo de medir a robustez e a sensibilidade dos métodos.

Os resultados revelam que não existe um algoritmo que se destaque universalmente em to-

dos os cenários. Algoritmos baseados em vizinhança, como kNN e LOF, mostraram-se eficazes em várias situações, sendo o kNN particularmente robusto em tarefas envolvendo anomalias globais. Já métodos como o LOF, projetados para identificar anomalias locais, apresentaram desempenho inferior em contextos com anomalias globais. Algoritmos estatísticos simples, como o HBOS, demonstraram excelente eficiência computacional, sendo indicados para aplicações com restrições de tempo de processamento. Métodos mais sofisticados, como os baseados em SVM (One-Class SVM e n-SVM) e PCA robusta (rPCA), também alcançaram bons resultados, mas tendem a ser mais sensíveis à escolha de parâmetros e à dimensionalidade dos dados.

Além da análise de desempenho, um dos principais méritos do trabalho está na disponibilização pública dos conjuntos de dados utilizados e do código-fonte dos algoritmos, promovendo a reprodutibilidade e a padronização de benchmarks na área. Dessa forma, o estudo de (GOLDSTEIN; UCHIDA, 2016) representa uma importante referência metodológica para pesquisas futuras, oferecendo uma base sólida para a escolha informada de algoritmos de detecção de anomalias não supervisionados em contextos reais.

2.2 MÉTRICAS DE AVALIAÇÃO DOS MÉTODOS DA DETECÇÃO DE ANOMALIAS

2.2.1 Avaliação supervisionada da Detecção de anomalias

Existem diferentes formas para avaliar o desempenho de métodos de AD na literatura. A avaliação pode ser feita através de métricas supervisionadas, que requerem dados de anomalias conhecidas e rotuladas (informação do GT). Nesse caso, o problema de AD é avaliado como um problema de classificação, em que a classe positiva contempla exemplos rotulados como anomalias, e por sua vez a classe negativa contempla os exemplos não-anômalos. Várias métricas supervisionadas já foram propostas na literatura, porém não existe uma melhor métrica para todas as situações. Nessa seção, descrevemos algumas métricas comumente adotadas no contexto de AD.

2.2.1.1 Acurácia

A acurácia (Acc) é uma das medidas mais comumente utilizadas para o desempenho da classificação e é definida como a razão entre as amostras classificadas corretamente e o número total de amostras, como definida na equação:

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.1}$$

onde *VP*, *VN*, *FN* e *FP* são respectivamente os números de: verdadeiros positivos; verdadeiros negativos; falsos negativos; e falsos positivos, obtidos pelo detector avaliado na base de dados de teste (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

A métrica de acurácia é sensível a dados com classes desbalanceadas. Outro problema com a acurácia é que dois classificadores podem produzir a mesma acurácia, mas funcionar de forma diferente no que diz respeito aos tipos de decisões corretas e incorretas que fornecem.

2.2.1.2 Precisão

A *Precision* denota a proporção de exemplos previstos como positivos que são corretamente positivos. No contexto de detecção de anomalias, essa métrica tem a vantagem de dar uma importância maior aos exemplos da classe de anomalias (positivas), que de fato é a classe de maior interesse. A precisão é definida então como o número de exemplos de teste corretamente preditos como anomalias dividido pelo número de exemplos preditos como anomalias (POWERS, 2020):

$$Prec = \frac{VP}{VP + FP} \tag{2.2}$$

Esta medida, assim como a *cobertura*, que será vista na próxima seção, foca nos exemplos e previsões da classe positiva. Valores altos dessa métrica são observados para um número alto de anomalias corretamente detectadas, no entando pode ser penalizada no caso de modelos com altas taxas de falsos positivos ou falsos alarmes (exemplos normais detectados incorretamente como anomalias). A métrica não considera verdadeiros negativos, que são de fato menos importantes no contexto de detecção de anomalias e no caso de problemas de classificação com classes muitos desbalanceadas.

2.2.1.3 Cobertura

A cobertura (*recall*, sensibilidade ou taxa de verdadeiros positivos) é a proporção de casos positivos reais que são previstos corretamente. No contexto de detecção de anomalias, essa

métrica captura a sensibilidade do detector de identificar as anomalias de dados quando elas de fato existem. A cobertura é definida pela equação:

$$Rec = \frac{VP}{VP + FN} \tag{2.3}$$

Essa medida, assim como a precisão, dá uma maior importância às predições corretas da classe positiva. No entanto, ela é penalizada no caso de taxas altas de falsos negativos (exemplos de anomalias que não foram detectadas).

2.2.1.4 F1-Score

Dentre as várias medidas de desempenho derivadas da matriz de confusão, uma das mais relevantes é a *F1-Score*, ou *F-measure* (RIJSBERGEN, 1979), definida como a média harmônica de Precisão (*Precision*) e Cobertura (*Recall*). Utilizada para avaliar o desempenho de modelos de classificação, especialmente em problemas de classificação binária (POWERS, 2019).

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec} \tag{2.4}$$

Essa medida de desempenho considera o equilíbrio entre duas métricas fundamentais: Precisão e Cobertura (ou Sensibilidade). A fórmula 2.4 assegura que a *F1-Score* atinja seu valor máximo de 1 quando tanto a *Precision* quanto o *Recall* são perfeitas, e um valor mínimo de 0 quando ambas são nulas.

2.2.1.5 Área sob a curva ROC

A Curva Característica Operacional do Receptor (do inglês *Receiver Operating Characteristic*) (ROC) é uma representação gráfica que avalia a performance de um classificador binário ao variar seu limiar de decisão, plotando a taxa de verdadeiros positivos (sensibilidade) no eixo *y* contra a taxa de falsos positivos (1 - especificidade) no eixo *x*. Permitindo visualizar o equilíbrio entre Sensibilidade e Especificidade do modelo, auxiliando na escolha do limiar que melhor distingue entre as classes.

Já a AUC, uma medida associada à curva ROC, é uma métrica que quantifica a capacidade geral de discriminação do classificador, onde um valor de 1 indica desempenho perfeito e 0,5 representa uma classificação aleatória, e valores abaixo de 0,5 sugerem desempenho pior que o

aleatório. Esta curva ROC é amplamente utilizada em estatística e ML para comparar e validar modelos de classificação, assim como a AUC também, fornecendo uma medida agregada de desempenho em diferentes limiares de decisão. (IVANOVSKA et al., 2021).

2.2.1.6 Área sob a curva Precision-Recall

A curva *Precision-Recall* é uma ferramenta gráfica utilizada para avaliar o desempenho de modelos de classificação binária, especialmente em situações onde há um desbalanceamento entre as classes. Nesta curva, a Precisão (ou *Precision*) é plotada no eixo *y*, enquanto o *Recall* (ou Sensibilidade) é plotado no eixo *x*. A curva demonstra como a Precisão do modelo varia em relação à Cobertura em diferentes limiares de decisão. Isso permite analisar o *trade-off* entre identificar corretamente as instâncias positivas (Cobertura) e a proporção de predições positivas que são corretas (Precisão).

Já a AUC-PR é uma métrica que resume o desempenho de um classificador binário, especialmente útil em conjuntos de dados desbalanceados. Pois quantifica a área sob essa curva, fornecendo uma medida única da capacidade do modelo de distinguir entre as classes. Valores de AUC-PR próximos de 1 indicam que o classificador possui alta Precisão e *Recall*, enquanto valores próximos de 0,5 sugerem um desempenho semelhante ao acaso (DAVIS; GOADRICH, 2006).

Diferentemente da AUC, a AUC-PR é mais informativa quando há um desbalanceamento significativo entre as classes, pois foca no desempenho da classe positiva, tornando-se uma ferramenta valiosa para avaliar modelos em cenários onde a classe de interesse é rara.

2.2.1.7 Log loss

A *Log Loss*, também conhecida como Perda Logarítmica ou *Cross-Entropy Loss*, é uma métrica utilizada para avaliar o desempenho de modelos de classificação, tanto binários quanto multiclasse. Ela mede a incerteza das previsões probabilísticas do modelo, penalizando fortemente previsões incorretas feitas com alta confiança. Valores mais baixos de *Log Loss* correspondem a modelos cujas previsões probabilísticas estão mais alinhadas com os rótulos reais, onde 0 indica previsões perfeitas, indicando melhor calibração do modelo. Essa métrica é especialmente útil em situações com classes desbalanceadas, pois considera as probabilidades previstas, oferecendo uma avaliação mais detalhada do desempenho do modelo em comparação

com métricas como acurácia (MAO; MOHRI; ZHONG, 2023).

Embora a *Cross-Entropy Loss* seja predominante em problemas de classificação supervisionada, ela também pode ser aplicada em cenários de AD, especialmente quando modelos probabilísticos, que envolvem a estimativa de probabilidades (Regressão Logística, *Naive Bayes*), ou de reconstrução de dados (*Autoencoders*, Modelos de Mistura Gaussiana, Redes Geradoras Adversariais) são utilizados também. Além disso, em Métodos de AD baseados em modelos de classificação, a *Log Loss* pode ser utilizada para treinar o modelo a distinguir entre classes normais e anômalas, atribuindo probabilidades que refletem a confiança nas previsões. (BISHOP; NASRABADI, 2006).

2.2.2 Avaliação não-supervisionada da Detecção de anomalias

O uso de medidas de avaliação supervisionadas requer a disponibilidade de dados rotulados. No caso de AD, nem sempre uma boa base de testes com dados rotulados está disponível para um cálculo confiável de medidas de avaliação supervisionadas. Existem outras técnicas para avaliar o desempenho de métodos de AD na literatura que não requerem dados rotulados e são consideradas assim não supervisionadas. Nesta seção, descrevemos algumas dessas técnicas comuns e suas limitações.

2.2.2.1 Pseudo-Ground Truth

Pseudo GT (ou pseudo rótulo-verdadeiro) é um termo usado em ML para se referir a rótulos gerados automaticamente, que são usados como substitutos temporários para dados rotulados manualmente (GT real). É utilizado em situações em que o rótulo verdadeiro não está disponível, ou é caro e demorado de ser obtido. Com o objetivo de permitir que modelos possam ser treinados de maneira semi-supervisionada, ou auto-supervisionada, usando esses pseudo rótulos como uma aproximação do rótulo verdadeiro real (KANDANAARACHCHI, 2022).

O processo para utilizar pseudo GT geralmente se inicia com a seleção de um conjunto de dados contendo observações não rotuladas. Em seguida, é gerado o pseudo GT a partir de alguma destas maneiras: Modelos pré-treinados (usar um modelo já treinado num conjunto de dados similar, para rotular *outliers* de maneira aproximada); Regras heurísticas (definir regras que identificam padrões de *outliers*, como valores extremos, ou desvio padrão acima de certo limiar); Métodos de clusterização (aplicar técnicas de clusterização, como *K-means* ou

DBSCAN, para separar dados em grupos e considerar os pontos distantes do centróide como *outliers*); e Modelos estatísticos (utilizar distribuições estatísticas para modelar o comportamento esperado, e identificar observações fora do intervalo esperado como *outliers*).

Outra maneira de gerar pseudo GT, é utilizando o *Mixture Modeling*, o qual usa distribuições probabilísticas (como o Modelos de Misturas Gaussianas (do inglês *Gaussian Mixture Models*) (GMMs)) para modelar a distribuição dos dados e identificar anomalias como as observações que possuem baixa probabilidade sob o modelo ajustado. Por exemplo, um GMMs é ajustado aos dados normais, então a probabilidade logarítmica de cada ponto no conjunto de dados completo é calculada. Depois um limiar de probabilidade é definido para identificar os *outliers*, como os pontos nos 5% mais baixos das probabilidades são considerados anomalias, os quais irão compor o pseudo GT (RAYANA; AKOGLU, 2016).

Há também na literatura, mais uma maneira de gerar pseudo GT, que é a técnica Unificando Pontuações de Anomalias (do inglês *Unifying Outlier Scores*) (UOS) para criar o pseudo GT. O UOS é um método que combina os *scores* de diferentes detectores de *outliers* para criar um *scores* unificado, permitindo a geração de um pseudo GT mais robusto. Por exemplo, usando os três métodos AD: *Isolation Forest*, *Local Outlier Factor* (LOF) e *One-Class SVM* (OCSVM), combina-se os *scores* de cada um deles, com uma média aritmética, por exemplo, para gerar um *scores* unificado. Somado a isto, um limiar é definido para identificar as anomalias nos *scores* unificados acima desse limiar, os quais irão compor o pseudo GT (RAYANA; AKOGLU, 2016).

Logo depois, é feito o treinamento do modelo de AD, utilizando o conjunto de pseudo GT para auxiliar no aprendizado em identificar observações marcadas como anomalias. Por fim, é avaliado o desempenho do modelo de AD, utilizando alguma métrica clássica comum (AUC, *F1-score*, entre outras), com base no pseudo GT e nas predições obtidas.

2.2.2.2 Excess-Mass e Mass-Volume

MV e EM são métricas utilizadas para avaliar algoritmos de detecção de anomalias em cenários não supervisionados, onde rótulos para os dados podem não estar disponíveis. Ambas as métricas foram projetadas para oferecer uma maneira de quantificar a eficácia dos algoritmos na identificação e priorização de regiões do espaço onde as anomalias estão mais concentradas. Diferentemente de métricas supervisionadas, como AUC ou AUC-PR, o MV e EM por sua vez não dependem diretamente de rótulos, mas sim das distribuições das pontuações de anomalia

atribuídas pelos modelos (GOIX, 2016).

MV mede a eficiência de um algoritmo ao capturar anomalias em pequenas regiões do espaço de dados. Para uma fração α da massa total de anomalias (pontuações de anomalia maiores ou iguais a um dado threshold u), $MV(\alpha)$ calcula o menor volume necessário para conter essa fração. Em termos matemáticos, $MV(\alpha)$ é definido como na fórmula 2.5:

$$MV(\alpha) = \inf_{u \geq 0} \{\text{Volume}(s(X) \geq u) \,|\, P(s(X) \geq u) \geq \alpha\} \tag{2.5}$$

onde $P(s(X) \geq u)$ é a probabilidade acumulada de encontrar pontos de dados com pontuações maiores ou iguais a u. Um bom algoritmo deve capturar altas frações de α em volumes pequenos.

EM, por outro lado, foca em priorizar as anomalias mais severas. Ele avalia o *trade-off* entre a fração acumulada da massa total de anomalias ($P(s(X) \geq u)$) e o volume penalizado associado a essa fração ($t \cdot \text{Volume}(s(X) \geq u)$), onde t é um parâmetro de penalização. A métrica $EM(t)$ é definida como na fórmula 2.6:

$$EM(t) = \sup_{u \geq 0} \{P(s(X) \geq u) - t \cdot \text{Volume}(s(X) \geq u)\} \tag{2.6}$$

O objetivo do EM é maximizar a concentração de anomalias em regiões pequenas, penalizando regiões de grande volume.

A principal diferença entre MV e EM está em seu foco. Enquanto MV avalia como um algoritmo consegue capturar frações específicas da massa de anomalias (α), EM considera o balanceamento entre a concentração da massa e o volume ocupado. Ambas as métricas são úteis em diferentes cenários: MV é mais interpretável para entender a eficiência espacial de um modelo, enquanto EM foca em priorização (GOIX, 2016).

Ambas as métricas dependem de estimativas de volume. Em práticas de alta dimensão, volumes são calculados empiricamente utilizando dados uniformemente distribuídos dentro do suporte dos atributos dos dados reais. Para MV, a fração de massa é integrada sobre um intervalo de α (geralmente $[0.9, 0.999]$), enquanto para EM, o cálculo considera a integração sobre thresholds t até t_{max}.

Áreas sob as curvas (AUC) para MV e EM fornecem uma visão geral do desempenho do algoritmo em diferentes *thresholds* ou frações. Áreas maiores indicam melhor concentração das anomalias em volumes pequenos (MV) ou maior eficiência em penalizar volumes grandes enquanto maximiza a fração acumulada de massa (EM).

MV e EM são particularmente úteis em cenários não supervisionados, onde as métricas tradicionais, como AUC (ROC) ou AUC-PR, não podem ser aplicadas devido à ausência de rótulos. Elas oferecem *insights* complementares sobre o comportamento do algoritmo, especialmente em casos onde a severidade ou concentração das anomalias é um fator crítico (GOIX, 2016).

Por fim, essas métricas são amplamente aplicáveis em áreas como detecção de fraudes financeiras, segurança cibernética e manutenção preditiva. A análise de EM e MV permite comparar algoritmos de detecção de anomalias em diferentes espaços de dados e identificar os melhores modelos para priorizar e concentrar anomalias em volumes compactos.

2.3 TEORIA DE RESPOSTA AO ITEM

A IRT abrange uma família de modelos de traços latentes que tem sido tradicionalmente usada em Psicometria educacional para determinar a habilidade dos estudantes em um teste e a dificuldade e discriminação das questões do teste. De acordo com (KANDANAARACHCHI, 2022), a IRT tem sido estendida para avaliação de algoritmos e conjuntos de dados, tarefas de classificação e detecção de anomalias não supervisionadas, ampliando a aplicabilidade da IRT ao domínio de ML e AI.

2.3.1 TRI em Psicometria

As primeiras aplicações do modelo IRT foram na área de Psicometria. O objetivo da IRT era ser uma ferramenta de avaliação confiável, capaz de prever a habilidade dos respondentes humanos ao responder a um conjunto de perguntas (HAMBLETON, 1991). Esse método de avaliação vai muito além de apenas contar o número de respostas corretas, mas dá uma maior importância a respostas corretas a diferentes questões dependendo do nível de dificuldade identificado. Além da dificuldade dos itens, diferentes formulações dos modelos IRT podem permitir outros parâmetros como a discriminação dos itens e a taxa de adivinhação (BIRNBAUM, 1968).

Existem vários tipos de modelos IRT, dependendo do domínio da resposta observada. Os primeiros modelos IRT propostos foram desenvolvidos para lidar apenas com respostas binárias (corretas ou incorretas) e são chamados de modelos dicotômicos (LORD; NOVICK, 2008). Esses modelos não são adequados para questões de resposta múltipla, o que motivou a derivação

dos modelos politômicos (SAMEJIMA, 1968). Contudo, se as categorias de resposta apresentam uma ordenação intrínseca, os modelos politômicos não são suficientes; nesses casos, os modelos de resposta graduada foram proposta para melhor caracterizar essa ordenação. Finalmente, surgiram os modelos contínuos, capazes de lidar com respostas contínuas e não apenas discretas, sendo a melhor escolha para modelar pontuações em questões abertas (RASCH, 1960) (NOEL; DAUVIER, 2007). Nas próximas subseções, entraremos em mais detalhes sobre cada um desses tipos de modelos.

2.3.1.1 Modelos de Resposta Dicotômica

São aqueles que consideram respostas em um formato binário, ou seja, classificam as respostas dos itens como corretas ou incorretas. Esses modelos são amplamente utilizados em psicometria e são baseados na ideia de que a probabilidade de um respondente acertar um item é determinada por sua habilidade em relação aos parâmetros do item.

Um dos elementos principais deste modelo são as ICCs, as quais relacionam a probabilidade de uma resposta correta com a habilidade latente do respondente. A curva é modelada por uma função logística e depende de parâmetros específicos do item.

Os possíveis parâmetros dos Modelos Dicotômicos são: Dificuldade (b), que representa o nível de habilidade necessário para que um respondente tenha 50% de chance de responder corretamente ao item; discriminação (a), a qual mede o quanto o item diferencia entre respondentes com diferentes níveis de habilidade. Um item com alta discriminação tem uma curva íngreme, indicando que pequenas variações na habilidade resultam em grandes diferenças na probabilidade de resposta correta; acerto casual (c), que é usado em modelos de três parâmetros (3PL), representa a probabilidade de um respondente acertar o item ao acaso.

Portanto, um dos Modelos Dicotômicos é o Modelo de Um Parâmetro (1PL), também conhecido como modelo de *Rasch*, considera apenas o parâmetro de dificuldade (RASCH, 1960), como pode ser visto na fórmula 2.7:

$$P(X = 1 \mid \theta) = \frac{1}{1 + e^{-(\theta - b)}} \tag{2.7}$$

Onde: $P(X = 1 \mid \theta)$ é a probabilidade do respondente acertar o item; o θ é a Habilidade latente do respondente; o b é o parâmetro de dificuldade do item; por fim, e representa a base do logaritmo natural.

Outro é o Modelo de Dois Parâmetros (2PL), que inclui os parâmetros de dificuldade e discriminação (LORD; NOVICK, 2008), como pode ser visto na fórmula 2.8:

$$P(X = 1 \mid \theta) = \frac{1}{1 + e^{-a(\theta - b)}} \tag{2.8}$$

Onde: a é o parâmetro de discriminação, que define a inclinação das ICCs (capacidade do item de distinguir habilidades próximas). Os demais parâmetros têm o mesmo significado que no modelo anterior.

Por fim, o Modelo de Três Parâmetros (3PL), o qual adiciona o parâmetro de acerto casual ao modelo de dois parâmetros (BIRNBAUM, 1968), como pode ser visto na fórmula 2.9:

$$P(X = 1 \mid \theta) = c + (1 - c) \cdot \frac{1}{1 + e^{-a(\theta - b)}} \tag{2.9}$$

Onde: c representa o parâmetro de acerto casual, com valores típicos entre 0 e 1. Os demais parâmetros têm os mesmos significados que nos modelos anteriores.

Tais modelos dicotômicos são frequentemente usados para medir habilidades latentes em testes educacionais e psicológicos, onde cada questão de um teste representa um item, e a resposta correta ou incorreta reflete a habilidade do respondente. Esses modelos também são aplicados na avaliação de classificadores em ML, conforme destacado nos documentos. Por exemplo, em contextos onde classificadores são avaliados com base em acertos e erros em classificações binárias.

2.3.1.2 Modelos de Resposta Politômica

Tais modelos são uma extensão da IRT que lidam com respostas em categorias ordenadas ou não ordenadas, indo além das respostas dicotômicas (certo ou errado). Esses modelos são usados em situações onde as respostas dos itens podem assumir múltiplas categorias, como escalas de *Likert*, níveis de proficiência ou classificações qualitativas.

Uma de suas principais características é ter resposta em múltiplas categorias: diferentemente dos modelos dicotômicos, os modelos politômicos lidam com situações onde as respostas têm mais de duas opções. Por exemplo, em uma escala de 1 a 5 ou em níveis como "baixo", "médio"e "alto". Outra característica, são nas ICCs: onde cada categoria de resposta possui sua própria curva característica, que descreve a probabilidade de um respondente escolher uma categoria específica com base em sua habilidade (θ). Mais outra característica, também pe-

culiar, são seus parâmetros do item: Dificuldade, que representa a habilidade necessária para atingir uma categoria específica; Discriminação, que mede a capacidade do item de diferenciar respondentes com diferentes níveis de habilidade; e parâmetros adicionais para capturar as transições entre categorias, que alguns modelos incluem.

Os modelos politômicos mais comuns são: Modelo de Resposta Gradual (do inglês *Graded Response Model*) (GRM), desenvolvido para itens com categorias ordenadas. Cada categoria tem um limiar de dificuldade (b_k), representando o ponto em que a probabilidade de escolher aquela categoria supera a da anterior (SAMEJIMA, 1968); Modelo de Crédito Parcial (do inglês *Partial Credit Model*) (PCM), que permite respostas ordenadas, mas não assume que os intervalos entre as categorias sejam iguais. É frequentemente usado em avaliações educacionais (MASTERS, 1982); Modelo Nominal, que lida com categorias não ordenadas, onde não há hierarquia implícita entre as opções de resposta (BOCK, 1972).

Esses modelos de resposta politômica são amplamente usados em psicometria e avaliação educacional, especialmente em contextos onde a resposta do indivíduo reflete níveis progressivos de habilidade ou proficiência. Também são aplicados em contextos como pesquisas com escalas de *Likert* e avaliação de atitudes ou preferências.

2.3.1.3 Modelos de Resposta Contínua

Agora nestes modelos, que são uma extensão dos modelos tradicionais (dicotômicos e politômicos) que lidam com respostas em escalas contínuas, ao invés de respostas binárias (certo/errado) ou categorizadas. Esses modelos são úteis em situações onde os resultados são representados por valores contínuos, ou seja, variáveis que podem assumir valores em um intervalo específico, como [0,1]. E não limitados à categorias discretas (LINDEN, 2018).

Suas características principais são: as respostas contínuas, que permitem a representação de valores em uma escala contínua. Por exemplo, erros absolutos em regressões, probabilidades de classificação, ou *scores* de desempenho ou medidas de precisão, que variam continuamente, em tarefas de medição; as distribuições estatísticas, necessárias para modelar as respostas contínuas, onde distribuições adequadas são utilizadas, como a Distribuição Beta, ou outras funções contínuas que ajustam os dados.

Por exemplo, o modelo β3-IRT (CHEN et al., 2019) é um modelo avançado da IRT, proposto para lidar com respostas contínuas . Ele é amplamente utilizado na avaliação de habilidades latentes em contextos onde as respostas variam continuamente , como taxas de acerto ou

probabilidades de sucesso. O β3-IRT assume que as respostas p_{ij} dadas por um respondente i a um item j seguem uma distribuição *Beta* com parâmetros dependentes da habilidade do respondente θ_i (representa a capacidade latente do respondente), dificuldade do item δ_j (representa o nível de dificuldade do item) e discriminação a_j (determina a capacidade do item de diferenciar respondentes com diferentes níveis de habilidade). Também introduz ICCs mais flexíveis, que podem ter diferentes formas além da tradicional curva logística, permitindo ajustar melhor diferentes padrões de resposta, como por exemplo: Sigmoidal (similar à IRT tradicional); Parabólica (quando $a_j = 1$); e Anti-sigmoidal (quando $0 < a_j < 1$). A fórmula das ICCs deste modelo está definida em 2.10. Onde p_{ij} representa a resposta esperada do respondente i ao item j.

$$E[p_{ij}|\theta_i, \delta_j, a_j] = \frac{1}{1 + \left(\frac{\delta_j}{1-\delta_j}\right)^{a_j} \left(\frac{\theta_i}{1-\theta_i}\right)^{-a_j}}, \tag{2.10}$$

Um outro exemplo também, é o modelo β4-IRT (FERREIRA-JUNIOR et al., 2023), que é a extensão e aprimoramento do modelo β3-IRT apresentado anteriormente. Desenvolvido para resolver limitações associadas à simetria e à estimativa de parâmetros do β3-IRT. Ele mantém a abordagem de modelagem de respostas contínuas e utiliza uma parametrização mais robusta para melhorar a acurácia na estimativa de habilidades latentes, dificuldades dos itens e discriminações. Assim como no modelo de sua versão anterior, as respostas p_{ij} seguem uma distribuição , mas com uma reformulação nos parâmetros para melhorar a robustez. Como na desagregação do parâmetro de discriminação (a_j), que agora nesta versão do modelo, é desmembrado em dois novos parâmetros: ω_j que representa a magnitude da discriminação (valor absoluto); τ_j que representa o sinal (positivo ou negativo) da discriminação. Tal reformulação reduz o problema de simetria, onde valores de discriminação com sinais invertidos geravam dificuldades na estimativa correta de outros parâmetros. Este novo modelo, utiliza agora métodos modernos de descida de gradiente, para estimar os parâmetros de forma eficiente e evitar problemas de inicialização inadequada. Isso é viável devido ao uso de funções de ligação (*link functions*) que transformam parâmetros como ω_j e τ_j em domínios apropriados para otimização. A fórmula das ICCs deste modelo está definida em 2.11.

$$E[p_{ij}|\theta_i, \delta_j, \omega_j, \tau_j] = \frac{1}{1 + \left(\frac{\delta_j}{1-\delta_j}\right)^{\tau_j \cdot \omega_j} \left(\frac{\theta_i}{1-\theta_i}\right)^{-\tau_j \cdot \omega_j}}, \tag{2.11}$$

As principais aplicações, dos Modelos de Resposta Contínua, são: em problemas de regressão, os modelos contínuos permitem avaliar o desempenho de algoritmos; em problemas

onde as respostas são probabilísticas ou não restritas a valores discretos, como resultados de previsões contínuas; e em análises de desempenho em modelos de aprendizado supervisionado.

Por fim, os Modelos de Resposta Contínua são uma evolução importante na IRT, permitindo a análise de dados contínuos e ampliando seu uso em contextos modernos, como avaliação de algoritmos de ML e tarefas de regressão. Sendo usados quando as respostas não são binárias, mas variam de forma contínua, como taxas de precisão ou probabilidades.

2.3.2 TRI em Inteligencia Artificial

IRT aplicada à AI é uma abordagem inovadora que utiliza os conceitos fundamentais da IRT para avaliar modelos de aprendizado de máquina e outras técnicas de AI. Nessa aplicação, os elementos tradicionais da IRT (itens e respondentes) são mapeados para instâncias de dados e modelos preditivos, respectivamente. Essa adaptação permite avaliar o desempenho de algoritmos, identificar regiões de dificuldade em conjuntos de dados e até mesmo analisar a relevância de variáveis (CHEN et al., 2019).

Essa aplicação do IRT busca modelar a interação entre: Sistemas de AI (respondentes), que correspondem aos algoritmos ou modelos de AI (como classificadores, redes neurais ou algoritmos de ML) que realizam as tarefas. A habilidade latente do respondente, na IRT, é análoga à capacidade ou competência do modelo em resolver problemas, ou fazer predições precisas em diferentes tipos de itens; os Itens (tarefas ou instâncias), que representam as instâncias de um conjunto de dados ou as tarefas que os modelos precisam resolver. Estes itens podem ser representados por exemplos de dados, tarefas específicas ou problemas que os sistemas devem resolver. Na IRT, cada item é caracterizado por parâmetros como dificuldade (quão complexo ele é) e discriminação (quão bem ele diferencia sistemas com habilidades distintas); e Respostas, que são os resultados do sistema de AI ao processar os itens, como predições de classes, probabilidades ou outras métricas contínuas. Essas respostas podem ser dicotômicas (correto/incorreto), politômicas (várias categorias), ou contínuas (como probabilidades ou erros absolutos).

Nos trabalhos de (KANDANAARACHCHI, 2022) há uma pesquisa extensa sobre diferentes tipos de mapeamentos do IRT para aplicações em AI, comparando outras abordagens ,como por exemplo, o mapeamento invertido: onde os Respondentes são os conjuntos de dados, e os Itens são os sistemas de AI. Em que as Respostas podem ser tanto os valores originais das respostas, quanto os *scores* derivados a partir de alguma métrica aplicada nas mesmas.

Alguns dos principais usos do IRT aplicado na AI, são em problemas de: Classificação (MARTÍNEZ-PLUMED et al., 2019); Regressão (MORAES, 2021); Reconhecimento de fala (OLIVEIRA et al., 2022); Otimização (KANDANAARACHCHI; SMITH-MILES, 2023), Processamento de Linguagem Natural (do inglês *Natural Language Processing*) (NLP) (LALOR; WU; YU, 2019)(LALOR et al., 2024), Inteligência Artificial Explicável (do inglês *Explainable AI*) (XAI) (MARTíNEZ-PLUMED et al., 2022)(FABRA-BOLUDA et al., 2022)(PRUDÊNCIO; HERNÁNDEZ-ORALLO; MARTINEZ-USÓ, 2015), entre vários outros (REINALDO, 2022)(GOMES, 2022)(MARTINEZ-PLUMED; HERNÁNDEZ-ORALLO, 2016).

Já os maiores benefícios da IRT em AI são: Avaliação detalhada, pois permite uma análise mais granular do desempenho de modelos, atribuindo dificuldade e discriminação a instâncias individuais. Identificando instâncias difíceis ou classificadores que falham consistentemente em itens específicos; Interpretação rica, pois os parâmetros da IRT oferecem insights sobre quais instâncias ou variáveis são mais críticas para o desempenho de um modelo; Generalização, pois tais modelos podem ser adaptados para diferentes tipos de tarefas de AI, como classificação probabilística, regressão e seleção de características. Utilizando habilidades latentes e parâmetros dos itens, para prever como um modelo se comportará em novos contextos ou instâncias; Comparação justa, pois oferece uma base comum para comparar modelos de AI em diferentes domínios.

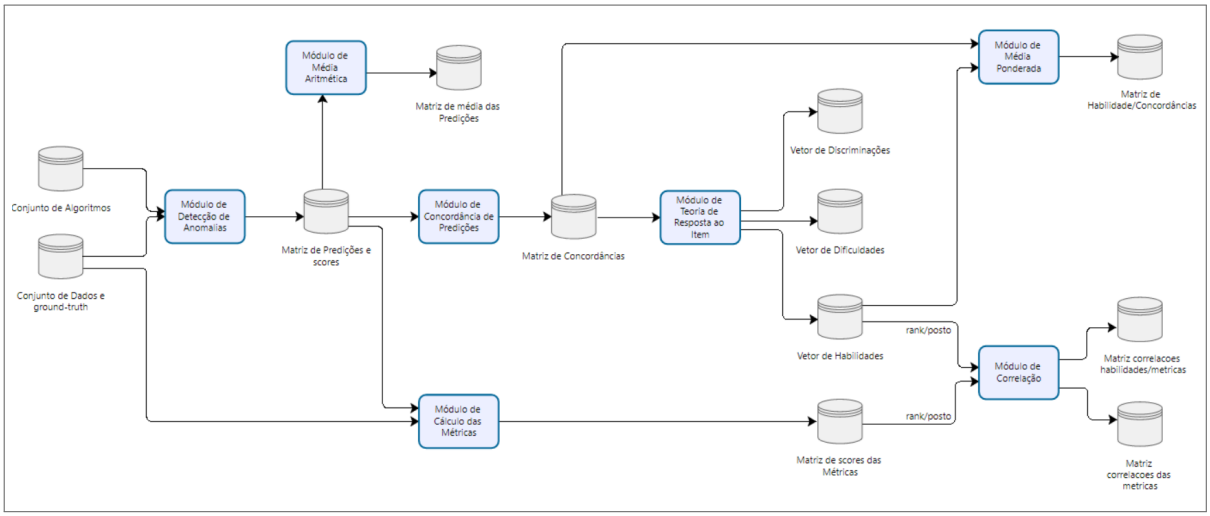
Em comparação com as abordagens que utilizam métricas convencionais em AI, como AUC ou *Precision* que fornecem apenas uma visão global do desempenho de um modelo, a IRT adiciona uma camada interpretativa, analisando o desempenho em nível de instância e atribuindo significado aos parâmetros de dificuldade, discriminação e habilidade. Contribuindo assim para a Explicabilidade de Modelos (XAI) (CARDOSO et al., 2022)(YEUNG, 2019)(FILHO et al., 2024), pois fornece explicações para as previsões de modelos de ML, ajudando a identificar quais exemplos influenciam determinadas decisões de um modelo. Ao combinar conceitos de psicometria com ML, oferece uma nova perspectiva para entender e avaliar o desempenho de sistemas de AI em um nível mais profundo e interpretável. Essa abordagem facilita diagnósticos detalhados e uma melhor comparação, entre diferentes modelos ou configurações.

3 AVALIAÇÃO DA DETECÇÃO DE ANOMALIAS BASEADA EM TEORIA DE RESPOSTA AO ITEM E EM MODELO DE CONCORDÂNCIA

Conforme mostrado no capítulo anterior, diferentes abordagens para avaliação de métodos de AD na literatura apresentam muitas limitações, seja pela falta de amostras de teste rotuladas ou por limitações específicas das técnicas de avaliação não supervisionadas. Portanto, este é um campo relevante e com potencial para novas soluções. Nesta dissertação, propomos um novo framework (ODAIRE) que adota IRT e concordância entre algoritmos, abordando assim desafios distintos associados à avaliação de AD. No ODAIRE, a IRT é adotada para avaliar simultaneamente as habilidades de um conjunto de modelos AD e as dificuldades das instâncias em um determinado conjunto de dados. A IRT tem se mostrado promissora em outros contextos de AI, mas é nova no contexto de avaliação de AD.

A Fig. 1 apresenta a solução proposta, que se baseia em um determinado conjunto de dados de interesse e em um conjunto de algoritmos para detectar anomalias. Inicialmente no Módulo AD, cada algoritmo AD do *pool* é adotado para detectar as anomalias no conjunto de dados. A matriz de predição produzida na primeira etapa é então fornecida como entrada para o Módulo de Concordância, que produzirá uma matriz de resposta para a IRT. Na proposta, cada respondente é um modelo AD no *pool* e cada item é uma instância. Cada resposta é o nível de concordância do respondente (em relação ao *pool*) ao prever se uma instância é uma anomalia ou não. Por fim, dada a matriz de respostas, o módulo IRT é responsável por estimar os valores latentes de habilidades (para respondentes) e dificuldades (para instâncias).

Figura 1 – *Pipeline* da Metodologia de Validação dos métodos de AD proposta.



Fonte: Autoria própria.

Altas habilidades são dadas aos modelos AD que são capazes não apenas de detectar anomalias mais fáceis em um conjunto de dados, mas também de detectar as mais difíceis. Na estrutura proposta, a concordância entre os modelos é considerada uma aproximação da resposta correta. Assim, são considerados casos difíceis quando existe um elevado nível de discordância entre os modelos, especialmente entre os bons. Isto é bastante diferente das abordagens existentes para avaliação da AD. Podemos apontar duas vantagens principais: (1) espera-se que as habilidades sejam mais robustas, uma vez que a IRT dá mais peso para os exemplos de teste mais difíceis durante a estimativa de habilidades; (2) a proposta é viável para avaliação de AD quando nenhum rótulo de classe é fornecida para calcular métricas supervisionadas.

3.1 MÓDULO DE DETECÇÃO DE ANOMALIAS

ODAIRE começa com o Módulo de AD, que basicamente adota um conjunto de M algoritmos $\mathcal{A} = \{A_j\}_{j=1}^{M}$ para detectar as anomalias em um conjunto de dados $\mathcal{D} = \{d_i\}_{i=1}^{N}$ de N instâncias. Cada algoritmo de detecção de anomalias é treinado no conjunto de dados e adotado para indicar suas anomalias na forma de previsões binárias (valor 1 para *outlier* e 0 para *inlier*),

Assim, este módulo retorna um $N \times M$ matrix P, que armazena as previsões p_{ij} feito pelos algoritmos para cada instância no conjunto de dados. Formalmente, $p_{ij} = 1$ se algoritmo A_j previu a instância d_i como um *outlier*; e $p_{ij} = 0$, caso contrário (ou seja, a instância é considerada uma inlier). Portanto, a entrada e a saída do Módulo AD podem ser resumidas como:

- **Entrada**: Dataset $\mathcal{D}$ e um conjunto de Métodos de AD $\mathcal{A}$.
- **Saída**: Matriz de predições P.

3.2 MÓDULO DE CONCORDÂNCIA

O próximo módulo do ODAIRE é o Módulo de Concordância, que transforma a matriz de predições P, recebida do Módulo AD, em uma matriz de resposta utilizada pelo modelo IRT.

Seguindo a nomenclatura IRT, os modelos AD são respondentes, enquanto as instâncias do conjunto de dados são itens. Conforme discutido no capítulo anterior, uma resposta r_{ij} na

IRT mede até que ponto o j-ésimo respondente responde corretamente ao i-ésimo item. No AD, entretanto, não é simples medir se um modelo está correto ou não para uma instância específica. Portanto, capturamos a noção de corretude na AD, baseando-nos na concordância entre os diferentes algoritmos do *pool*. Uma resposta de um modelo AD para uma instância específica será medida pela quantidade de concordância com os outros modelos em $\mathcal{A}$ ao prever essa instância como uma anomalia ou não. Portanto, o valor de cada célula da matriz de resposta do IRT para um dado algoritmo é a proporção da concordância com demais algoritmos de AD.

Formalmente, seja R uma matriz $N \times M$, armazenando cada resposta r_{ij} como uma medida de concordância do modelo A_j ao prever a instância d_i. Seja $a^i_{j,j'}$ o indicador de concordância entre modelos A_j e $A_{j'}$ ao prever a instância d_i, sendo denotado por:

$$a^i_{j,j'} = \begin{cases} 1, \ if \ p_{ij} = p_{ij'} \\ 0, \ caso \ contrário \end{cases} \tag{3.1}$$

onde $a^i_{j,j'} = 1$ quando ambos os modelos concordam que a instância é uma anomalia ou ambos os modelos concordam que a instância não é uma anomalia. De outra forma $a^i_{j,j'} = 0$ é uma indicação de desacordo entre os modelos.

Assim, a resposta r_{ij} modelada pelo ODAIRE é a proporção de concordância entre os modelos, como segue:

$$r_{ij} = \frac{1}{(M-1)} \sum_{j' \neq j} a^i_{j,j'} \tag{3.2}$$

Um valor de resposta alto r_{ij} indica um alto consenso de A_j para o conjunto de respondentes quando prevê especificamente a instância d_i. Se dois modelos forem bons, tenderão a concordar com as suas previsões. Portanto, esta metodologia vai além, fornecendo avaliação baseada em acordos em nível de instância. O acordo entre dois bons modelos deve impactar positivamente as habilidades de ambos.

A entrada e a saída do Módulo de Concordância podem ser resumidas como:

- **Entrada**: Matriz de predições P.
- **Saída**: Resposta/Concordância matrix R.

3.3 MÓDULO IRT

O terceiro módulo do *framework* proposto é o Módulo IRT, que estima a habilidade θ_j de cada modelo AD e a dificuldade δ_i de cada instância, dada a matriz de resposta de acordo R recebida do Módulo de Concordância.

Ao utilizar a IRT, a habilidade de um modelo é maior quando fornece uma melhor concordância, não apenas para instâncias fáceis, mas consistentemente para as difíceis. Esta é a diferença crucial entre o desempenho estimado, por exemplo, a partir de uma simples média da resposta do modelo entre as instâncias. Em outras palavras, a IRT pondera as instâncias de acordo com seus níveis de dificuldade. Tais aspectos têm sido abordados há décadas na IRT ao avaliar respondentes humanos, e recentemente abordados em ML supervisionado (MARTÍNEZ-PLUMED et al., 2016). A aplicação para avaliação de modelos de AD ainda não foi investigada na literatura.

Uma vez estimado o modelo IRT usando os dados da matriz de resposta, o ODAIRE retorna para cada instância uma dificuldade δ_i, o que reflete o quão difícil é para os modelos concordarem sobre como prever esta instância. Além disso, ODAIRE retorna uma habilidade θ_j para cada modelo, o que é alto para modelos que alcançam maior consenso na detecção, principalmente nas instâncias mais difíceis. A entrada e a saída deste módulo podem ser resumidas da seguinte forma:

- **Entrada**: Matriz de resposta R.
- **Saída**: Dificuldade δ_i de cada instância e habilidade θ_j de cada modelo.

3.4 OUTROS MÓDULOS

Os outros módulos exibidos no *pipeline* da metodologia proposta são: Módulo de cálculo das métricas, onde são calculadas as métricas clássicas de avaliação de Modelos AD; Módulo de correlação, em que são calculados os coeficientes de correlação, a partir dos postos (*ranks*) de *scores* obtidos em módulos anteriores; e os Módulos de média Aritmética/Ponderada, onde são calculadas as médias das predições e/ou concordâncias dos métodos AD. Mais informações sobre as funções presentes nestes módulos, bem como todos os seus respectivos artefatos de saída, serão descritos de forma detalhada apenas mais à frente, ao longo da seção 4.5, juntamente com todos os processos de funcionamento relacionados.

4 EXPERIMENTOS REALIZADOS

Neste capítulo, apresentamos os experimentos realizados para avaliar a solução proposta para avaliação da AD. Inicialmente, apresentamos informações relevantes sobre os conjuntos de dados adotados, os algoritmos utilizados para detectar anomalias em cada conjunto de dados, a implementação do da IRT e os *baselines* usados para comparação. No capítulo seguinte, apresentamos os resultados obtidos.

4.1 CONJUNTOS DE DADOS

Nos experimentos realizados, adotamos cinco conjuntos de dados artificiais e quinze conjuntos de dados reais, que serão descritos a seguir.

4.1.1 Dados Sintéticos

Os conjuntos de dados artificiais foram produzidos por meio da ferramenta *PyOD* (ZHAO; NASRULLAH; LI, 2019), que possui uma função nativa chamada *Data Generator*. Essa função permite criar padrões de baixa densidade e valores anômalos globais, considerados tarefas difíceis em algoritmos de AD. Todos os dados sintéticos foram obtidos na geração de dados com diversas variações nos valores dos parâmetros.

Um dos parâmetros desta função nativa do *PyOD*, é a 'Quantidade de *clusters*', o qual foram atribuídos os valores 2 e 3. Este parâmetro define o número de centros (ou seja, *clusters*) a serem gerados.

Outro parâmetro é a 'Contaminação', o qual foram atribuídos os valores 1%, 5% e 10%. Este parâmetro define a quantidade de contaminação do conjunto de dados, ou seja, a proporção de *outliers* no conjunto de dados.

Outro parâmetro também é a 'Tamanho dos *clusters*', o qual foram atribuídos os valores 'igual' e 'diferente'. Este parâmetro define o tamanho de cada *cluster*: 'igual' gera *clusters* com o mesmo tamanho, 'diferente' gera *clusters* com tamanhos diferentes.

Mais um outro parâmetro é a 'Densidade dos *clusters*', o qual foram atribuídos os valores 'igual' e 'diferente'. Este parâmetro define a densidade de cada *cluster*: 'igual' gera *clusters* com a mesma densidade, 'diferente' gera *clusters* com densidades diferentes.

Por fim um outro parâmetro é a 'Distância entre *clusters*', o qual foram atribuídos os valores 10%, 15% e 25%. Este parâmetro define a Distância entre *clusters*. A qual deve estar entre 0 e 1,0. Sendo usado para evitar que *clusters* se sobreponham o máximo possível. No entanto, se o número de amostras e o número de *clusters* forem muito altos, é improvável que eles sejam separados completamente, mesmo se parâmetro *dist* for definido como 1,0.

Também foram feitos alguns experimentos na geração de dados com Contaminação: 0%, apenas para o conjunto de treinamento (*Train dataset*) com o auxílio da ferramenta *Pandas*. Porém nem todas as configurações testadas foram utilizadas ao longo deste trabalho, apenas as que estão listadas no Quadro 1:

Quadro 1 – Configuração dos parâmetros na função geradora de dados artificiais

dataset	n_dimensions	n_clusters	contamination	size	density	dist
DS3	2	2	0,01	same	same	0,25
DS4	2	2	0,01	same	different	0,10
DS12	2	2	0,01	different	different	0,25
DS14	2	2	0,05	same	same	0,15
DS51	2	3	0,05	same	same	0,25

Fonte: Elaborada pelo autor (2024)

A definição de cada parâmetro da geração dos dados sintéticos pode ser encontrado na documentação da função *generate_data_clusters* em *PyOD* (ZHAO; NASRULLAH; LI, 2019), ou no trecho de código implementado, disponível no repositório[1] de código-fonte deste presente trabalho. Os parâmetros com valores fixos definidos na função geradora, foram: tamanho do conjunto de treinamento = 500; e tamanho do conjunto de teste = 250. A visualização da distribuição das instâncias, nos *datasets* artificiais gerados, está disponível na Figura2

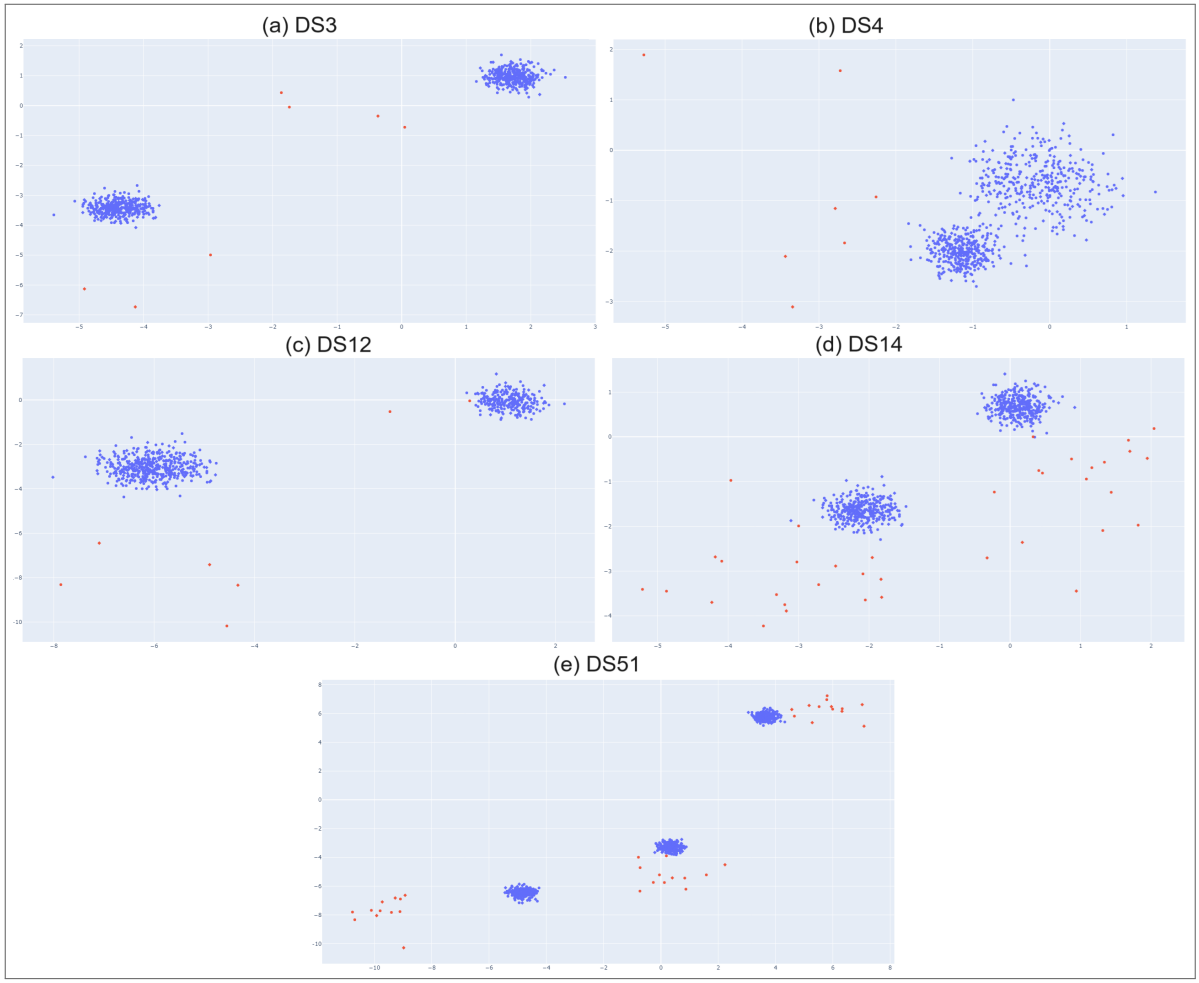
4.1.2 Dados Reais

Por sua vez, os conjuntos de dados reais foram coletados da base de dados *ODDS* (RAYANA, 2016), que se trata de uma grande coleção de conjuntos de dados de AD, com dados reais de acesso aberto. Essa base fornece conjuntos de dados de diferentes domínios, com dados multidimensionais. Na dissertação, foram utilizados os conjuntos listados na Tabela 1

As definições de cada base de dados listada, nas colunas desta tabela citada, são (da esquerda para a direita): nome do conjunto de dados; quantidade de atributos do *dataset*;

[1] https://github.com/ericlrf/dissertacao_eric_franca.git

Figura 2 – Conjuntos de dados artificiais gerados sinteticamente pela ferramenta *PyOD*. Os pontos em azul representam os *inliers*, já os *outliers* estão em vermelho.



Fonte: Elaborado pelo autor (2024)

quantidade de pontos de dados, ou tamanho do *dataset*; tamanho do sub-conjunto de dados para treino; tamanho do sub-conjunto de dados para teste; quantidade de pontos de dados rotulados como *outliers*, no sub-conjunto de dados para treino; e quantidade de pontos de dados rotulados como *outliers*, no sub-conjunto de dados para teste.

Devido a grande quantidade de exemplos em alguns *datasets*, para não dificultar a continuidade dos experimentos, foram realizadas sub-amostragem nos seguintes *datasets*: *annthyroid*, *cardio*, *musk*, *speech*, *thyroid*, e *vowels*. O novo número da quantidade reduzida dos pontos de dados, nestes seis *datasets*, ficaram em torno de mil instâncias para a soma total dos sub-conjutos de treino e teste.

Cada *dataset* possui peculiaridades e características que podem ser encontradas com mais profundidida na documentação da base de dados *ODDS* (RAYANA, 2016). Também, uma breve apresentação, adaptada da documentação original, está disponível no Anexo A .

Tabela 1 – Características dos *datasets* com dados reais utilizados.

dataset	dimensions	original_points	trainset	testset	train_outliers	test_outliers
annthyroid	7	7200	504	216	37	16
breastw	9	683	478	205	167	72
cardio	21	1831	640	275	62	26
glass	9	214	149	65	6	3
ionosphere	33	351	245	106	88	38
letter	32	1600	1120	480	70	30
musk	166	3062	704	307	22	10
pima	8	768	537	231	187	81
speech	400	3686	641	277	11	5
thyroid	6	3772	660	283	16	7
vertebral	6	240	168	72	21	9
vowels	12	1456	762	327	26	11
wbc	30	378	264	114	15	6
wine	13	129	90	39	7	3
yeast	8	1484	1038	446	63	27

Fonte: Elaborada pelo autor (2024)

4.1.3 Pré-processamento dos dados

Inicialmente, foram realizados o tratamento e limpeza dos dados nos *datasets* adotados. Porém, como se tratam de dados sintéticos artificialmente gerados e de datasets públicos conhecidos e bastante utilizados para *benchmarking*, não foi necessário realizar muitas intervenções como tratamento de valores nulos, valores categóricos, ou outros problemas comuns em dados. Foi realizado apenas a transformação dos dados nas diferentes dimensões, para uma mesma escala numérica, onde possibilita o correto funcionamento da maioria dos algoritmos de AD utilizados nessa dissertação.

Neste redimensionamento (*scaling*) ou "normalização"do tipo *Min-Max*, com o conjunto de dados de treinamento e validação, foi transformado cada atributo (*feature*) individualmente para um determinado intervalo, como por exemplo: entre zero (0) e um (1).

Existem algumas maneiras comuns de fazer com que todos os atributos de um *dataset* tenham a mesma escala, e uma delas é o dimensionamento *Min-Max*. Utilizamos isto subtraindo o valor mínimo, e dividindo pelo máximo menos o mínimo. A fórmula completa deste transformador está descrito na função 4.1:

$$X' = \frac{X - Xmin}{Xmax - Xmin} \tag{4.1}$$

O *Scikit-Learn* fornece uma implementação simples chamada *MinMax-Scaler*[2], já pronta para isto. Mais detalhes sobre o ambiente de execução *python* usado para executar os experimentos, e suas respectivas versões das bibliotecas importadas para realizar os experimentos, são apresentados no repositório[3] de código-fonte do presente trabalho.

Em seguida, para cada um dos *datasets* selecionados para este trabalho, foi utilizado o método *hold-out* para avaliação de modelos, seguindo a divisão (*split*) em treino/teste já descrita na seção 4.1, e para os dados reais públicos a mesma proporção 70/30 (% dos sub-conjuntos treino/teste).

Como já citado anteriormente, foram realizadas sub-amostragens em alguns conjuntos de Dados Reais, pois possuíam mais de mil instâncias, as quais exigiriam mais tempo de processamento e recursos computacionais. Portanto realizar todos os experimentos desta dissertação, mesmo com a divisão nos sub-conjuntos de treino e teste, seria algo difícil com os *datasets* ainda em seu tamanho original, devido ao desafio da grande quantidade de *datasets* e Algoritmos de detecção de *outliers* selecionados. Somado a isto, o processo de validação cruzada em que há técnicas com repetições, como nos métodos *k-fold* e *leave-one-out*, não foram utilizados neste trabalho pela mesma razão do tempo de processamento, que também ocasionaria um processo mais custoso para realizar tais experimentos.

4.1.4 Características dos dados

Todos os *datasets* (sintéticos ou reais) selecionados para treinamento dos Métodos AD, possuem algumas características intrinsecas em seus dados que, segundo a literatura, impactam o processamento de conjunto de dados em projetos de ML.

Uma dessas características é o conceito de Esparsidade, que ocorre quando a maioria dos elementos de um *dataset* são zeros ou valores nulos. Em outras palavras, numa matriz de dados, apenas uma pequena fração dos valores é não-nula. Uma das implicações disto é no desempenho dos Modelos AD, pois muitos algoritmos (como regressões com regularização L1 ou modelos baseados em árvores) podem se beneficiar da esparsidade, porém outros podem

[2] https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[3] https://github.com/ericlrf/dissertacao_eric_franca.git

precisar de cuidados especiais, pois a alta proporção de zeros pode influenciar na aprendizagem (BISHOP; NASRABADI, 2006).

Outro conceito é a Dimensionalidade, que refere-se ao número de atributos (ou *features*) presentes em um *dataset*. Um conjunto de dados de alta dimensionalidade possui muitas variáveis, o que pode introduzir desafios específicos. Suas implicações ocorrem à medida que o número de variáveis aumentam, pois o espaço de amostragem se expande exponencialmente. Isso pode levar a problemas como *Overfitting*, onde modelos podem se ajustar demais aos dados de treinamento, capturando ruídos em vez de padrões reais. E o problema do aumento do tempo computacional, pois processar e armazenar dados de alta dimensionalidade demanda mais recursos (BISHOP; NASRABADI, 2006).

Por fim, um outro conceito são os Ruídos, que se referem a informações incorretas, imprecisas ou irrelevantes, que podem surgir de erros de medição, entrada incorreta de dados, variações naturais ou outras fontes de interferência. Como implicações, há o impacto na qualidade do modelo, pois dados ruidosos podem levar a modelos que aprendem padrões errados ou que são sensíveis a flutuações aleatórias nos dados, resultando em desempenho inconsistente. Outra consequência dos ruídos, é o *Overfitting*, em que modelos complexos podem capturar o ruído presente nos dados, prejudicando a generalização para novos dados (BISHOP; NASRABADI, 2006).

Todos esses conceitos citados aqui, os quais podem ser inter-relacionados, não foram avaliados nos conjuntos de dados *ODDS* e *datasets* sintéticos adotados neste trabalho. Tampouco foram aplicadas estratégias de tratamento para tais características, durante pré-processamento dos dados, pois não se tratava do foco desta dissertação. Contudo, explorar tais conceitos nos *datasets* empregados nos experimentos, é um tema potencial para investigação em trabalhos futuros.

4.1.5 Características das anomalias

As anomalias presentes nos conjuntos de dados, utilizados neste trabalho de dissertação, possuem certas características que definem o seu tipo ou sua família. Há várias classificações segundo estudos na literatura (CHANDOLA; BANERJEE; KUMAR, 2009)(GOLDSTEIN; UCHIDA, 2016)(AGGARWAL, 2017), porém vamos destacar apenas algumas principais. Como por exemplo, os tipos de Anomalia Pontual (um único ponto de dado é anormal em relação ao restante), Anomalia Global (o valor é anômalo em relação ao conjunto total), e Anomalias Sintéticas

(inseridas manualmente em datasets para testar modelos de detecção).

Há também a família de anomalias baseadas em ML, que dependem de modelos treinados para aprender padrões normais, onde o modelo aprende o que é "normal" e detecta padrões que não se encaixam. A qual possui o subgrupo dos não-supervisionados, que detectam anomalias sem rótulos, baseadas em distâncias, densidade ou agrupamento. Nesse sentido, há as Anomalias por distância (em que pontos de dados estão longe dos outros em termos de distância, geralmente Euclidiana, no espaço das variáveis), Anomalias por densidade (onde um ponto numa região de baixa densidade comparado aos seus vizinhos locais), entre outras.

Entretanto, não foram escolhidos os *datasets* de acordo com os conceitos dessas características, tampouco foram investigadas as consequências dessas escolhas, nos testes realizados com detecção de *outliers*. Portanto, a exploração deste tema nos dados escolhidos, é um tópico relevante para prosseguimento nos trabalhos futuros.

4.2 ALGORITMOS DE AD

Vários algoritmos AD não supervisionados bem conhecidos e amplamente utilizados na literatura, foram empregados em nossos experimentos. Adotamos algoritmos de diferentes famílias de métodos para ajustar os dados sob uma variedade de suposições, otimizando diferentes critérios, levando assim a possíveis instâncias diferentes detectadas como anomalias. Os algoritmos adotados estão disponíveis em ferramentas padrão para AD, como a biblioteca *PyOD* (ZHAO; NASRULLAH; LI, 2019), *ADBench* (HAN et al., 2022) e *Scikit-learn* (PEDREGOSA et al., 2011). A lista de todos os algoritmos AD utilizados é mostrada na Tabela 2, agrupados por tipo de algoritmo. Devido ao elevado número de algoritmos, descrevemos apenas de forma breve cada um dos métodos de detecção de *outliers* no Anexo B , bem como as respectivas referências para uma completa documentação das bibliotecas implementadas, bem como alguns artigos relacionados para mais detalhes.

De acordo com a literatura, há várias organizações dos algoritmos AD em grupos, ou famílias de algoritmos, que nada mais são do que os tipos de algoritmos que compartilham características semelhantes (CHANDOLA; BANERJEE; KUMAR, 2009)(AGGARWAL, 2017). Como por exemplo os algoritmos de detecção de *outliers* baseados em Distância, Densidade, Reconstrução, entre outros. As famílias de algoritmos foram selecionadas no intuito de gerar diversidade de resultados, a partir de diferentes técnicas de detecção. E não com o objetivo de obter um melhor desempenho nos resultados, como observado nas seleções de tipos de algorit-

Tabela 2 – Métodos de AD que foram adotados, agrupados por bibliotecas de implementação.

Tipo de Algoritmo	Nome do Método AD	Implementação
Linear Model	*Principal Component Analysis* (PCA) *Outlier Detector*	PyOD
Linear Model	*One-Class Support Vector Machines* (OCSVM)	PyOD
Linear Model	*Cook's distance for outlier detection* (CD)	PyOD
Proximity-based	*Local Outlier Factor* (LOF)	PyOD
Proximity-based	*Clustering-Based Local Outlier Factor* (CBLOF)	PyOD
Proximity-based	*Connectivity-Based Outlier Factor* (COF)	PyOD
Proximity-based	*Histogram-based Outlier Score* (HBOS)	PyOD
Proximity-based	*K-Nearest Neighbor* (KNN)	PyOD
Proximity-based	*Subspace Outlier Detection* (SOD)	PyOD
Proximity-based	*Fast outlier detection using the Local Correlation Integral* (LOCI)	PyOD
Probabilistic	*Copula-Based Outlier Detection* (COPOD)	PyOD
Probabilistic	*Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions* (ECOD)	PyOD
Probabilistic	*Outlier detection with Kernel Density Functions* (KDE)	PyOD
Probabilistic	*Probabilistic Mixture Modeling for Outlier Analysis* (GMM)	PyOD
Outlier Ensembles	*Isolation Forest* (IForest)	PyOD
Outlier Ensembles	*Lightweight On-line Detector of Anomalies* (LODA)	PyOD
Outlier Ensembles	*Locally Selective Combination of Parallel Outlier Ensembles* (LSCP)	PyOD
Neural Networks	*Deep One-Class Classification* (DeepSVDD)	PyOD
Neural Networks	*Fully connected AutoEncoder* (AutoEncoder)	PyOD
Neural Networks	*Variational AutoEncoder* (VAE)	PyOD
Neural Networks	*Anomaly Detection with Generative Adversarial Networks* (AnoGAN)	PyOD
Neural Networks	*Multiple-Objective Generative Adversarial Active Learning* (MO_GAAL)	PyOD
Neural Networks	*Single-Objective Generative Adversarial Active Learning* (SO_GAAL)	PyOD

Fonte: Elaborada pelo autor (2024)

mos para compor *Emsembles* (KANDANAARACHCHI, 2022)(KANDANAARACHCHI; SMITH-MILES, 2023). E como impacto dessa diversificação de distintas famílias de algoritmos, temos uma satisfatória variabilidade nos resultados, dentre os muitos *datasets* utilizados. A avaliação da influência na escolha desses tipos de algoritmos AD, nos experimentos deste trabalho, pode ser objeto de estudo a ser mais aprofundado em trabalhos futuros.

4.2.1 Parâmetros dos Modelos

Ao usar os valores padrão dos parâmetros e/ou hiperparâmetros de cada modelo, muito provavelmente não será obtido o melhor desempenho para esses modelos durante treinamento. Ajustar tais parâmetros para encontrar um modelo mais adequado é uma prática comumente necessária. Então inicialmente buscou-se encontrar duas configurações distintas (conjunto de valores para ajuste fino dos parâmetros e/ou hiper-parâmetros num modelo de AD para cada um dos modelos de AD, seja por meio de uma busca aleatória, ou em grade dos melhores parâmetros a serem ajustados. O que duplicaria o conjunto de métodos de AD a serem utilizados nos experimentos.

Contudo, como a otimização de hiperparâmetros é uma tarefa muito custosa e exponencialmente demorada, além de também não ser o foco deste trabalho, os parâmetros dos modelos foram então selecionados manualmente. Apenas uma única configuração de parâmetros e hiperparâmetros foi usada, e mesmo assim pode fornecer bons resultados em menor tempo de processamento.

Todos os métodos de AD implementados a partir das bibliotecas: *PyOD* [4] ; *ADBench* [5] ; e *Scikit-learn* [6] , conforme já citado anteriormente, adotaram as configurações padrões (*default*) de valores para os parâmetros, segundo suas respectivas documentações oficiais.

Com exceção apenas para os seguintes algoritmos de AD, listados a seguir, que foram explicitamente definidos estes seguintes parâmetros:

- ***Fully connected AutoEncoder* (AutoEncoder)**: *hidden_neurons*=[**N*, 32, 32, **N*]
- ***Variational AutoEncoder* (VAE)**: *encoder_neurons*=[**N*, 32], *decoder_neurons*=[32, **N*]

[4] https://pyod.readthedocs.io/en/latest/pyod.models.html
[5] https://github.com/Minqi824/ADBench
[6] https://scikit-learn.org/stable/modules/outlier_detection.html

- ***Locally Selective Combination of Parallel Outlier Ensembles* (LSCP)**: *detector_list* = [LOF(), LOF()]

Onde **N* representa o número (inteiro) de dimensões do *dataset*, no qual o algoritmo está sendo ajustado. Também para garantir a reprodutibilidade, foi definida uma semente (*random seed*) para todas as tarefas que possam ter um fator aleatório, como as divisões de treino/teste dos conjuntos de dados e os próprios modelos. Um exemplo da definição dos parâmetros e hiperparâmetros atribuídos para todos os métodos de detecção de *outiliers* utilizados, em apenas um dos datasets selecionados para os experimentos, encontra-se no repositório[7] de código-fonte deste trabalho.

4.2.2 Treinamento dos Modelos

Após a definição e configuração dos modelos AD, já apresentados anteriormente, é iniciado então a execução do *pipeline* do ODAIRE. A partir do primeiro Módulo AD, é recebido como *input* para cada único *Dataset* ($\mathcal{D}$), um conjunto de Métodos de AD ($\mathcal{A}$). E então são realizados os treinamentos (*fit*), ou ajustes dos métodos de detecção, em seus respectivos conjuntos de dados (*train subset*). Em todos eles, foi utilizada uma mesma semente de aleatoriedade, para reprodutibilidade futura.

Esta etapa de execução dos experimentos foi o mais longa e demorou mais tempo, devido o custo computacional em ajustar diversos algoritmos de detecção de *outliers*, para diversos *datasets* em determinado período de tempo. Assim, o tamanho de alguns conjuntos de dados foi reduzido para aproximadamente 1000 (mil) instâncias, facilitando assim o andamento dos experimentos e um tempo de execução mais viável. Também não foram realizados busca por otimização de modelos, a partir de diferentes valores dos hiperparâmetros.

Em seguida são geradas os arquivos Valores Separados por Vírgulas (do inglês *Comma-Separated Values*) (CSV) contendo as matrizes de predições obtidas, de cada um dos Métodos AD adotados, obtidas em todas as instâncias em seu respectivo *dataset*. Predições estas, que serão usadas como entrada pelo Módulo de Concordância.

Também foram armazenadas, neste mesmo sistema de arquivos estruturados, as respectivas matrizes de predições em formato de *score* (valor contínuo, e não um rótulo binário), para fins de outros testes nos experimentos a serem realizados.

[7] https://github.com/ericlrf/dissertacao_eric_franca.git

Um exemplo dessas Matrizes de predições entre modelos de AD, pode ser observada no repositório[8] de código-fonte deste presente trabalho. Onde cada coluna representa um Algoritmo de AD, e cada linha suas respectivas predições (ou valor de *score* do algoritmo) para determinadas instâncias, as quais pertencem à algum conjunto de dados escolhido.

Por fim, foram gerados vários arquivos individuais destas matrizes para cada *dataset* de dados sintéticos e reais, separados por seus sub-conjuntos de treino e teste, num estruturado sistema de arquivos e diretórios que facilitam sua reutilização durante os experimentos desenvolvidos.

4.2.3 Predições no conjunto de testes

Após o ajuste dos métodos de *AD*, cada modelo é usado para realizar as predições do valor dos rótulos (*outlier* ou *inlier*) das instâncias do conjunto de teste, gerando assim a matriz P para o próximo passo do *framework* proposto. Como os modelos já estão ajustados e não será necessário treiná-los novamente, apenas identificar quais instâncias do sub-conjunto de dados de teste (de tamanho menor), são de fato anomalias, o tempo de execução foi bastante curto, em relação ao tempo de treinamento anteriomente, alcançando em média no máximo 10 minutos.

4.3 MÓDULO DE CONCORDÂNCIA

Em seguida, apos a realização do treinamento de todos os métodos de AD selecionados, em cada um dos datasets sintéticos e reais escolhidos, assim como a geração da matriz de predições, foi finalmente executado o segundo módulo do ODAIRE, conforme o descrito na Seção 3.2, chamado Módulo de Concordância.

Assim como já mencionado anteriormente, é carregada uma matriz de predições como entrada, onde para cada *dataset* sintético ou real (sub-conjuntos de treino ou teste) há um arquivo .CSV gerado no Módulo AD anterior, o qual contém as predições de cada um dos métodos AD adotados, que foram obtidas em todas as instâncias do seu respectivo conjunto de dados selecionado.

As predições são processadas para gerar uma matriz de concordância para cada *dataset*

[8] https://github.com/ericlrf/dissertacao_eric_franca.git

utilizado neste experimento. O código-fonte deste algoritmo está disponível no repositório[9] de código-fonte deste trabalho, e seu funcionamento já foi descrito na seção 3.2 que descreve todo o funcionamento do Módulo de Concordância presente no ODAIRE.

Em seguida são geradas os arquivos .CSV contendo os matrizes de concordância obtidas, as quais serão usadas como entrada pelo Módulo IRT. Um exemplo dessa matriz concordância entre modelos de detecção de *outlier*, pode ser observada no Quadro 2. Onde cada coluna representa um Modelo de AD, e em cada linha sua respectiva proporção de concordâncias entre as predições de determinada instância, pertencente à um conjunto de dados escolhido.

Quadro 2 – Trecho da Matriz de Concondância derivada das predições obtidas no conjunto de dados *DS4*.

PCA	OCSVM	LOF	CBLOF	COF	HBOS	KNN
1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.952	0.952	0.952	0.954	0.000	0.950	0.952
1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.904	0.904	0.904	0.909	0.909	0.900	0.904
0.952	0.952	0.952	0.954	0.954	0.950	0.952
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.904	0.904	0.000	0.909	0.909	0.900	0.904
0.857	0.047	0.857	0.863	0.863	0.850	0.857
1.000	1.000	1.000	1.000	1.000	1.000	1.000

Fonte: Elaborado pelo autor (2024)

Por fim, foram gerados os arquivos individuais destas matrizes para cada um dos conjuntos de dados sintéticos e reais utilizados, separados por sub-conjuntos de treino e teste.

4.4 MODELO IRT

Finalmente, o terceiro e último módulo do ODAIRE é o Módulo de IRT, já descrito anteriormente na seção 3.3, é então implementado em *python* para estes experimentos. Inicialmente são importados em *notebook Jupyter*, os arquivos .CSV com as matrizes de concordância produzidas no módulo anterior. Em seguida, é instalada no ambiente *python* utilizado, a biblioteca *BIRT-gd*[10], que é uma implementação em linguagem *python* do β*4-IRT* (já descrito

[9] https://github.com/ericlrf/dissertacao_eric_franca.git
[10] https://pypi.org/project/birt-gd/

anteriormente noutro capítulo) com o uso de *Gradient Descent* (ou Método do Gradiente, bastante utilizado na literatura para otimização em ML).

Após feita instalação do *BIRT-gd*, é então possível importar a classe ***Beta4***, a qual será parametrizada para gerar o modelo IRT contínuo 3PL, que será ajustado aos dados de entrada deste Módulo *IRT*, definido em nossa solução ODAIRE.

Foram ajustados neste modelo, apenas os parâmetros para quantidade de respondentes, e a quantidade de itens de avaliação. Os outros parâmetros restantes da classe *Beta4*, foram definidos com os valores *default* (padrão) segundo a documentação[11] de sua respectiva biblioteca.

Por fim, após definido corretamente o modelo, é feito o seu ajuste aos conjuntos de dados selecionados neste trabalho. A execução tem um tempo de duração variável, a depender do tamanho (quantidade de instâncias) e complexidade (quantidade de dimensões) do *dataset* utilizado, bem como os recursos da máquina físico (processador, memória, etc) onde foi processado o código desenvolvido.

Após o modelo IRT ajustado no *dataset*, é possível obter os traços latentes de habilidade (θ_j) dos respondentes (Métodos de detecção de *outliers*), e dificuldade (δ_i) e discriminação (α_i) dos itens de avaliação (instâncias de um conjunto de dados).

No repositório[12] de código-fonte deste trabalho, é possível visualizar os traços latentes obtidos após a convergência do modelo *BIRT-gd* com os dados do *dataset* sintético *DS4*. Como por exemplo vetor de Habilidades, onde os mais de 20 algoritmos AD obtiveram valores muito altos (próximos de 1).

Já quanto ao vetor de Dificuldades, ainda referente ao *dataset* DS4, pode-se notar que no trecho destacado, apenas uma instância obteve valor de dificuldade um pouco maior, referente à concordância entre os modelos AD, se os itens de avaliação eram considerados ou não *outliers*. Quanto ao vetor de Discriminação, também seguiu o mesmo comportamento de Dificuldades.

4.5 *BASELINES*

Após concluídas todas as etapas dos módulos ODAIRE, foi iniciada a avaliação dos Modelos AD e também dos *datasets* utilizados como *benchmarking*. A partir dos rótulos (*labels*)

[11] https://github.com/Manuelfjr/birt-gd
[12] https://github.com/ericlrf/dissertacao_eric_franca.git

originais, também chamados de GT, que já classificam *a priori* qual é o tipo (anomalia ou não) de cada instância dos conjuntos de dados. Seja o atributo preditor fornecido pelo gerador de dados sintéticos (ferramenta *PyOD*) ou pela base de dados reais pública (*ODDS*), foram carregado as 'respostas' para as detecções de anomalias, como *baselines* para as seguintes comparações entre as métricas de avaliação de algoritmos de ML, ou traços latentes de IRT, que realizaremos neste experimento.

4.5.1 Métricas de avaliação

Inicialmente, após concluídos os treinamentos dos modelos AD, e suas respectivas predições de anomalias nos *datasets* escolhidos: foram então realizados os cálculos com as diferentes medidas, já apresentadas anteriormente nas seções 2.2.1 e 2.2.2, para comparar assim o desempenho dos diferentes modelos adotados.

Para tal, foi elaborado um *script* também em linguagem *python*, para carregar os arquivos .CSV com as predições de cada um dos conjuntos de dados sintéticos e/ou reais, juntamente com os arquivos originais de seus respectivos GT. Num primeiro momento, apenas para os subconjuntos de dados de treino, depois para os dados de teste. Em seguida foram utilizadas as funções da biblioteca *Sklearn*, que possui a implementação para calcular as diferentes métricas clássicas (AUC, *Precision*, *Recall*, entre outros) de avaliação de modelos, para cada um dos Métodos AD dentre os dados de predições carregados.

No repositório[13] de código-fonte desta dissertação, é possível visualizar o *script* desenvolvido para gerar de forma automática os valores das diferentes métricas de avaliação (AUC, acurácia, etc) entre os vários Algoritmos de AD, para todos os *datasets* da base *ODDS* adotadas. A saída deste script é então gravada num arquivo .CSV com todos os *scores* das medidas de avaliação, para cada um dos conjuntos de dados. Um trecho deste artefato pode ser visualizado também no repositório de código-fonte deste trabalho . Onde cada coluna representa uma diferente medida de avaliação de modelos, e cada linha representa o valor das métricas para um determinado Modelo (tipo de algoritmo) de detecção de *outliers*.

[13] https://github.com/ericlrf/dissertacao_eric_franca.git

4.5.2 Ranqueamentos de Métodos AD

Os passos deste processo de ranqueamento se iniciam com o carregamento dos arquivos .CSV, contendo as matrizes com os valores das métricas de avaliação de Métodos AD, de todos os conjuntos de dados utilizados no experimento. Em seguida um *script* é executado para atribuir a posição de cada valor de *score*. Foi utilizada a função *.rank()* da biblioteca *Pandas*[14] com o parâmetro *method* = *'max'*, pois empiricamente apresentou-se como a melhor abordagem, para as situações com repetição dos mesmos valores de *scores* para uma mesma medida de avaliação de modelos, em que registros que têm os mesmos valores são classificados usando a classificação mais alta.

No repositório de código-fonte deste trabalho, é possível visualizar este *script* desenvolvido, para gerar de forma automática: estes valores de *Rank* (posição ordenada) entre os *scores* obtidos nos cálculos das métricas de avaliação dos Modelos; e também os arquivos externos com as matrizes contendo o ranqueamento dos *scores* das métricas de avaliação. Um trecho deste artefato, da matriz de ranqueamento para os valores das medidas de avaliação obtidos no *dataset Pima*, pode ser visualizado no Quadro 3.

Quadro 3 – Trecho da Matriz de ranqueamento das métricas obtidas nas avaliações dos Métodos AD ajustados no conjunto de dados reais *Pima*.

.	AUC_score	PRECISION_score	RECALL_score	FSCORE_score
PCA	10	10	10	10
OCSVM	11	11	11	11
LOF	15	15	15	15
CBLOF	15	15	15	15
COF	5	5	5	5
HBOS	**2**	**2**	**2**	**2**
KNN	20	20	19	19

Fonte: Elaborado pelo autor (2024)

4.5.3 Correlação das Métricas de Métodos AD

Agora neste processo de correlação dos valores de ranqueamento, derivado dos *scores* das Medidas de avaliação (AUC, *Precision*, ou *Recall*), foram realizados vários testes para obtenção de *insights*, fundamentais para a evolução dos experimentos. Inicialmente foram

[14] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rank.html

lidos os arquivos contendo as Matrizes de ranqueamento, tanto para os conjuntos de dados sintéticos quanto para os reais.

Em seguida foi desenvolvido um *script*, que ao ser executado calcula a correlação entre as diferentes listas de valores, referentes às Métricas de avaliação de modelos de AD. Foi utilizada a função *.corr()* da biblioteca *Pandas*[15] com o parâmetro *method = 'spearman'*, pois julgou-se mais adequado para esta situação utilizar o Coeficiente de correlação de postos de *Spearman*.

É possível ver parte deste *script* no repositório[16] de código-fonte do presente trabalho. O resultado é geração de várias Matrizes de correlação de *Spearman*, para cada dos conjuntos de dados escolhidos. Graças a grande quantidade de diferentes Modelos de AD e de Medidas de avaliação de modelos, foram obtidos alguns resultados robustos para comparações entre as matrizes de correlação. Na Figura 22 do Apêndice B é possível visualizar tal matriz.

No Quadro 4 é possível visualizar a correlação entre as Medidas de avaliação de Modelos AD, referente ao ranqueamento dos *scores* obtidos no *dataset* DS4. Já no Quadro 5, vemos a correlação entre as Medidas de avaliação referente ao *dataset* com dados reais *Pima*. Por fim, tais matrizes com as correlações foram salvas em arquivos .CSV no diretório de arquivos utilizado neste trabalho, para posterior reutilização nos experimentos.

Quadro 4 – Trecho de Matriz de correlação entre Métricas de avaliação de Métodos AD ajustados no conjunto de dados artificiais *DS4*.

.	AUC	PRECISION	RECALL	FSCORE	AUCPR	ACC	LOGLOSS
AUC	1.00	0.91	**0.95**	0.91	**1.00**	0.89	-0.89
PRECISION	0.91	1.00	0.82	**1.00**	0.91	**0.99**	-0.99
RECALL	**0.95**	0.82	1.00	0.82	**0.95**	0.81	-0.81
FSCORE	0.91	**1.00**	0.82	1.00	0.91	**0.99**	-0.99
AUCPR	**1.00**	0.91	**0.95**	0.91	1.00	0.89	-0.89
ACC	0.89	**0.99**	0.81	**0.99**	0.89	1.00	-0.99
LOGLOSS	-0.89	-0.99	-0.81	-0.99	-0.89	-0.99	1.00

Fonte: Elaborado pelo autor (2024)

[15] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html
[16] https://github.com/ericlrf/dissertacao_eric_franca.git

Quadro 5 – Trecho de Matriz de correlação entre Métricas de avaliação de Métodos AD ajustados no subconjunto de treino de dados reais *Pima*.

.	AUC	PRECISION	RECALL	FSCORE	AUCPR	ACC	LOGLOSS
AUC	1.00	1.00	0.99	0.99	1.00	0.96	-0.96
PRECISION	1.00	1.00	0.99	0.99	1.00	0.96	-0.96
RECALL	0.99	0.99	1.00	1.00	0.99	0.92	-0.92
FSCORE	0.99	0.99	1.00	1.00	0.99	0.92	-0.92
AUCPR	1.00	1.00	0.99	0.99	1.00	0.96	-0.96
ACC	0.96	0.96	0.92	0.92	0.96	1.00	-0.99
LOGLOSS	-0.96	-0.96	-0.92	-0.92	-0.96	-0.99	1.00

Fonte: Elaborado pelo autor (2024)

4.5.4 Comparação de combinações da Matriz de Predições, Concordâncias, com o Traço latente de IRT

Neste momento do trabalho, foram realizados de forma exploratória alguns testes, para comparação das predições dos Modelos AD e suas respectivas habilidades (IRT), bem como incluindo também comparações com matrizes de concordância, recém obtidas, e seus respectivos traços latentes (IRT).

Primeiramente, foi elaborado um *script* em linguagem *python*, para calcular a média aritmética das predições de um conjunto de Métodos AD, em cada uma das instâncias de um determinado *dataset*. Em seguida, foi também elaborado *script* para a média aritmética da quantidade de Concordâncias entre os Métodos de AD, para cada instância de um *dataset* escolhido, onde ambos os códigos-fontes estão disponíveis no repositório deste trabalho.

A fórmula matemática para representar estas médias aritmeticas, é dada por: Seja uma lista $X = [x_1, x_2, x_3, ..., x_N]$ onde cada elemento x_i representa uma instância na Matriz de Predições (ou na Matriz de Concordância). Instância esta, a qual representa uma lista com M valores, ou seja, uma lista interna $x_i = [x_{i1}, x_{i2}, ..., x_{iM}]$, onde i representa a posição na lista de Métodos AD utilizados, que é interna dentro de superlista X. Por fim, a fórmula para a média aritmética dos valores contidos apenas numa lista interna x_i é descrita na função 4.2:

$$\bar{x}_i = \frac{1}{M} \sum_{j=1}^{M} x_{ij} \tag{4.2}$$

onde:

- x_{ij} é o valor na posição j da lista interna x_i,

- M é o número total de valores na lista interna x_i,
- $\sum_{j=1}^{M} x_{ij}$ representa a soma de todos os valores da lista interna x_i.

Assim, a média aritmetica $\bar{x}_i$ é calculada somando todos os valores da lista x_i e dividindo pelo total de valores M nesta lista.

Após obtidos os valores dessas médias calculadas, foram contadas as frequências dos valores únicos em ordem decrescente. As quais são referentes tanto para uma Matriz de Predições, quanto para uma Matriz de Concordância, como pode-se observar no Quadro 6.

Segundamente, foi elaborado um *script* também em *python*, para calcular a média ponderada das predições de um conjunto de Métodos AD, em cada uma das instâncias de um determinado *dataset*, como pode ser observado no código-fonte disponível no repositório[17] do presente trabalho. Em seguida, foi também foi elaborado mesmo *script* para a média ponderada da quantidade de Concordâncias entre os Métodos de AD, para cada instância de um *dataset* escolhido.

Ambos os códigos, utilizaram como pesos os valores do traço latente *Habilidade* do modelo IRT ajustado anteriormente. Onde cada *score* de habilidade θ, referente à um Método de AD, é multiplicado como um peso em seus respectivos valores de predição binária (ou quantidade de concordâncias), para cada uma das instâncias de um *dataset*. Um exemplo deste vetor de habilidades, obtido como traço latente no ajuste de um modelo IRT, pode ser observado na Figura 17.

A fórmula matemática para representar estas médias ponderadas, é dada por: Seja uma lista $X = [x_1, x_2, x_3, ..., x_N]$ onde cada elemento x_i representa uma instância na Matriz de Predições (ou na Matriz de Concordância). Instância esta, a qual representa uma lista com M valores, ou seja, uma lista interna $x_i = [x_{i1}, x_{i2}, ..., x_{iM}]$, onde i representa a posição na lista de Métodos AD utilizados, que é interna dentro de superlista X. Por fim, a fórmula para a média ponderada dos valores contidos apenas numa lista interna x_i é descrita na função 4.3:

$$\bar{x}_i = \frac{1}{M} \sum_{j=1}^{M} \theta_j x_{ij} \tag{4.3}$$

onde:

- θ_j é o valor da Habilidade (traço latente), do respectivo Método AD na posição j, obtido do modelo IRT ajustado no conjunto de instâncias X,

[17] https://github.com/ericlrf/dissertacao_eric_franca.git

- x_{ij} é o valor na posição j da lista interna x_i,
- M é o número total de valores na lista interna x_i,
- $\sum_{j=1}^{M} \theta_j x_{ij}$ representa a soma de todos os valores, multiplicados por seus pesos (habilidades), da lista interna x_i.

Assim, a média ponderada $\bar{x}_i$ é calculada somando todos os valores da lista x_i e dividindo pelo total de valores M nesta lista.

Após obtidos os valores dessas médias calculadas, foram contadas as frequências dos valores únicos em ordem decrescente. As quais são referentes tanto para uma Matriz de Predições, quanto para uma Matriz de Concordância, como pode-se observar no Quadro 7.

A partir desses novos valores obtidos durante os cálculos das médias ponderadas, foi possível associá-los às instâncias de um conjunto de dados, para estimar o grau de habilidade na identificação ou não, de *outliers* corretamente. Sendo possível estimar regiões de instâncias pontuais, em visualizações de planos bidimensionais, que houveram maior efetividade da detecção de anomalias entre o conjunto de Modelos AD adotados nos experimentos. Tais resultados desses testes com média ponderada entre predições, concordâncias, e habilidades de IRT, serão descritos no Capítulo 5.

4.5.5 Ranqueamento de Traço latente de IRT

Os passos deste processo de ranqueamento se iniciam com o carregamento dos arquivos .CSV, contendo os vetores com os valores do traço latente de Habilidade do modelo IRT, anteriormente ajustado individualmente em cada conjunto de dados utilizado neste trabalho. Em seguida um *script* é executado para atribuir a posição de cada valor de habilidade do IRT. Foi novamente utilizada a função *.rank()* da biblioteca *Pandas* com o parâmetro *method* = *'max'*.

No repositório[18] de código-fonte deste trabalho, é possível visualizar um *script* desenvolvido, para gerar de forma automática: estes valores de *Rank* (posição ordenada) entre os Habilidades obtidas para cada Método AD. Neste mesmo código, também são gerados os arquivos externos com os *datasets* contendo o ranqueamento das Habilidades de IRT entre os Métodos de detecção de *outliers*, separados por entre os diferentes conjuntos de dados

[18] https://github.com/ericlrf/dissertacao_eric_franca.git

artificiais ou reais. Um trecho deste artefato, do *dataset* de ranqueamento para os valores do traço latente de Habilidades, obtidos no ajuste do modelo IRT no *dataset* Pima, pode ser visualizado na Figura 21 do Apêndice B.

4.5.6 Correlação das Métricas de avaliação e Traço latente de IRT

Finalmente, a partir de agora são realizados os processos de correlação dos valores de ranqueamento, derivados tanto do traço latente de Habilidade quanto das Medidas de avaliação de modelos (que exigem a variável resposta, ou GT) , também foram realizados vários testes para obtenção de *insights* durante os experimentos. Inicialmente foram lidos os arquivos contendo as Matrizes de ranqueamento, referente as Métricas de avaliação, bem como as Habilidades do modelo IRT, para cada um dos *datasets* utilizados (ambos os conjuntos de dados sintéticos, quanto para os reais públicos).

Em seguida foi desenvolvido um *script*, que ao ser executado calcula a correlação entre os diferentes ranqueamentos, relacionados ao conjunto de distintos Modelos de AD. Foram utilizadas as funções *pearsonr* e *spearmanr*, ambas da biblioteca *scipy.stats* [19], pois julgou-se importante para esta situação avaliar o Coeficiente de correlação de postos de *Spearman* (não-paramétrico, para varíaveis ordinais), juntamente com o Coeficiente de correlação de *Pearson* (assume uma distribuição Normal, para variáveis quantitativas).

É possível ver tal *script* no repositório[20] de código-fonte deste trabalho, como é feita a geração de gráficos de dispersão demonstrando a correlação entre as variáveis, bem como a estatística de cada coeficiente de correlação adotado. Novamente. em virtude da grande quantidade de diferentes Modelos de AD e de Medidas de avaliação de modelos, foram obtidos resultados mais robustos para comparações entre as correlações.

Também foi desenvolvido um código em *script*, para automatizar a geração de matriz com as estatísticas de todos os coeficientes de correlação de *Spearman*, entre o traço latente (IRT) de Habilidade e as Medidas de avaliação de Métodos AD, agrupados por todos os conjuntos de dados reais e artificiais. Um trecho deste *script* também está disponível no repositório de código-fonte.

No capítulo 5 serão apresentados todas estas correlações geradas, entre as Medidas de avaliação de Modelos AD versus Traço latente de Habilidade do IRT, referente ao ranqueamento

[19] https://docs.scipy.org/doc/scipy/reference/stats.html#statistical-functions-scipy-stats
[20] https://github.com/ericlrf/dissertacao_eric_franca.git

dos valores obtidos nos diversos *datasets* adotados.

5 RESULTADOS DOS EXPERIMENTOS

Agora neste capítulo, apresentamos os resultados dos experimentos realizados no presente trabalho. Inicialmente, apresentamos a correlação entre a métrica de habilidade estimada pelo ODAIRE e outras medidas de avaliação clássicas já existentes na literatura. Também comparamos os diferentes níveis de habilidade e dificuldade, entre os algoritmos de AD empregados nas diferentes regiões dos conjuntos de dados utilizados.

5.1 CORRELAÇÃO ENTRE ODAIRE E OUTRAS MÉTRICAS

Nesta próxima sessão de experimentos, calculamos a correlação entre a habilidade e diferentes critérios de validação supervisionados, conforme mostrado na Tabela 3 para todos os conjuntos de dados utilizados (sintéticos e reais). E apresentando na última linha, os valores médios por métrica de avaliação.

Tabela 3 – Correlação entre Habilidades (IRT) e outras Métricas de avaliação de modelos.

.	AUC-ROC	Precision	Recall	F1	AUC-PR	Acc	Log loss	Pseudo GT	EM	MV
DS12	0.29	0.29	0.26	0.29	0.29	0.17	-0.20	0.08	-0.02	0.22
DS4	0.54	0.59	0.50	0.59	0.54	0.59	-0.56	0.52	-0.07	0.20
DS14	0.41	0.49	0.39	0.49	0.41	0.49	-0.49	0.69	-0.22	0.08
DS3	0.38	0.38	0.35	0.38	0.38	0.38	-0.40	0.38	-0.72	0.51
DS51	0.40	0.45	0.40	0.44	0.52	0.59	-0.60	0.50	-0.58	0.48
musk	0.48	0.48	0.49	0.48	0.48	0.48	-0.46	0.48	-0.15	0.21
thyroid	0.63	0.67	0.62	0.67	0.65	0.67	-0.67	0.63	-0.37	0.44
vertebral	-0.30	-0.40	-0.40	-0.40	-0.40	-0.26	0.18	-0.30	-0.33	0.41
ionosphere	0.51	0.51	0.52	0.51	0.51	0.51	-0.52	0.51	0.04	0.05
letter	0.06	0.06	0.06	0.06	0.06	0.11	-0.09	0.06	-0.30	0.10
wine	0.35	0.35	0.34	0.35	0.35	0.35	-0.34	0.35	-0.28	0.28
yeast	0.54	0.57	0.51	0.55	0.55	0.57	-0.53	0.54	-0.23	0.17
cardio	0.57	0.63	0.57	0.59	0.63	0.63	-0.62	0.57	-0.17	0.23
vowels	0.37	0.34	0.34	0.34	0.34	0.37	-0.36	0.37	-0.26	0.27
breastw	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	-0.76	**0.78**	-0.22	0.43
glass	0.17	0.18	0.15	0.17	0.17	0.18	-0.16	0.17	-0.01	0.12
pima	0.31	0.31	0.26	0.26	0.31	0.36	-0.36	0.31	-0.36	0.48
annthyroid	0.49	0.55	0.49	0.53	0.53	0.55	-0.55	0.49	-0.08	0.24
speech	-0.60	-0.66	-0.66	-0.66	-0.66	-0.50	0.48	-0.60	-0.27	0.36
wbc	0.62	0.66	0.62	0.66	0.66	0.67	-0.63	0.62	-0.31	0.30
MÉDIA:	0.35	0.36	0.32	0.35	0.35	**0.38**	-0.38	0.35	-0.24	0.27

No geral, a métrica de Habilidade do IRT é bem correlacionada com as métricas de avali-

ação clássicas (AUC, *Precision*, entre outros). Outra observação é sobre o *dataset* ***breastw***, que apresentou os maiores graus de correlação, na maioria das Métricas de avaliação, entre a Habilidade derivada da concordância dos vários Métodos AD utilizados. Demonstrando, inclusive, uma alta correlação para a métrica de avaliação *Pseudo GT* (não supervisionada). Pode-se notar também, que o *dataset* ***wbc*** apresentou, na correlação de *Spearman*, valores acima de **0.6** em quase todas as métricas de avaliação adotadas, até mesmo no *Pseudo GT* (o qual utilizou a técnica *Unifying Outlier Scores*).

Outras informações observadas na correlação entre os diferentes *datasets* utilizados (artificiais ou reais), com os *scores* referentes às Medidas de Avaliação de Métodos AD, estão descritas na Tabela 4 . Onde a média aritmética de todos os *scores* de métrica AUC, obtidos pelos Métodos AD adotados, obteve a pontuação máxima de **0.84** no conjunto de dados *DS4*. Seguido pelo *dataset* artificial *DS14*, que por sua vez foi a maior pontuação da métrica *F1-Score* (*F-Measure*). Quanto aos menores valores de AUC *score*, foram identificados nos conjuntos de dados reais *speech* e *vertebral*. Inclusive este último, alcançou os menores *scores* na maioria das métricas de avaliação restantes, como *F1-Score*, *Recall*, entre outros. Demonstrando assim a dificuldade que este conjunto de dados apresentou, para que os Algoritmos de AD identificassem corretamente anomalias.

Tabela 4 – Principais conjuntos de dados (artificiais e reais), que apresentaram em média mais *scores* (métricas AUC, *F-Measure*) na detecção de anomalias, bem como também aqueles com menor *score*.

.	AUC-ROC	Precision	Recall	F1	AUC-PR	Acc	Log loss	Pseudo GT	EM	MV
DS4	**0.84**	0.06	**0.77**	0.12	0.42	0.90	3.26	**0.87**	5.2e-03	1.51
DS14	0.83	**0.38**	0.72	**0.49**	**0.56**	**0.95**	2.67	0.86	2.6e-03	3.07
speech	0.46	5.4e-03	0.03	9.2e-03	0.02	0.88	4.06	0.46	1.6e-05	5.4e02
vertebral	0.45	0.02	0.01	0.01	0.07	0.78	7.92	0.45	4.3e-12	1.9e09

Outro exemplo de *insights* obtidos neste experimento, são os conjuntos de dados que obtiveram os maiores valores de traço latente Dificuldade, a partir do modelo IRT ajustado. Como podemos observar na na Tabela 5 , o *dataset* ***Ionosphere*** foi o que apresentou a maior dificuldade média, entre suas amostras (*outliers* ou não), dentre os conjuntos de dados reais ou artificiais. Quanto ao *dataset* ***DS12***, por sua vez, apresentou o menor valor médio do traço latente Dificuldade, dentre todos os conjuntos de dados ajustados em Algoritmos de AD adotados neste experimento.

Outro fato importante para se destacar, são os Métodos AD que alcançaram os maiores

Tabela 5 – Principais conjuntos de dados (artificiais e reais), que apresentaram em média mais Dificuldade (traço latente IRT), e também aqueles com menos Dificuldade.

Origem dos dados	Conjunto de dados	Dificuldade IRT média
ODDS (*real-world*)	ionosphere	**0.0130**
ODDS (*real-world*)	wine	0.0128
PyOD generator (artificial/sintético)	DS51	**0.0092**
ODDS (*real-world*)	yeast	0.0070
PyOD generator (artificial/sintético)	DS14	0.0055
PyOD generator (artificial/sintético)	DS12	0.0052

Fonte: Elaborada pelo autor (2024).

valores médios do traço latente Habilidade (IRT), durante ajuste entre os diferentes conjuntos de dados selecionados. Na Tabela 6 pode-se observar que os algoritmos OCSVM, LOF, e *AutoEncoder* possuem quase o mesmo valor de Habilidade, o maior dentre os 23 algoritmos utilizados neste experimento. Já em relação ao menor valor médio de Habilidade IRT, o algoritmo *Deep SVDD* atingiu o pior resultado.

Tabela 6 – Principais algoritmos de AD, que apresentaram em média mais Habilidade (traço latente IRT), e também aqueles com menos Habilidade.

Algoritmos de AD	Habilidade IRT média
One-Class Support Vector Machines	**0.9995**
Local Outlier Factor	0.9994
Fully Connected AutoEncoder	0.9994
Single-Objective Generative Adversarial Active Learning	0.9531
Deep Support Vector Data Description	0.6319

Fonte: Elaborada pelo autor (2024).

5.2 HABILIDADES ESTIMADAS

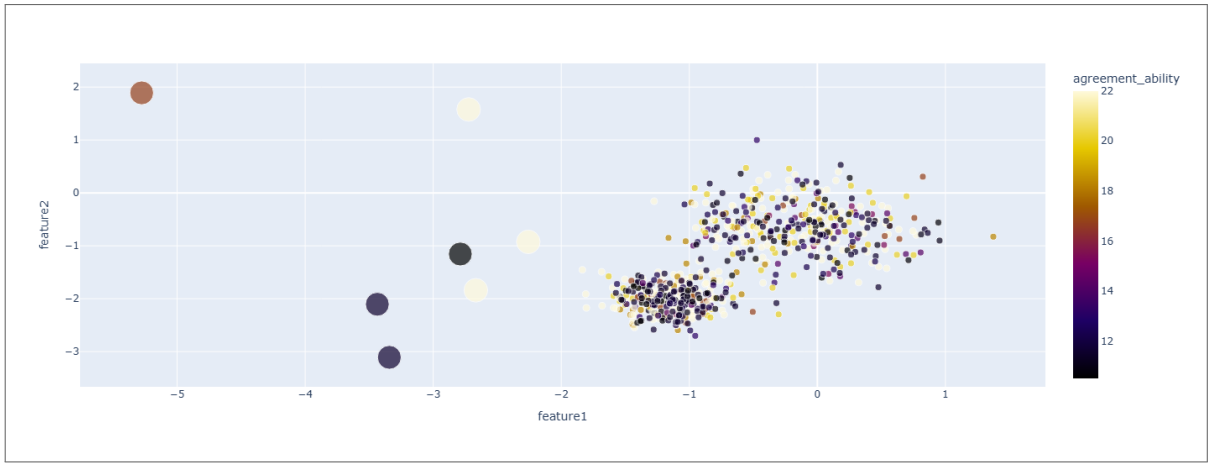
Outro dos resultados observados nos experimentos, foi após a execução do Módulo IRT do ODAIRE, quando foram obtidos os traços latentes de Habilidade (derivados da concordância dos respondentes, neste caso os Métodos AD) para cada uma das instâncias de um *dataset*, conforme já foi descrito anteriormente na seção 4.5.4 .

Com os dados originais do conjunto de dados *DS4* e o vetor de Habilidades (IRT) obtido, após calculada a média ponderada (descrito na Função 4.3) das concordâncias entre Modelos AD, foi possível gerar o gráfico de dispersão na Figura 3, onde é possível observar que a medida que aumenta a dificuldade de uma instância (ou seja, o quão difícil foi para os Métodos AD concordarem se é uma *anomalia* ou não) sua cor fica mais amarelada (ou laranja).

E pode-se notar que as instâncias em algumas regiões das áreas mais densas, apresentam menos concordâncias. Foram gerados também, os gráficos para os conjuntos de dados artificiais *DS3* (Figura 4), *DS12* (Figura 5), *DS14* (Figura 6) e *DS51* (Figura 7).

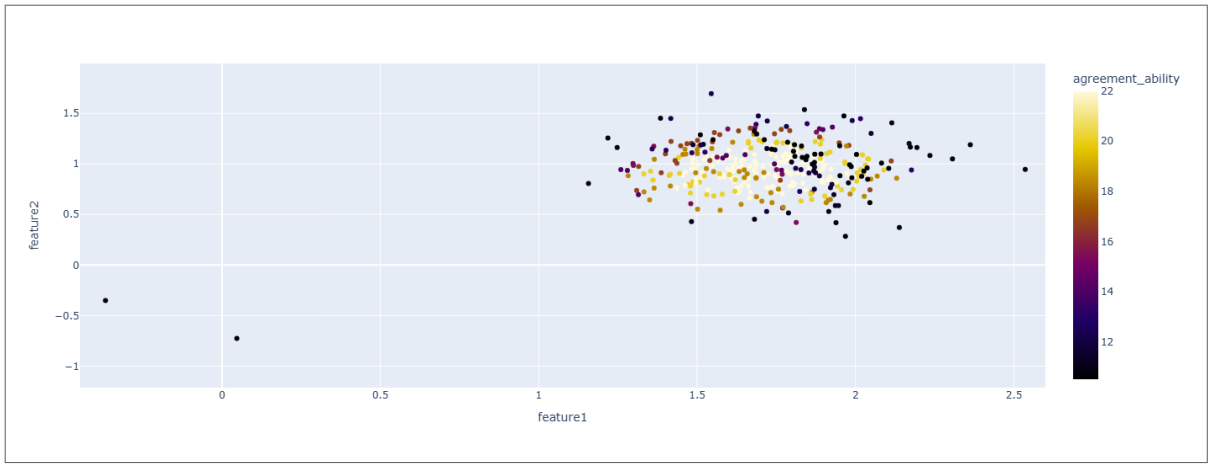
Outros gráficos de dispersão e valores de traço latente Habilidade IRT (inclusive com média ponderada das predições), referentes aos outros conjuntos de dados reais públicos adotados, podem ser encontrados no Apêndice C, com maiores detalhes.

Figura 3 – Gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS4*. Destacando no tamanho dos pontos, as classes positivas.



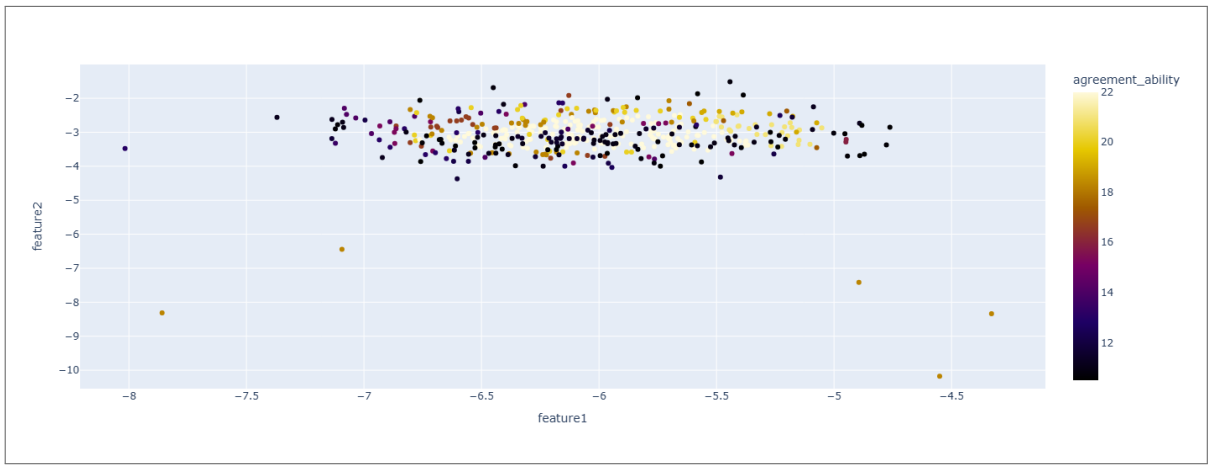
Fonte: Autoria própria.

Figura 4 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS3*



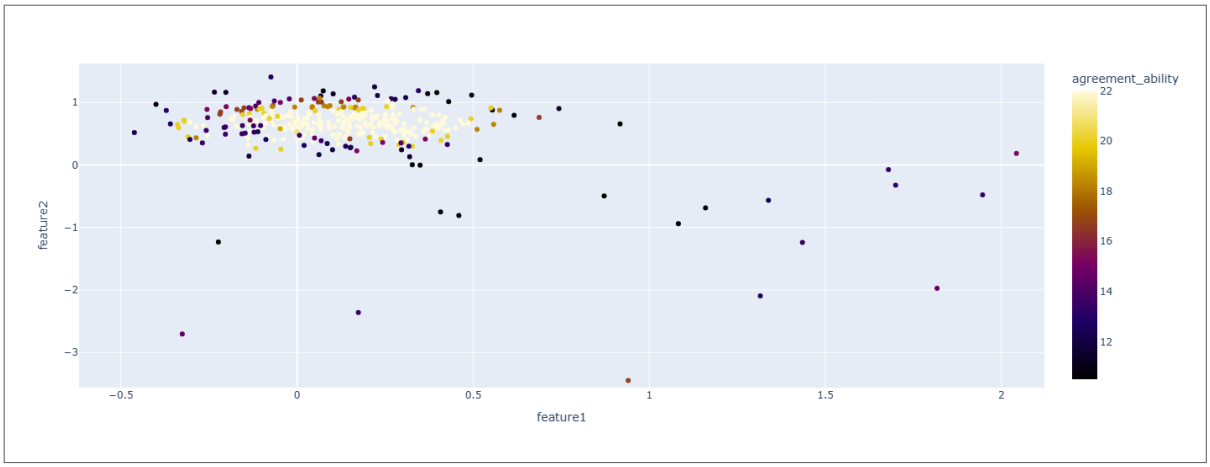
Fonte: Autoria própria.

Figura 5 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS12*



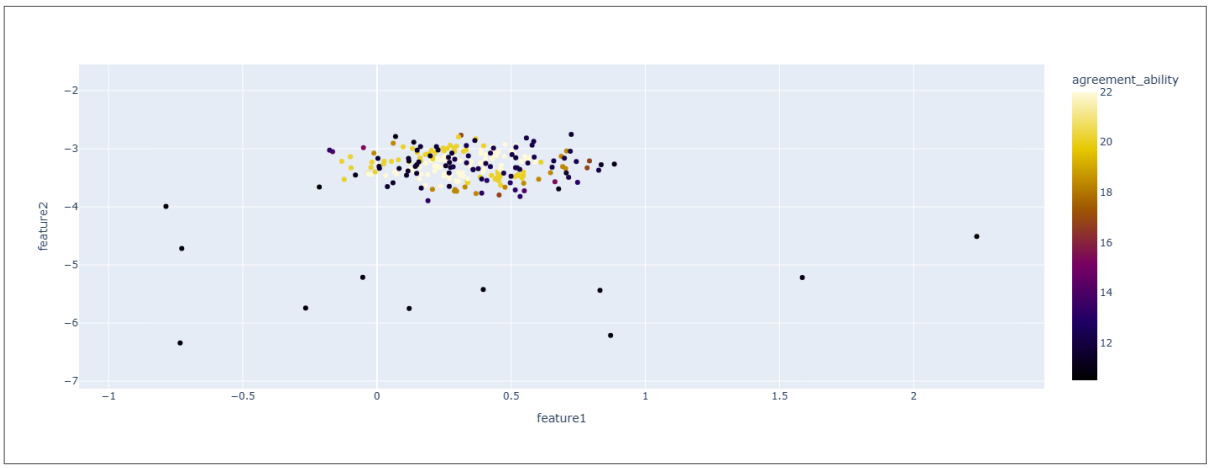
Fonte: Autoria própria.

Figura 6 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS14*



Fonte: Autoria própria.

Figura 7 – Trecho de gráfico de dispersão 2D representando média ponderada de concordâncias dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS51*



Fonte: Autoria própria.

5.3 DIFICULDADES ESTIMADAS

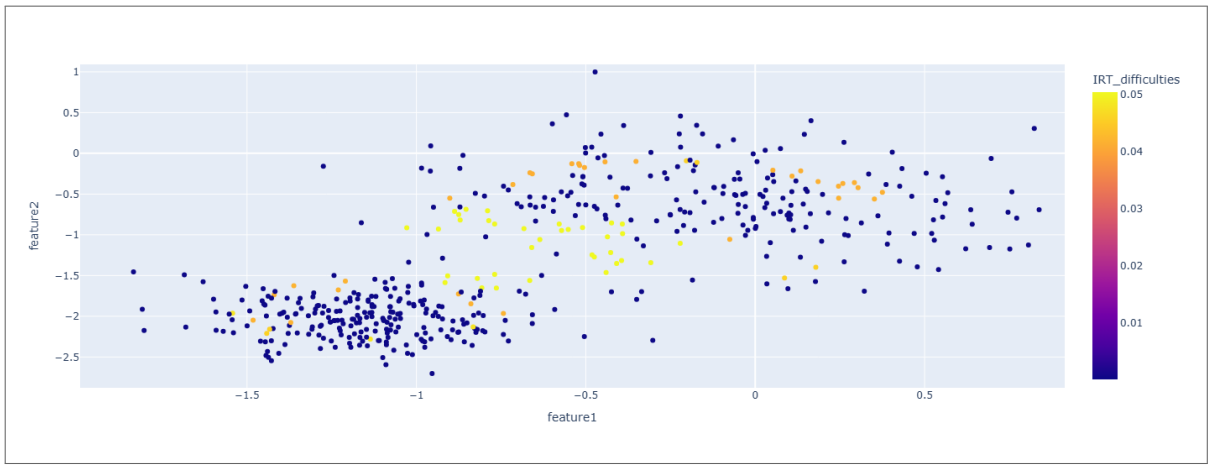
Mais um dos resultados observados nestes experimentos, foi após a execução do Módulo IRT do ODAIRE, quando foram obtidos os traços latentes de Dificuldade (derivados da concordância dos respondentes, neste caso os Métodos AD) para cada uma das instâncias de um *dataset*, conforme foi descrito na seção 4.5 anteriormente.

Com os dados originais do conjunto de dados *DS4* e o vetor de Dificuldades (IRT) obtido, foi possível gerar o gráfico de dispersão na Figura 8, onde é possível observar que a medida que aumenta a dificuldade de uma instância (ou seja, o quão difícil foi para os Métodos AD concordarem se é uma *anomalia* ou não) sua cor fica mais amarelada (ou laranja).

E pode-se notar que as instâncias em região de fronteira dos *clusters*, apresentam as maiores divergências. Também na borda do *cluster* mais disperso, foi identificado um grau médio de dificuldade também. Foram gerados igualmente, os gráficos para os conjuntos de dados artificiais *DS3* (Figura 9), *DS12* (Figura 10), *DS14* (Figura 11) e *DS51* (Figura 12).

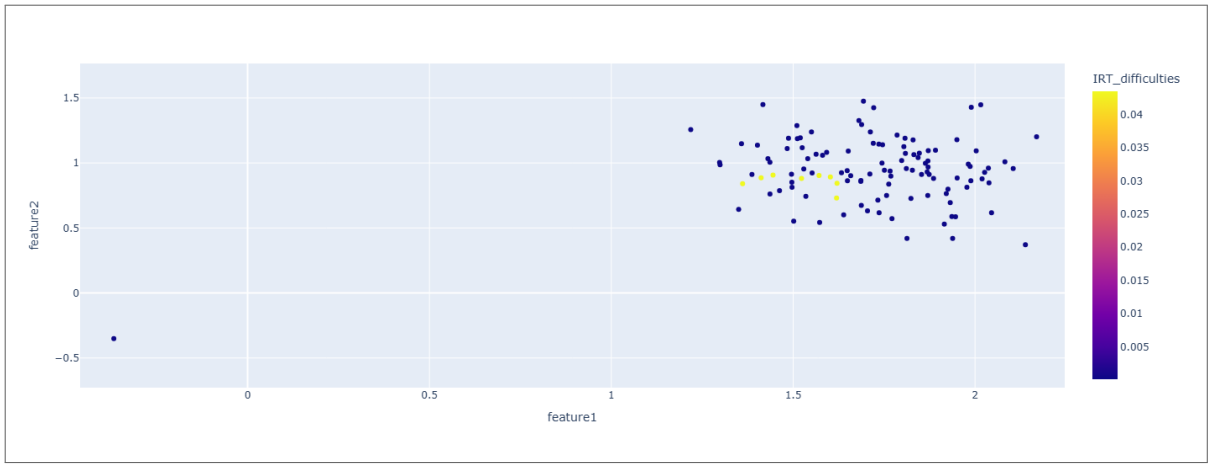
Os outros gráficos de dispersão e valores de traço latente Dificuldade (IRT), referentes aos outros *datasets* além do *DS4* (inclusive os conjuntos de dados reais públicos adotados), podem ser encontrados no Apêndice C, com maiores detalhes.

Figura 8 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS4* (sub-conjuntos de treino e teste).



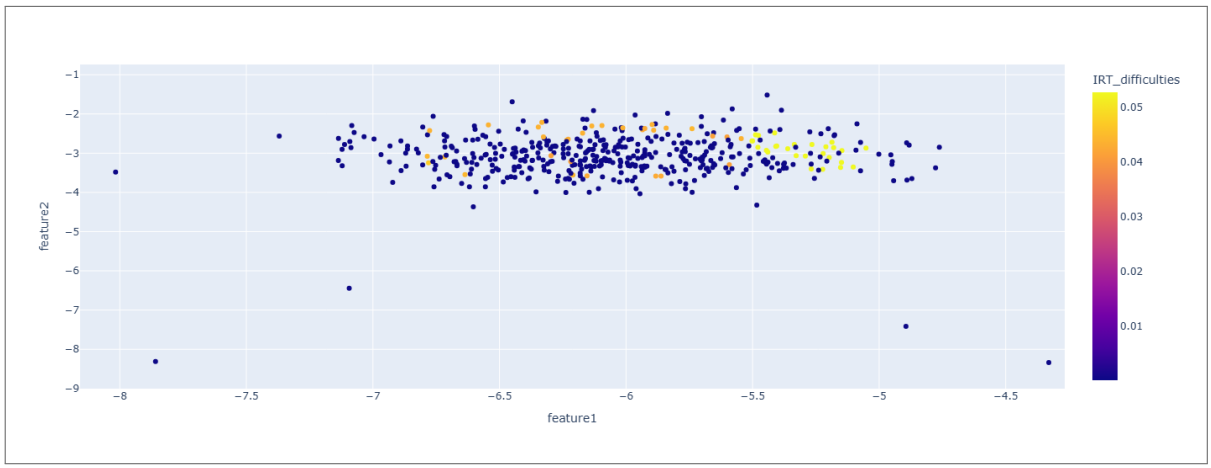
Fonte: Autoria própria.

Figura 9 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS3* (sub-conjuntos de treino e teste).



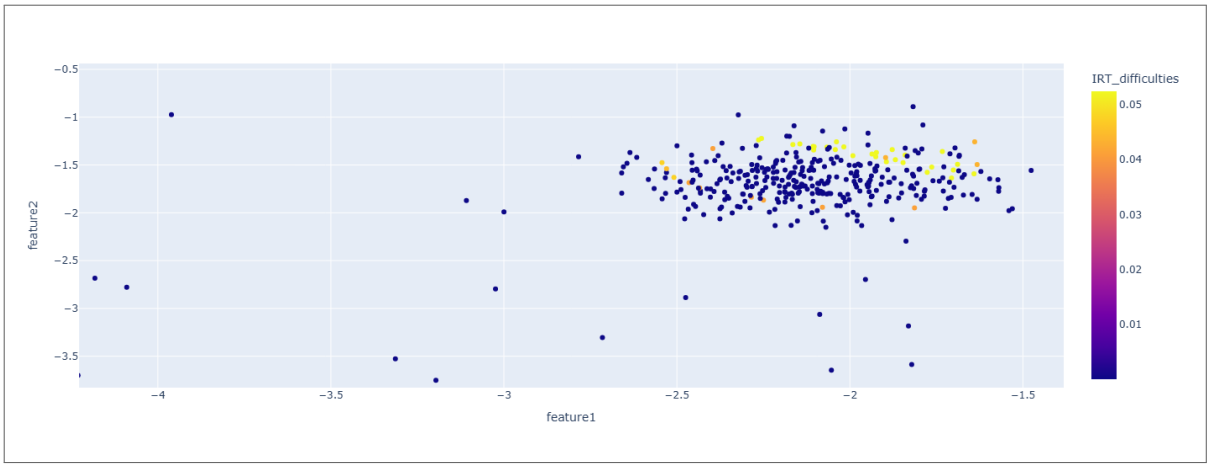
Fonte: Autoria própria.

Figura 10 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS12* (sub-conjuntos de treino e teste).



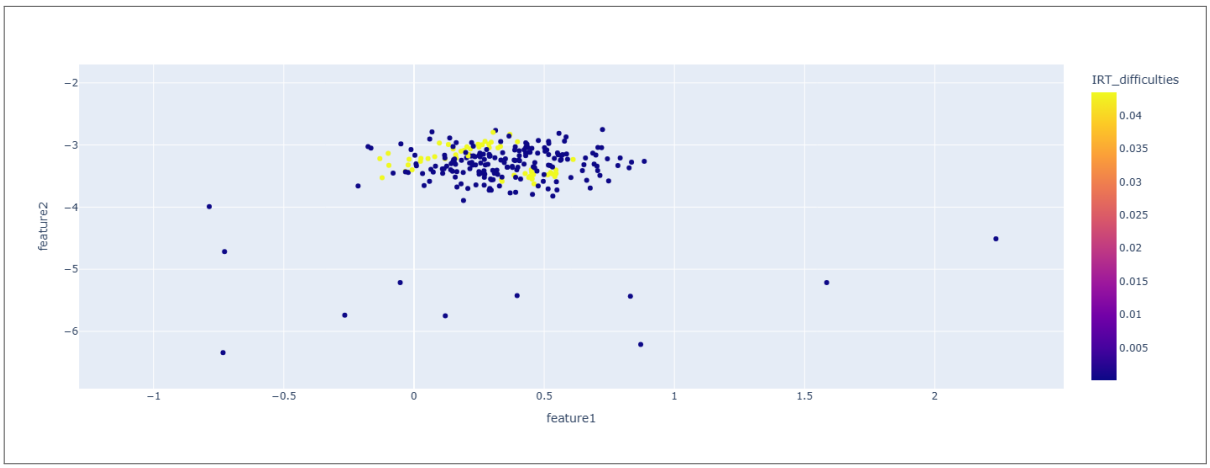
Fonte: Autoria própria.

Figura 11 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS14* (sub-conjuntos de treino e teste).



Fonte: Autoria própria.

Figura 12 – Trecho do gráfico de dispersão em espaço 2D representando traço latente *Dificuldade* em cada instância do *DS51* (sub-conjuntos de treino e teste).



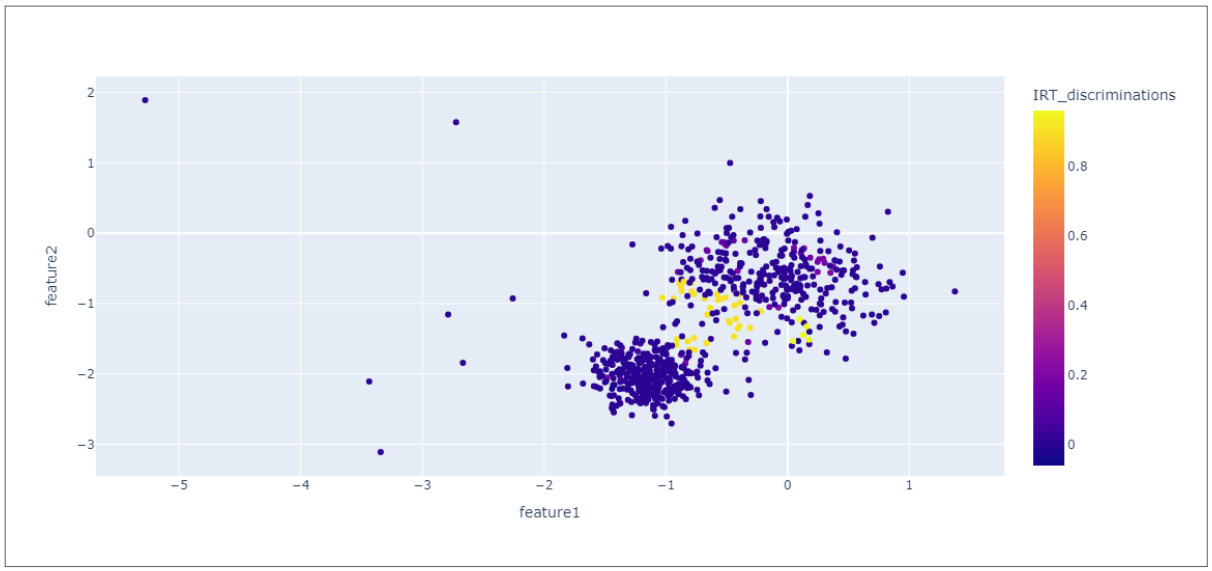
Fonte: Autoria própria.

5.4 DISCRIMINAÇÕES ESTIMADAS

Mais um dos resultados observados nestes experimentos, foi após a execução do Módulo IRT do ODAIRE, quando foram obtidos os traços latentes de Discriminação (derivados da concordância dos respondentes, neste caso os Métodos AD) para cada uma das instâncias de um *dataset*, conforme foi descrito na seção 4.5 anteriormente.

Com os dados originais do conjunto de dados *DS4* e o vetor de Discriminação (IRT) obtido, foi possível gerar o gráfico de dispersão na Figura 13, onde é possível observar que a medida que aumenta a discriminação de uma instância (ou seja, o quão restrito foram os únicos Métodos AD que concordaram entre si, se era uma *anomalia* ou não) sua cor fica mais amarelada (ou laranja). E pode-se notar que as instâncias principalmente em região de fronteira dos *clusters*, apresentam as maiores divergências.

Figura 13 – Gráfico de dispersão em espaço 2D representando traço latente *Discriminação* em cada instância do *DS4* (sub-conjuntos de treino e teste).



Fonte: Autoria própria.

Os outros gráficos de dispersão e valores de traço latente Discriminação (IRT), referentes aos outros *datasets* além do *DS4* (inclusive os conjuntos de dados reais públicos adotados), podem ser encontrados no Apêndice C, com maiores detalhes.

5.5 PONTOS NEGATIVOS

Ao longo deste trabalho, foi possível identificar de forma prática, algumas limitações e desafios potenciais do método proposto. Como por exemplo, a dependência de múltiplos modelos. Pois o ODAIRE depende de um conjunto diverso e numeroso de modelos de detecção de anomalias (*pool* de algoritmos), para estimar concordância. Isso implica que, o método não é aplicável se houver poucos modelos disponíveis, ou se os modelos forem muito semelhantes entre si (pouca diversidade pode gerar concordância espúria). Somado a isto, o desempenho depende da qualidade média dos modelos no *pool*, pois se muitos forem fracos, a estimativa de concordância pode ser enviesada.

Outro aspecto que vale destacar, é a suposição forte de consenso como verdade, em que o método proposto assume que melhores Modelos de Detecção de anomalias, sempre concordam sobre quais instâncias são anômalas. Porém esta suposição pode não sustentar-se em dados muito complexos ou ambíguos, onde mesmo bons modelos podem discordar. Isso pode comprometer a precisão da habilidade estimada.

Há também a questão da aplicabilidade limitada em cenários com pouca variabilidade. Como o ODAIRE se apoia na variabilidade de respostas entre modelos, em casos onde todos os modelos concordam demais (por exemplo, em conjuntos triviais), ou discordam em excesso (em dados ruidosos ou mal estruturados), o método proposto pode não gerar discriminação significativa entre instâncias ou modelos, dificultando a avaliação dos *outliers*.

Também de forma empírica, durante os experimentos realizados com datasets sintéticos, observou-se que o ODAIRE não performava bem em dados pontuais, contendo mais de dois *clusters* não próximos entre si. Em que a identificação de *outliers* apresentava mais classificações incorretas de modo geral.

Por fim, um aspecto negativo da solução proposta, é o custo computacional, complexidade algorítmica e tempo. Onde a solução requer execução de vários modelos sobre todos os dados, construção de matrizes de concordância, e estimativas com o modelo contínuo de IRT (β4-IRT). O que pode representar alto custo computacional para grandes volumes de dados e muitos modelos de detecção.

5.5.1 Vieses nos dados

Na literatura há vários estudos (MAVROGIORGOS et al., 2024) sobre a sensibilidade de modelos de ML, assim como modelos de AD, à vieses nos dados usados para treinamento. Contudo, neste presente trabalho podemos destacar alguns tipos, para maior aprofundamento de análise dos possíveis impactos nos resultados, e como subsídio para possíveis trabalhos futuros.

Um desses tipos é o Viés de Distribuição, que ocorre quando a distribuição dos dados de treinamento não corresponde à distribuição dos dados reais ou dos dados de teste. O problema é que tal modelo aprende padrões que não se aplicam corretamente a novos dados. (PELIKAN; GOLDBERG; LOBO, 2002) Por exemplo, um sistema de detecção de anomalias, treinado principalmente com amostras de um determinado tipo de *outlier*, pode ter baixa precisão ao analisar amostras com outros tipos de *outliers*.

Outro tipo é o Viés de Densidade, que surge quando há diferenças na densidade de amostragem entre diferentes partes dos dados. Seu problema é que algumas classes ou categorias têm mais exemplos do que outras, influenciando desproporcionalmente o aprendizado do modelo (ALBERTINI, 2012). Um exemplo seria um modelo de detecção de fraudes bancárias, que pode ser treinado com poucos exemplos de fraudes reais, fazendo com que tenha dificuldade em identificá-las.

E por fim, outro importante tipo é o Viés de Dimensionalidade, que acontece quando há muitas variáveis (*features*) nos dados em relação à quantidade de amostras, tornando o aprendizado do modelo mais difícil. Seu problema é que o modelo pode sofrer com a 'maldição da dimensionalidade', onde muitas variáveis irrelevantes tornam difícil encontrar padrões significativos. (POGGIO et al., 2017) Por exemplo, um modelo de previsão de preços de imóveis pode ter centenas de atributos irrelevantes, dificultando a identificação dos fatores mais importantes.

5.5.2 Vieses nos algoritmos

Há também na literatura atualmente, várias publicações (AGGARWAL, 2017) sobre a sensibilidade nos modelos de ML, bem como Métodos de AD, aos vieses em seus algoritmos. Entretanto, neste trabalho de dissertação vamos destacar alguns desses tipos, que podem comprometer a detecção de anomalias ao afetar a capacidade do modelo de capturar padrões relevantes e distinguir corretamente comportamentos normais e anômalos. Este assunto, têm potencial de maior investigação e análise dos possíveis implicações nos resultados dos experi-

mentos. Além de sua mitigação ser tema de possíveis trabalhos futuros.

Um dos tipos é o Viés de Família de Algoritmos, que é quando um algoritmo ou uma classe específica de algoritmos, impõe restrições que limitam sua capacidade de identificar corretamente anomalias. Este viés está relacionado às premissas embutidas na abordagem do algoritmo (AGGARWAL, 2017). Por exemplo, modelos Baseados em Distribuições Estatísticas (*Gaussian Mixture Models*) assumem que os dados seguem uma distribuição estatística específica (normal), podendo falhar se a distribuição real for muito diferente.

Há também o Viés de Hiperparâmetros, é um tipo que surge quando as configurações dos hiperparâmetros do modelo favorecem certos padrões ou distribuições de anomalias, reduzindo sua capacidade de generalização. E pode ocorrer devido à escolha inadequada dos valores dos hiperparâmetros, impactando a sensibilidade do modelo (AGGARWAL, 2017). Por exemplo no algoritmo *K-Means*, se o k for muito pequeno ou muito grande, o modelo pode classificar erroneamente instâncias normais como anômalas ou ignorar anomalias reais.

Por fim, um outro tipo de viés é o Erro Sistemático, que ocorre quando há um viés estrutural ou uma tendência constante nos dados que afeta a detecção de anomalias. Diferente do erro aleatório (que é imprevisível e se distribui de forma variável), o erro sistemático leva a desvios consistentes nas previsões (HAWKINS, 1980). Por exemplo, erros introduzidos por transformações de dados, onde se um modelo é treinado em dados padronizados de forma inadequada, pode não reconhecer corretamente padrões anômalos em dados reais.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste dissertação, propusemos uma nova abordagem para validação de AD, chamada ODAIRE, que se baseia na concordância do modelo e na IRT para estimar a capacidade do modelo AD e a dificuldade da instância. Os experimentos classificaram um conjunto de métodos AD em conjuntos de dados com características diferentes. As classificações dos modelos por capacidade foram sólidas, considerando aspectos como suposições do modelo, o número certo de valores discrepantes e características do conjunto de dados. Somado a isto, a capacidade foi correlacionada com diferentes medidas de validação supervisionada, embora nenhuma informação supervisionada tenha sido fornecida ao ODAIRE. Também, a classificação do modelo ODAIRE foi eficaz mesmo quando havia um grande número de modelos aleatórios no conjunto.

ODAIRE aproveita um conjunto de modelos para chegar a um consenso sobre a distribuição de *outliers*. Inclusive, a IRT dá mais peso às instâncias que são mais difíceis de rotular como *outliers*, o que aumenta a discriminação do desempenho do modelo. A dificuldade da instância também pode ser utilizada para uma análise mais profunda dos resultados, identificando, por exemplo, áreas de incerteza como fronteiras de densidade de pontos.

6.1 TRABALHOS FUTUROS

Como possíveis trabalhos futuros, para expandir o escopo da solução proposta nesta dissertação, temos como direções de pesquisa:

- **Características dos dados**: Como já discutido anteriormente, a investigação dos conceitos de Esparsidade, Dimensionalidade e Ruído nos *datasets* utilizados, e seus respectivos efeitos nos resultados dos experimentos.
- **Características das anomalias**: Como já discutido em capítulos anteriores, a investigação dos conceitos como anomalia pontual, global, sintética, etc, nos *datasets* utilizados. Bem como seus respectivos efeitos nos resultados dos experimentos.
- **Famílias de algoritmos AD**: Como já discutido anteriormente, a investigação dos tipos de famílias de algoritmos, como os baseados em Distância, Densidade, Reconstrução,

entre outros, nos *datasets* utilizados. Bem como seus respectivos efeitos nos resultados dos experimentos.

- **Vieses nos dados**: Como já discutido em capítulos anteriores, a investigação dos conceitos de Viés de Distribuição, Densidade, Dimensionalidade, etc, nos *datasets* utilizados. Bem como seus respectivos efeitos nos resultados dos experimentos.
- **Vieses nos algoritmos**: Como já discutido anteriormente, a investigação dos tipos de Viés de Família de Algoritmos, Viés de Hiperparâmetros, Erro Sistemático, nos *datasets* utilizados. Bem como seus respectivos efeitos nos resultados dos experimentos.
- **Aplicação da abordagem ODAIRE em outras tarefas de AI**: Expandir o uso da combinação entre IRT e modelos de concordância para domínios além de detecção de anomalias, como classificação, *clustering* ou seleção de características.
- **Suposição de que bons modelos sempre concordam**: Investigar métricas alternativas de consenso ponderado, que considerem cenários onde o desacordo entre bons modelos é natural (ex: anomalias sutis ou ambíguas).
- **Baixa discriminação em dados triviais ou com excesso de ruído**: Incorporar técnicas de normalização ou filtragem de ruído nos dados de entrada e desenvolver versões robustas da matriz de concordância.
- **Custo computacional elevado**: Propor versões mais eficientes computacionalmente, com redução da dimensionalidade da matriz de resposta ou uso de IRT com estimativa incremental.
- **Necessidade de diversidade no pool de modelos**: Explorar estratégias de geração automática de diversidade de modelos (ex: *ensembles* com variação de hiperparâmetros, métodos evolutivos).
- **Limitações na visualização e interpretação dos traços latentes**: Criar ferramentas de visualização interativa dos traços IRT, para auxiliar usuários a entenderem o desempenho por instância e modelo.

Já em relação aos possíveis trabalhos futuros, com o intuito de mitigar limitações do ODAIRE, também temos as seguintes direções de pesquisa:

- **Poucos modelos de detecção de anomalias disponíveis**: Uma limitação do ODAIRE é o desempenho prejudicado quando há poucos algoritmos no **pool** de entrada. Uma evolução futura, pode ser desenvolver variantes com 'amostragem inteligente' ou 'seleção ativa' de modelos.

- **Assume que bons modelos sempre concordam**: Tal suposição pode falhar em casos ambíguos. Então buscar investigar 'consenso ponderado', que reconheça desacordos legítimos entre bons modelos.

- **Desempenho fraco em dados triviais ou ruidosos**: Baixa capacidade de discriminação nestes cenários. Então buscar avançar na inclusão de 'filtros de ruído' e técnicas de 'normalização' para aumentar robustez.

- **Custo computacional elevado**: Na solução proposta, atualmente exige múltiplas execuções e ajustes complexos. Entretanto, futuramente pode ser investigado um 'IRT incremental' e 'redução de dimensionalidade' na matriz de concordância.

- **Necessidade de diversidade entre modelos**: Atualmente a avaliação dependente da variedade de algoritmos. Contudo, no futuro, pode ser investigado a criação de diversidade com 'ensembles automatizados' ou variação sistemática de hiperparâmetros.

- **Interpretação dos traços latentes**: No momento é um pouco difícil visualização dos parâmetros (habilidade, dificuldade). Portanto, um avanço seria desenvolver 'ferramentas visuais interativas' com foco exploratório e interpretativo.

REFERÊNCIAS

AEBERHARD, S.; FORINA, M. *Wine*. 1991. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J.

AGGARWAL, C. C. *Outlier Analysis*. 2nd. ed. Cham: Springer International Publishing, 2017. ISBN 978-3-319-47578-3.

ALBERTINI, M. K. Adaptação de viés indutivo de algoritmos de agrupamento de fluxos de dados. 2012.

AN, J.; CHO, S. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, v. 2, n. 1, p. 1–18, 2015.

BIRNBAUM, A. Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*, Addison-Wesley, 1968.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.

BOCK, R. D. Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, Springer, v. 37, n. 1, p. 29–51, 1972.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 2000. p. 93–104.

BURNELL, R.; SCHELLAERT, W.; BURDEN, J.; ULLMAN, T. D.; MARTINEZ-PLUMED, F.; TENENBAUM, J. B.; RUTAR, D.; CHEKE, L. G.; SOHL-DICKSTEIN, J.; MITCHELL, M.; KIELA, D.; SHANAHAN, M.; VOORHEES, E. M.; COHN, A. G.; LEIBO, J. Z.; HERNANDEZ-ORALLO, J. Rethink reporting of evaluation results in ai. *Science*, v. 380, n. 6641, p. 136–138, 2023. Disponível em: <https://www.science.org/doi/abs/10.1126/science.adf6369>.

CARDOSO, L. F. F.; RIBEIRO, J. de S.; SANTOS, V. C. A.; SILVA, R. L.; MOTA, M. P.; PRUDêNCIO, R. B. C.; ALVES, R. C. O. *Explanation-by-Example Based on Item Response Theory*. 2022. Disponível em: <https://arxiv.org/abs/2210.01638>.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 41, n. 3, jul. 2009. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/1541880.1541882>.

CHEN, Y.; FILHO, T. S.; PRUDENCIO, R. B.; DIETHE, T.; FLACH, P. B3-irt: A new item response model and its applications. In: PMLR. *The 22nd International Conference on Artificial Intelligence and Statistics*. [S.l.], 2019. p. 1013–1021.

COOK, R. D. Detection of influential observation in linear regression. *Technometrics*, [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], v. 19, n. 1, p. 15–18, 1977. ISSN 00401706. Disponível em: <http://www.jstor.org/stable/1268249>.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2006. (ICML '06), p. 233–240. ISBN 1595933832. Disponível em: <https://doi.org/10.1145/1143844.1143874>.

EMBRETSON, S. E.; REISE, S. P. *Item response theory*. [S.l.]: Psychology Press, 2013.

FABRA-BOLUDA, R.; FERRI, C.; MARTÍNEZ-PLUMED, F.; RAMÍREZ-QUINTANA, M. J. Robustness testing of machine learning families using instance-level irt-difficulty. In: *EBeM@ IJCAI*. [S.l.: s.n.], 2022.

FERREIRA-JUNIOR, M.; REINALDO, J. T.; NETO, E. A. L.; PRUDENCIO, R. B. et al. B4-irt: A new b3-irt with enhanced discrimination estimation. *arXiv preprint arXiv:2303.17731*, 2023.

FILHO, J. de S. R.; CARDOSO, L. F. F.; SILVA, R. L. S. da; CARNEIRO, N. J. S.; SANTOS, V. C. A.; ALVES, R. C. de O. Explanations based on item response theory (exirt): A model-specific method to explain tree-ensemble model in trust perspective. *Expert Systems with Applications*, Elsevier BV, v. 244, p. 122986, jun. 2024. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2023.122986>.

GOIX, N. How to evaluate the quality of unsupervised anomaly detection algorithms? *arXiv preprint arXiv:1607.01152*, 2016.

GOLDSTEIN, M.; UCHIDA, S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE*, Public Library of Science, v. 11, n. 4, p. 1–31, 04 2016. Disponível em: <https://doi.org/10.1371/journal.pone.0152173>.

GOMES, J. M. D. *Use of item response theory as a metric for evaluation of decoders for human-machine interfaces*. 2022. Trabalho de Conclusão de Curso (Graduação em Engenharia Biomédica) – Departamento de Engenharia Biomédica, Universidade Federal do Rio Grande do Norte. Disponível em: <https://repositorio.ufrn.br/handle/123456789/51195>.

HAMBLETON, R. K. *Fundamentals of item response theory*. [S.l.]: Sage, 1991.

HAN, S.; HU, X.; HUANG, H.; JIANG, M.; ZHAO, Y. Adbench: Anomaly detection benchmark. In: *Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2022.

HAWKINS, D. M. *Identification of outliers*. [S.l.]: Springer, 1980. v. 11.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. *Pattern recognition letters*, Elsevier, v. 24, n. 9-10, p. 1641–1650, 2003.

IVANOVSKA, M.; PERŠ, J.; TABERNIK, D.; SKOČAJ, D. Evaluation of anomaly detection algorithms for the real-world applications. In: IEEE. *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.], 2021. p. 6196–6203.

KANDANAARACHCHI, S. Unsupervised anomaly detection ensembles using item response theory. *Information Sciences*, Elsevier, v. 587, p. 142–163, 2022.

KANDANAARACHCHI, S.; SMITH-MILES, K. Comprehensive algorithm portfolio evaluation using item response theory. *Journal of Machine Learning Research*, v. 24, n. 177, p. 1–52, 2023.

KUO, C. *Handbook of Anomaly Detection: With Python Outlier Detection*. New York: Independently published, 2023. ISBN 979-8372339767.

KWITT, R.; HOFMANN, U. Unsupervised anomaly detection in network traffic by means of robust pca. In: IEEE. *2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*. [S.l.], 2007. p. 37–37.

LALOR, J. P.; RODRIGUEZ, P.; SEDOC, J.; HERNANDEZ-ORALLO, J. Item response theory for natural language processing. In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. [S.l.: s.n.], 2024. p. 9–13.

LALOR, J. P.; WU, H.; YU, H. Learning latent parameters without human response patterns: Item response theory with artificial crowds. In: NIH PUBLIC ACCESS. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. [S.l.], 2019. v. 2019, p. 4240.

LI, Z.; ZHAO, Y.; BOTTA, N.; IONESCU, C.; HU, X. Copod: Copula-based outlier detection. In: *2020 IEEE International Conference on Data Mining (ICDM)*. [S.l.: s.n.], 2020. p. 1118–1123.

LINDEN, W. J. Van der. *Handbook of item response theory: Three volume set*. [S.l.]: CRC Press, 2018.

LIU, Y.; LI, Z.; ZHOU, C.; JIANG, Y.; SUN, J.; WANG, M.; HE, X. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, v. 32, n. 8, p. 1517–1528, Aug 2020. ISSN 1558-2191.

LORD, F. M.; NOVICK, M. R. *Statistical theories of mental test scores*. [S.l.]: IAP, 2008.

MAO, A.; MOHRI, M.; ZHONG, Y. Cross-entropy loss functions: theoretical analysis and applications. In: *Proceedings of the 40th International Conference on Machine Learning*. [S.l.]: JMLR.org, 2023. (ICML'23).

MARTINEZ-PLUMED, F.; HERNÁNDEZ-ORALLO, J. Ai results for the atari 2600 games: difficulty and discrimination using irt. *EGPAI, Evaluating General-Purpose Artificial Intelligence*, v. 33, 2016.

MARTÍNEZ-PLUMED, F.; PRUDÊNCIO, R. B.; MARTÍNEZ-USÓ, A.; HERNÁNDEZ-ORALLO, J. Making sense of item response theory in machine learning. In: *ECAI 2016*. [S.l.]: IOS Press, 2016. p. 1140–1148.

MARTÍNEZ-PLUMED, F.; PRUDÊNCIO, R. B.; MARTÍNEZ-USÓ, A.; HERNÁNDEZ-ORALLO, J. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, Elsevier, v. 271, p. 18–42, 2019.

MARTíNEZ-PLUMED, F.; CASTELLANO, D.; MONSERRAT-ARANDA, C.; HERNáNDEZ-ORALLO, J. When ai difficulty is easy: The explanatory power of predicting irt difficulty. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 36, n. 7, p. 7719–7727, Jun. 2022. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/20739>.

MASTERS, G. N. A rasch model for partial credit scoring. *Psychometrika*, Springer, v. 47, n. 2, p. 149–174, 1982.

MAVROGIORGOS, K.; KIOURTIS, A.; MAVROGIORGOU, A.; MENYCHTAS, A.; KYRIAZIS, D. Bias in machine learning: A literature review. *Applied Sciences*, v. 14, n. 19, 2024. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/14/19/8860>.

MORAES, J. V. C. *-IRT: An Item Response Theory Model for Evaluating Regression Algorithms*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, Brazil, mar. 2021. Dissertação de Mestrado em Ciência da Computação. Disponível em: <https://repositorio.ufpe.br/handle/123456789/50976>.

NOEL, Y.; DAUVIER, B. A beta item response model for continuous bounded responses. *Applied Psychological Measurement*, v. 31, n. 1, p. 47–73, 2007. Disponível em: <https://doi.org/10.1177/0146621605287691>.

OLIVEIRA, C. S.; MORAES, J. V.; FILHO, T. S.; PRUDÊNCIO, R. B. A two-level item response theory model to evaluate speech synthesis and recognition. *Speech Communication*, Elsevier, v. 137, p. 19–34, 2022.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY Édouard. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <http://jmlr.org/papers/v12/pedregosa11a.html>.

PELIKAN, M.; GOLDBERG, D. E.; LOBO, F. G. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, Springer, v. 21, p. 5–20, 2002.

POGGIO, T.; MHASKAR, H.; ROSASCO, L.; MIRANDA, B.; LIAO, Q. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, Springer, v. 14, n. 5, p. 503–519, 2017.

POWERS, D. M. W. *What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes*. 2019. Disponível em: <https://arxiv.org/abs/1503.06410>.

POWERS, D. M. W. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 2020. Disponível em: <https://arxiv.org/abs/2010.16061>.

PRUDÊNCIO, R.; HERNÁNDEZ-ORALLO, J.; MARTINEZ-USÓ, A. Analysis of instance hardness in machine learning using item response theory. In: *Second International Workshop on Learning over Multiple Contexts in ECML*. [S.l.: s.n.], 2015. v. 3.

QUINLAN, R. *Thyroid Disease*. 1987. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5D010.

RASCH, G. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danmarks Paedagogiske Institut, 1960. (Studies in mathematical psychology). ISBN 9780598554512. Disponível em: <https://books.google.com.br/books?id=aB9qLgEACAAJ>.

RAYANA, S. *ODDS Library*. 2016. Disponível em: <https://odds.cs.stonybrook.edu>.

RAYANA, S.; AKOGLU, L. Less is more: Building selective anomaly ensembles. *ACM Trans. Knowl. Discov. Data*, Association for Computing Machinery, New York, NY, USA, v. 10, n. 4, maio 2016. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2890508>.

REINALDO, J. T. de S. *Using Item Response Theory to Evaluate Feature Relevance in Missing Data Scenarios*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, Brazil, mar. 2022. Dissertação de Mestrado em Ciência da Computação. Disponível em: <https://repositorio.ufpe.br/handle/123456789/46381>.

RIJSBERGEN, C. V. *Information Retrieval*. Butterworths, 1979. ISBN 9780408709293. Disponível em: <https://books.google.com.br/books?id=t-pTAAAAMAAJ>.

ROSENBLATT, M. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 27, n. 3, p. 832 – 837, 1956. Disponível em: <https://doi.org/10.1214/aoms/1177728190>.

RUFF, L.; VANDERMEULEN, R.; GOERNITZ, N.; DEECKE, L.; SIDDIQUI, S. A.; BINDER, A.; MÜLLER, E.; KLOFT, M. Deep one-class classification. In: DY, J.; KRAUSE, A. (Ed.). *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 4393–4402. Disponível em: <https://proceedings.mlr.press/v80/ruff18a.html>.

SAMEJIMA, F. Estimation of latent ability using a response pattern of graded scores 1. *ETS Research Bulletin Series*, Wiley Online Library, v. 1968, n. 1, p. i–169, 1968.

SCHLEGL, T.; SEEBÖCK, P.; WALDSTEIN, S. M.; SCHMIDT-ERFURTH, U.; LANGS, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: SPRINGER. *International conference on information processing in medical imaging*. [S.l.], 2017. p. 146–157.

SCHöLKOPF, B.; WILLIAMSON, R. C.; SMOLA, A.; SHAWE-TAYLOR, J.; PLATT, J. Support vector method for novelty detection. In: SOLLA, S.; LEEN, T.; MüLLER, K. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: MIT Press, 1999. v. 12.

SCHöLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J.; SMOLA, A. J.; WILLIAMSON, R. C. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, v. 13, n. 7, p. 1443–1471, 07 2001. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/089976601750264965>.

SHYU, M.-L.; CHEN, S.-C.; SARINNAPAKORN, K.; CHANG, L. A novel anomaly detection scheme based on principal component classifier. In: IEEE PRESS. *Proceedings of the IEEE foundations and new directions of data mining workshop*. [S.l.], 2003. p. 172–179.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: SPRINGER. *Australasian joint conference on artificial intelligence*. [S.l.], 2006. p. 1015–1021.

TANG, J.; CHEN, Z.; FU, A. W.-C.; CHEUNG, D. W. Enhancing effectiveness of outlier detections for low density patterns. In: SPRINGER. *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6*. [S.l.], 2002. p. 535–548.

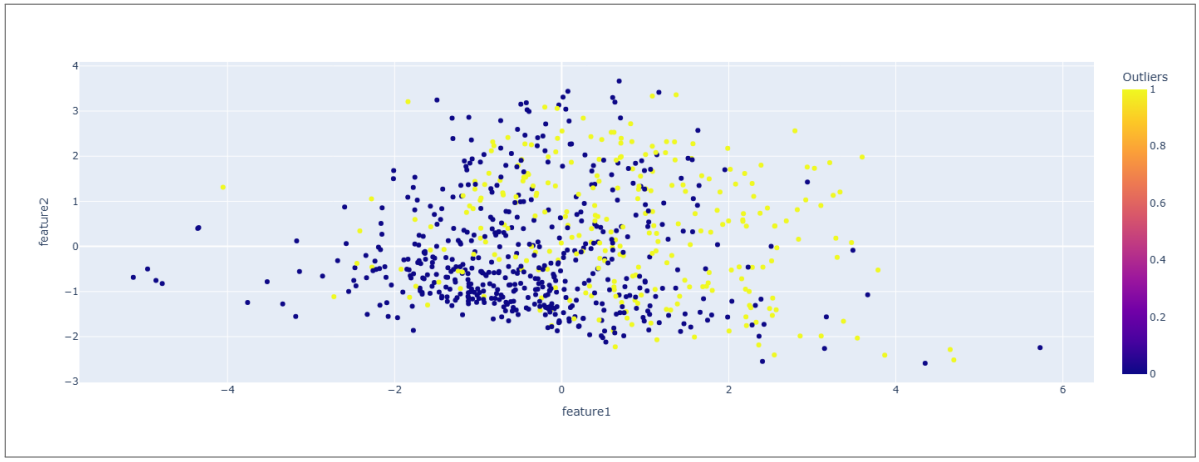
YEUNG, C.-K. *Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory*. 2019. Disponível em: <https://arxiv.org/abs/1904.11738>.

ZHAO, Y.; NASRULLAH, Z.; HRYNIEWICKI, M. K.; LI, Z. Lscp: Locally selective combination in parallel outlier ensembles. In: _____. *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*. [s.n.]. p. 585–593. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9781611975673.66>.

ZHAO, Y.; NASRULLAH, Z.; LI, Z. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, v. 20, n. 96, p. 1–7, 2019. Disponível em: <http://jmlr.org/papers/v20/19-011.html>.

APÊNDICE A – REPRESENTAÇÃO GRÁFICA BIDIMENSIONAL DOS DATASETS

Figura 14 – Conjunto de dados reais públicos chamado *Pima* (proveniente da base *ODDS*) com dimensão reduzida pela técnica PCA. Os pontos em azul representam os *inliers*, já os *outliers* estão em amarelo.



Fonte: Elaborado pelo autor (2024)

APÊNDICE B – VISUALIZAÇÃO TABULAR DOS DADOS GERADOS POR MÓDULOS DO ODAIRE OU EM SEUS EXPERIMENTOS

Figura 15 – Trecho da Matriz de predições obtidas dos Métodos de AD durante treinamento no *dataset* dados reais *Pima*.

PCA	OCSVM	LOF	CBLOF	COF	HBOS	KNN
0	0	1	1	0	0	0
0	0	0	1	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Fonte: Autoria própria.

Figura 16 – Trecho da Matriz de *scores* obtidos nas predições dos Métodos AD ajustados no conjunto de dados reais *Pima*.

PCA	OCSVM	LOF	CBLOF	COF	HBOS	KNN
355.6881581	5.38300705	1.397029366	0.5868272612	1.20107672	-6.821692706	0.3091242471
345.9221235	6.707879501	1.28702005	0.6927785929	1.1726109	-8.604050084	0.3768155042
205.5888355	-2.643995235	1.08151248	0.329217399	1.149300928	-9.555698831	0.2647826179
229.6187867	0.41977756	1.059846565	0.3842122502	1.06373988	-8.560599169	0.2653063538
206.0242148	-0.7692771802	1.180654613	0.2829829811	1.144046969	-9.419118554	0.2459580646
219.8315106	-0.7885909744	1.047938116	0.2700483502	0.9820511613	-14.59856722	0.2139409371
240.4747332	2.12162862	1.04188335	0.2256470819	1.013721503	-10.4712748	0.1995756089
274.2754603	2.641535986	1.253985469	0.4033600133	1.153488101	-8.660913493	0.2733472575
182.3724183	-2.599193099	1.124627319	0.3024995182	1.073909435	-10.60536632	0.2405374725
256.0211359	1.035781385	1.125711533	0.2641358915	1.015925975	-11.04782174	0.2014743419
261.2409356	-0.8165812246	1.166438879	0.370967491	1.138702916	-7.656686748	0.3008556888
166.3996535	-4.136427045	0.9984480812	0.2704716613	0.9987138683	-10.6043237	0.2084519978

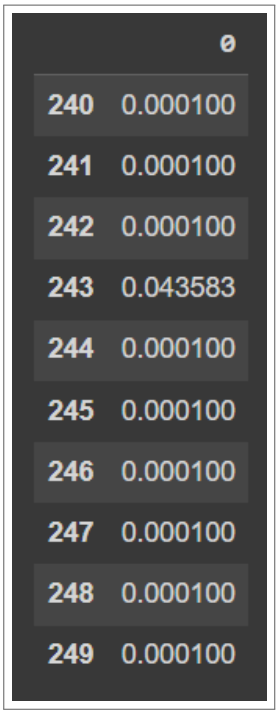
Fonte: Autoria própria.

Figura 17 – Vetor de Habilidades obtido pelo modelo IRT ajustado no sub-conjunto dos dados de teste do dataset *DS4* com o traço latente de todos Métodos de AD utilizados como respondentes.

	IRT
PCA	0.999900
OCSVM	0.999900
LOF	0.999900
CBLOF	0.999900
COF	0.973384
HBOS	0.999900
KNN	0.999900
SOD	0.990543
COPOD	0.996781
ECOD	0.998340
IForest	0.988983
LODA	0.984304
DeepSVDD	0.965578
KDE	0.999900
LOCI	0.999900
AutoEncoder_keras	0.999900
VAE	0.999900
AnoGAN	0.962455
MO_GAAL	0.943685
SO_GAAL	0.942118
GMM	0.999900
LSCP	0.999900
CD	0.993662

Fonte: Autoria própria.

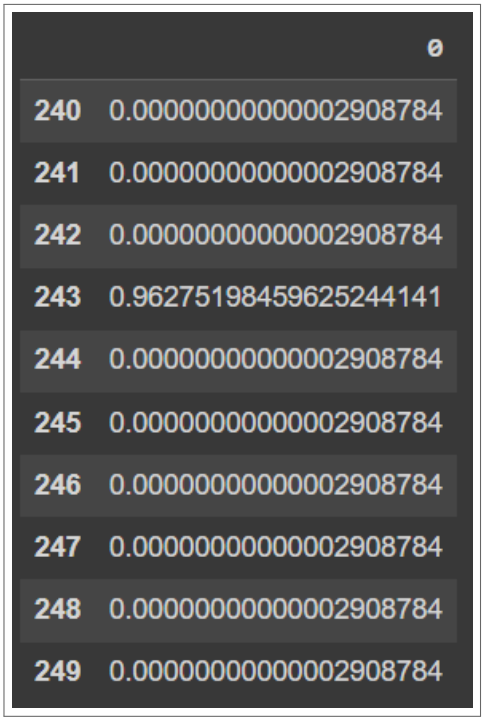
Figura 18 – Trecho do vetor Dificuldades obtido pelo modelo IRT ajustado no dataset *DS4* com o traço latente de apenas 10 instâncias do sub-conjunto dos dados de teste.



	0
240	0.000100
241	0.000100
242	0.000100
243	0.043583
244	0.000100
245	0.000100
246	0.000100
247	0.000100
248	0.000100
249	0.000100

Fonte: Autoria própria.

Figura 19 – Trecho do vetor Discriminação obtido pelo modelo IRT ajustado no dataset *DS4* com o traço latente de apenas 10 instâncias do sub-conjunto dos dados de teste.



	0
240	0.00000000000002908784
241	0.00000000000002908784
242	0.00000000000002908784
243	0.96275198459625244141
244	0.00000000000002908784
245	0.00000000000002908784
246	0.00000000000002908784
247	0.00000000000002908784
248	0.00000000000002908784
249	0.00000000000002908784

Fonte: Autoria própria.

Figura 20 – Trecho da matriz de métricas de avaliação de modelos versus os Modelos AD a partir das predições obtidas no *dataset* com dados reais chamado *Pima*.

	AUC_score	PRECISION_score	RECALL_score	FSCORE_score
PCA	0.5295187166	0.4814814815	0.1390374332	0.2157676349
OCSVM	0.5254163484	0.462962963	0.1336898396	0.2074688797
LOF	0.5213139801	0.4444444444	0.128342246	0.1991701245
CBLOF	0.5213139801	0.4444444444	0.128342246	0.1991701245
COF	0.537723453	0.5185185185	0.1497326203	0.2323651452
HBOS	0.5582352941	0.6111111111	0.1764705882	0.2738589212
KNN	0.5049045073	0.3703703704	0.1069518717	0.1659751037

Fonte: Autoria própria.

Figura 21 – Trecho da Matriz de ranqueamento das Habilidades obtidas pelo modelo IRT a partir das concordâncias entre predições dos Métodos AD ajustados no conjunto de dados reais *Pima*.

	IRT
PCA	11
OCSVM	11
LOF	11
CBLOF	11
COF	16
HBOS	16
KNN	11
SOD	20
COPOD	18
ECOD	14
IForest	11
LODA	21
DeepSVDD	23
KDE	14
LOCI	11
AutoEncoder_keras	11
VAE	11
AnoGAN	18
MO_GAAL	22
SO_GAAL	19
GMM	11
LSCP	14
CD	11

Fonte: Autoria própria.

Quadro 6 – Contagem de valores distintos obtidos após cálculos das médias aritméticas a partir das predições e das concordâncias em cada uma das instâncias do *dataset DS4*.

Matriz de Predições		Matriz de Concordância	
Valor	Frequência	Valor	Frequência
0.000	601	6.000	621
0.142	54	4.285	68
0.285	23	2.571	32
1.000	20	3.142	29
0.428	19		
0.714	14		
0.571	11		
0.857	8		

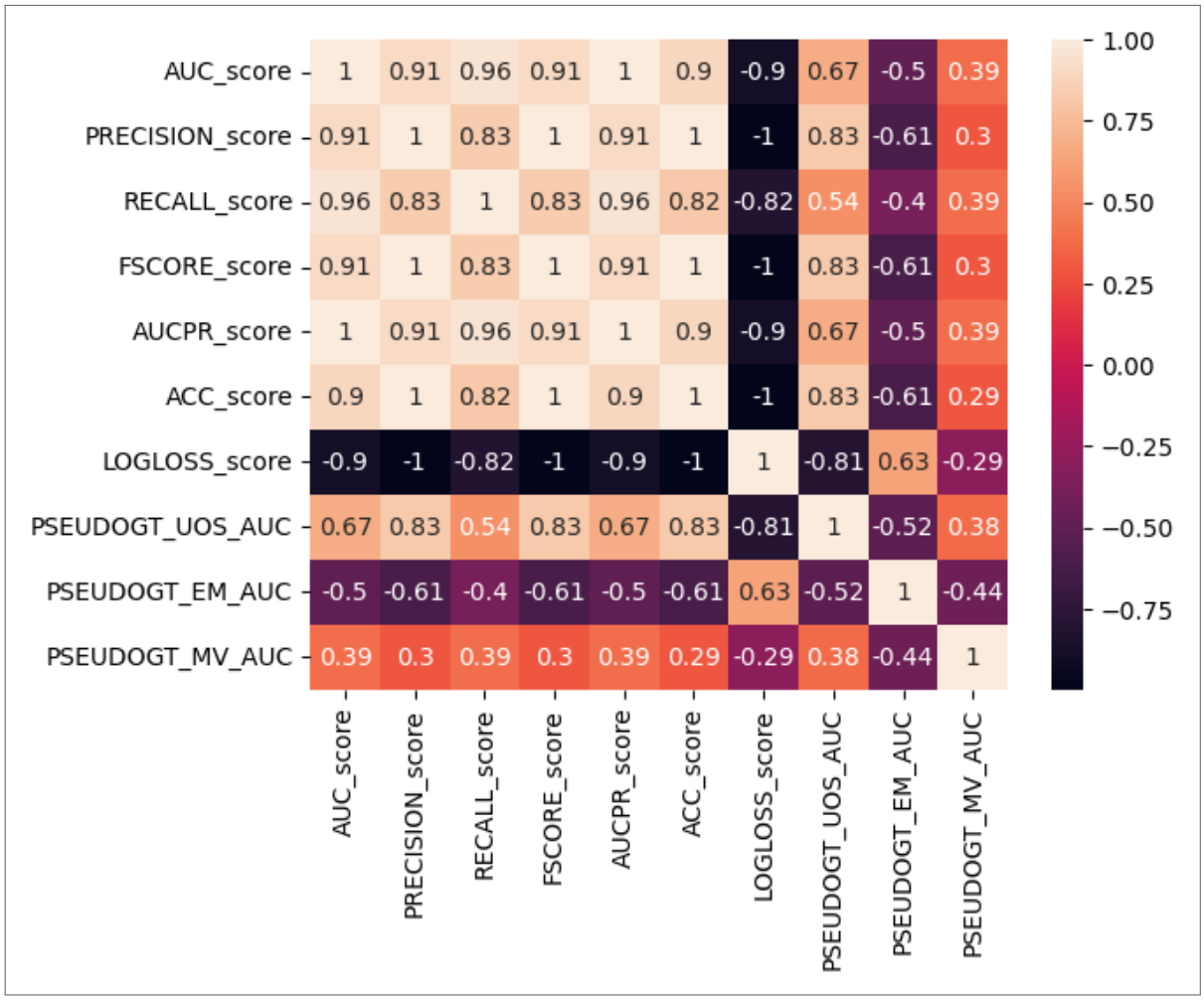
Fonte: Elaborado pelo autor (2024).

Quadro 7 – Trecho da contagem de valores distintos obtidos após cálculos das médias ponderadas (possuindo Habilidade IRT como pesos) a partir das predições e das concordâncias em cada uma das instâncias do *dataset DS4*

Matriz de Predições		Matriz de Concordância	
Valor	Frequência	Valor	Frequência
0.000	601	6.000	621
0.025	27	4.871	29
1.000	20	4.242	18
0.151	14	4.182	16
0.163	12	3.432	12
0.352	7	2.647	11
0.316	6	2.517	8
0.810	5	2.645	4

Fonte: Elaborado pelo autor (2024).

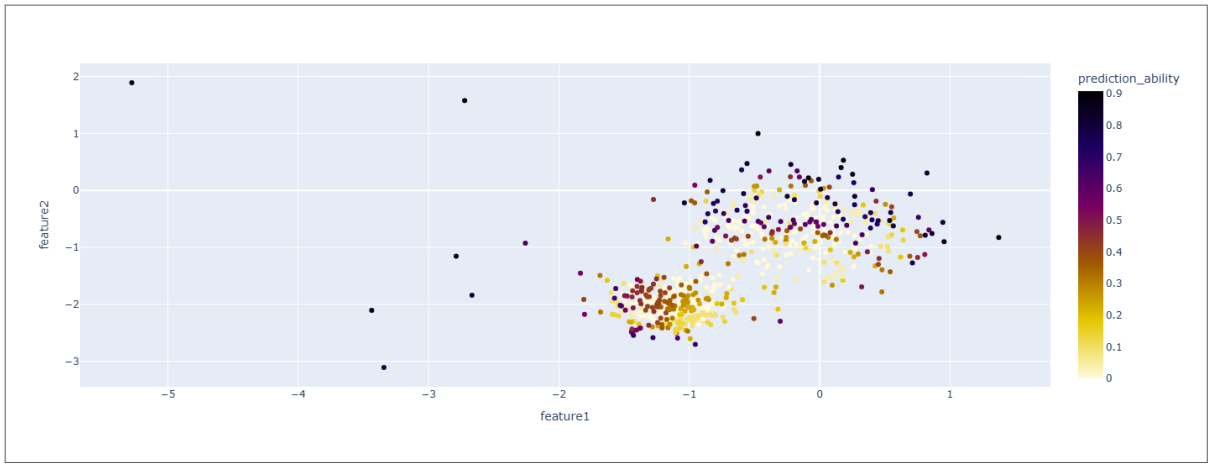
Figura 22 – Gráfico do mapa de calor representando a tabela de correlações, utilizando o coeficiente de Spearman, entre os valores de ranqueamento/postos de todas as métricas utilizadas, durante avaliação de Modelos AD ajustados ao *dataset DS4*.



Fonte: Autoria própria.

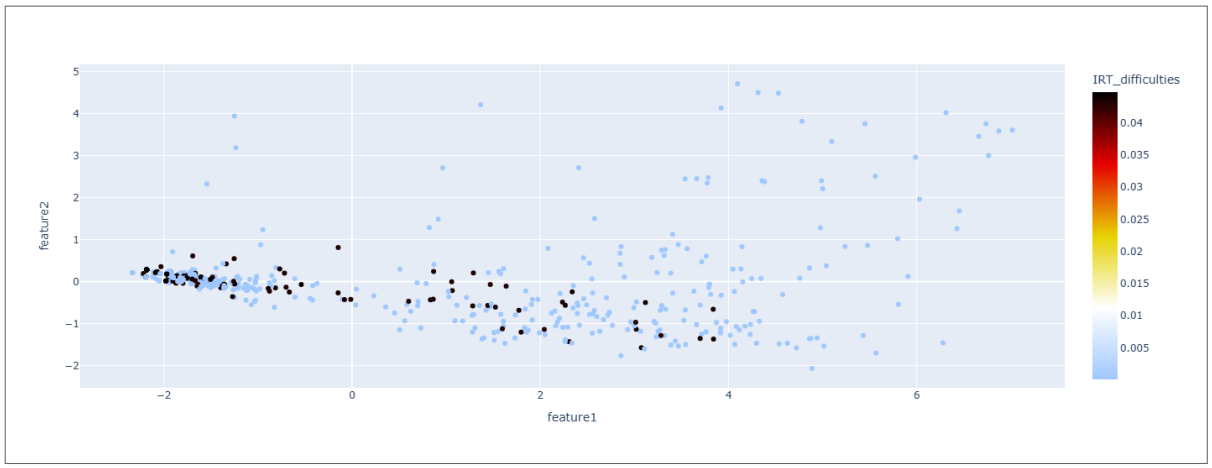
APÊNDICE C – VISUALIZAÇÃO GRÁFICA BIDIMENSIONAL DOS DATASETS E SEUS TRAÇOS LATENTES IRT

Figura 23 – Gráfico de dispersão 2D representando média ponderada de predições dos Métodos AD com traço latente *Habilidade* (IRT) em cada instância do *dataset* sintético *DS4*



Fonte: Autoria própria.

Figura 24 – Trecho do gráfico de dispersão em espaço 2D, com dimensão reduzida pela técnica PCA, representando traço latente *Dificuldade* em cada instância do *dataset Breastw* (sub-conjuntos de treino e teste).



Fonte: Autoria própria.

ANEXO A – DADOS REAIS PÚBLICOS UTILIZADOS

A.1 ANNTHYROID

O conjunto de dados *Ann-Thyroid*, disponível no repositório *UCI*, é utilizado para a detecção de anomalias, especificamente para identificar pacientes com condições anormais da glândula tireoide. Este *dataset* é composto por 7200 instâncias e 21 atributos, que incluem várias medições clínicas e bioquímicas. A variável alvo é binária, distinguindo entre condições normais e anormais da tireoide. O *dataset Ann-Thyroid* é frequentemente usado para avaliar e comparar a eficácia de algoritmos de detecção de anomalias, sendo uma ferramenta valiosa para pesquisas em ML e diagnóstico médico. (RAYANA, 2016)

A.2 BREASTW

O conjunto de dados *Breast Cancer Wisconsin* (*breastw*), disponível no repositório *UCI*, é utilizado para a classificação de tumores de mama como benignos ou malignos. Este dataset contém 699 instâncias, cada uma descrita por 9 atributos preditivos derivados de uma imagem digitalizada de uma biópsia aspirativa por agulha fina (FNA) de uma massa mamária. Os atributos incluem características como o espessamento do tecido, uniformidade do tamanho e forma das células, adesão marginal, entre outros. A variável alvo é binária, indicando se o tumor é benigno ou maligno. Este conjunto de dados é amplamente utilizado para treinar e avaliar algoritmos de classificação em aplicações de diagnóstico médico, contribuindo para avanços na detecção precoce do câncer de mama. (RAYANA, 2016)

A.3 CARDIO

O conjunto de dados *Cardiotocography* (*cardio*), disponível no repositório *UCI*, é utilizado para a classificação de traçados de cardiotocografia fetal, que monitoram a frequência cardíaca fetal e as contrações uterinas durante a gravidez. Este *dataset* contém 2126 instâncias, cada uma descrita por 21 atributos que incluem medições como a variabilidade da frequência cardíaca, acelerações, desacelerações e padrões das contrações. A variável alvo categoriza os traçados em três classes: normais, suspeitos e patológicos. O *dataset cardio* é valioso para o desenvolvimento de modelos de ML destinados a avaliar a saúde fetal e ajudar na tomada de

decisões clínicas durante o pré-natal. (RAYANA, 2016)

A.4 GLASS

O conjunto de dados *Glass Identification* (*glass*), disponível no repositório *UCI*, é utilizado para a classificação de diferentes tipos de vidro com base em suas propriedades químicas. Este *dataset* contém 214 instâncias e 9 atributos, que representam as porcentagens em peso de diferentes elementos presentes no vidro, como óxidos de silício, sódio, magnésio, alumínio, sílica, potássio, cálcio, bário e ferro. A variável alvo classifica o vidro em uma das seis categorias, incluindo vidros de construção de janelas, vidros de veículos e vidros de recipientes. Este conjunto de dados é amplamente utilizado em pesquisas de ML e mineração de dados, especialmente em aplicações de identificação forense e industrial, onde a composição química do vidro pode fornecer informações críticas. (RAYANA, 2016)

A.5 IONOSPHERE

O conjunto de dados *Ionosphere*, disponível no repositório *UCI*, é utilizado para a classificação de retornos de radar da ionosfera, uma região da atmosfera terrestre que influencia a transmissão de sinais de rádio. Este *dataset* contém 351 instâncias, cada uma descrita por 34 atributos, que são valores numéricos derivados de medições de sinais de radar. A variável alvo é binária, indicando se o retorno do radar mostra a presença de estrutura na ionosfera (classe "*good*") ou não (classe "*bad*"). O conjunto de dados *Ionosphere* é frequentemente usado para desenvolver e avaliar algoritmos de classificação, sendo uma ferramenta valiosa para pesquisas em processamento de sinais e ML aplicados à meteorologia e comunicação. (RAYANA, 2016)

A.6 LETTER

O conjunto de dados *Letter Recognition* (*letter*), disponível no repositório *UCI*, é utilizado para a classificação de letras do alfabeto romano maiúsculo com base em suas características tipográficas. Este dataset contém 20.000 instâncias, cada uma representando uma letra, descrita por 16 atributos numéricos que capturam várias propriedades das letras, como altura, largura, ângulos e número de cruzamentos. A variável alvo é categórica, indicando qual letra (de A a Z) cada instância representa. O conjunto de dados letter é amplamente utilizado

para treinar e avaliar algoritmos de reconhecimento de padrões e ML, sendo uma ferramenta essencial em pesquisas sobre Reconhecimento Óptico de Caracteres (*OCR*) e processamento de imagens. (RAYANA, 2016)

A.7 MUSK

O conjunto de dados *Musk*, disponível no repositório *UCI*, é utilizado para a classificação de moléculas como almíscar ou não-almíscar, com base em suas características químicas. Este *dataset* contém duas versões: *Musk* (*Version* 1) com 476 instâncias e 168 atributos, e *Musk* (*Version* 2) com 6598 instâncias e 166 atributos. Os atributos representam propriedades moleculares e descritores químicos que ajudam a identificar a fragrância de almíscar. A variável alvo é binária, indicando se a molécula possui a característica de almíscar ou não. O conjunto de dados *Musk* é amplamente utilizado em pesquisas de e química computacional, sendo uma referência para o desenvolvimento e avaliação de modelos de classificação na identificação de compostos químicos com características específicas. (RAYANA, 2016)

A.8 PIMA

Também chamado de *Pima Indians diabetes dataset*, é um conjunto de dados de classificação binária. Amplamente utilizado na pesquisa de ML, focado na predição da diabetes em mulheres da etnia *Pima*, com idade a partir de 21 anos. O conjunto de dados contém 768 instâncias e 8 atributos preditivos, incluindo número de gestações, glicose, pressão sanguínea, espessura da dobra cutânea do tríceps, insulina, Índice de Massa Corporal (IMC), histórico familiar de diabetes (*pedigree*) e idade. A variável alvo é binária, indicando a presença ou ausência de diabetes, diagnosticada de acordo com critérios do *National Institute of Diabetes and Digestive and Kidney Diseases*. Este *dataset* é valioso para o desenvolvimento e avaliação de modelos de classificação na área médica. (RAYANA, 2016)

A.9 SPEECH

O conjunto de dados "*Speech*"do repositório *UCI* foi projetado para auxiliar no desenvolvimento e avaliação de algoritmos de processamento de sinal para reconhecimento de fala. Este *dataset* contém gravações de voz de diferentes falantes, expressando várias frases em inglês.

Os dados são apresentados com múltiplos atributos acústicos derivados das gravações, tais como frequências fundamentais, intensidade, e características espectrais que são cruciais para o reconhecimento automático de fala. A variável alvo frequentemente se concentra em categorizar ou identificar características específicas da fala, como emoção, intenção ou identidade do falante. O conjunto de dados "*Speech*"é amplamente usado por pesquisadores na área de AI e engenharia de áudio para melhorar tecnologias de assistentes de voz, sistemas de segurança por biometria vocal e outras aplicações interativas baseadas em voz. (RAYANA, 2016)

A.10 THYROID

O conjunto de dados "*Thyroid*", disponível no repositório *UCI*, é amplamente utilizado para o diagnóstico de distúrbios da tireoide. Este dataset inclui diversas instâncias com múltiplos atributos clínicos e laboratoriais que ajudam a identificar diferentes condições tireoidianas, como hipotireoidismo, hipertireoidismo e estados eutireoidianos. Os atributos incluem medidas hormonais, como os níveis de TSH e T3, além de dados demográficos e históricos médicos dos pacientes. A variável alvo é tipicamente multiclasse, categorizando os casos em diferentes tipos de disfunções tireoidianas. Este conjunto de dados é essencial para o desenvolvimento de modelos de aprendizado de máquina na medicina, facilitando a automação do diagnóstico e melhorando a precisão do tratamento de doenças tireoidianas. (QUINLAN, 1987)

A.11 VERTEBRAL

O conjunto de dados "*Vertebral Column*", disponível no repositório *UCI*, é utilizado para a classificação de pacientes com base em diagnósticos ortopédicos relacionados à coluna vertebral. Este *dataset* contém 310 instâncias, cada uma com 6 atributos biomecânicos derivados de imagens de ortopedia, como ângulos de inclinação pélvica, inclinação sacral, e curvatura lombar. Os pacientes são classificados em três categorias: Normal, Hernia de Disco, ou Espondilolistese. A variável alvo é categórica e visa auxiliar na determinação da condição ortopédica específica do paciente. O *dataset* "Vertebral Column"é frequentemente utilizado em pesquisas de ML para desenvolver algoritmos que ajudam na diagnóstico precoce e na classificação precisa de condições que afetam a coluna vertebral, melhorando assim as estratégias de tratamento e intervenção. (RAYANA, 2016)

A.12 VOWELS

O conjunto de dados "*Vowels*"do repositório *UCI* é focado na tarefa de reconhecimento de vogais faladas, sendo utilizado para avaliar e desenvolver algoritmos de ML na área de NLP e reconhecimento de fala. Este *dataset* contém gravações de 990 instâncias de vogais pronunciadas por 15 locutores nativos do japonês, com cada locutor representando 6 diferentes vogais japonesas. Os dados são representados por 12 coeficientes que descrevem a frequência e outras características acústicas das vogais. A variável alvo é categórica, identificando qual das seis vogais japonesas está sendo falada. O conjunto de dados "*Vowels*"é amplamente utilizado para experimentos em reconhecimento de padrões vocais e técnicas de classificação de fala, ajudando a aprimorar sistemas de assistência por voz e tecnologias de interação homem-máquina baseadas em reconhecimento vocal. (RAYANA, 2016)

A.13 WBC

O conjunto de dados "*Wisconsin Breast Cancer*"(*WBC*), disponível no repositório *UCI*, é essencial para a pesquisa e desenvolvimento de algoritmos de ML aplicados ao diagnóstico de câncer de mama. O *dataset* inclui 699 instâncias de amostras de células de biópsias de tumores de mama, detalhadas por 9 atributos baseados em características celulares visualizadas em imagens microscópicas, como a espessura do agrupamento, uniformidade do tamanho celular e adesividade celular. A variável alvo é binária, distinguindo entre tumores benignos e malignos. O *dataset WBC* é frequentemente utilizado para treinar e testar modelos de classificação, ajudando a melhorar a precisão do diagnóstico precoce do câncer de mama e a eficácia das intervenções médicas. (RAYANA, 2016)

A.14 WINE

O conjunto de dados "*Wine*"do repositório *UCI* é utilizado para classificar três tipos diferentes de vinhos cultivados na mesma região da Itália, mas derivados de três diferentes cultivares. O *dataset* contém 178 instâncias e 13 atributos químicos, incluindo teor alcoólico, ácido málico, cinzas, alcalinidade das cinzas, magnésio, fenóis totais, flavonoides, fenóis não flavonoides, proantocianidinas, intensidade da cor, matiz, OD280/OD315 de vinhos diluídos, e prolinas. A variável alvo é categórica, indicando o tipo de vinho. Este conjunto de dados é

amplamente utilizado em pesquisas de ML para desenvolver e avaliar algoritmos de classificação, sendo um exemplo clássico para experimentos de análise multivariada e reconhecimento de padrões. (AEBERHARD; FORINA, 1991)

A.15 YEAST

O conjunto de dados "*Yeast*"do repositório *UCI* é utilizado para classificar diferentes tipos de proteínas de levedura com base em sua localização celular. Este *dataset* contém 1484 instâncias e 8 atributos preditivos, que incluem características como erro de sequência, conteúdo alifático, e frequência de ocorrência de aminoácidos ácidos e básicos. A variável alvo é categórica, classificando as proteínas em 10 diferentes classes de localização celular, como citoplasma, membrana nuclear, mitocôndria, entre outras. O *dataset* "*Yeast*"é frequentemente utilizado para pesquisas em bioinformática e ML, fornecendo uma base sólida para o desenvolvimento de métodos de predição de funções e localizações proteicas, crucial para entender a biologia celular e o desenvolvimento de novas terapias. (RAYANA, 2016)

ANEXO B – MÉTODOS DE DETECÇÃO DE ANOMALIAS UTILIZADOS

B.1 PRINCIPAL COMPONENT ANALYSIS (PCA) OUTLIER DETECTOR

A PCA (SHYU et al., 2003), lançada em 2003 para fins de AD, e pertencente aos métodos rasos representativos, é uma técnica comumente usada para detectar subespaços em conjuntos de dados. Também pode servir como uma técnica de AD, de modo que desvios dos subespaços normais possam indicar instâncias anômalas. Os componentes principais são os autovetores da matriz de covariância e, portanto, seu cálculo compartilha as mesmas dificuldades do CMGOS (Pontuação *Outlier* Gaussiana Multivariada baseada em *cluster*): As anomalias têm uma grande influência na matriz de covariância e a estimativa de densidade pode estar incorreta. Quase idêntica à técnica de redução do CMGOS, uma versão robusta (*rPCA*) foi proposta em (KWITT; HOFMANN, 2007), que também calcula a matriz de covariância duas vezes com base na distância de *Mahalanobis*. Uma vez determinados os componentes principais, a questão é quais componentes devem ser usados para pontuar instâncias anômalas. O uso dos componentes principais mostra desvios globais da maioria dos dados, enquanto o uso de componentes menores pode indicar desvios locais menores. Na nossa avaliação seguimos estratégias diferentes, nomeadamente usando todos os componentes, usando apenas os componentes principais, usando apenas os componentes menores e, finalmente, usando os componentes maiores e menores, negligenciando os do meio (SHYU et al., 2003). Observe que há uma forte conexão entre *rPCA* e *CMGOS-Red*: Quando o *rPCA* leva todos os componentes em consideração, o método é basicamente o mesmo que aplicar CMGOS com configuração $k = 1$ nos parâmetros (GOLDSTEIN; UCHIDA, 2016).

B.2 ONE-CLASS SUPPORT VECTOR MACHINES (OCSVM)

OCSVM (SCHöLKOPF et al., 1999)(SCHöLKOPF et al., 2001), lançado em 2001 para fins de AD e pertencente aos métodos rasos representativos, são frequentemente usados para detecção de anomalias semi-supervisionadas . Nesta configuração, uma Máquina de Vetores de Suporte (do inglês *Support-Vector Machine*) de uma classe é treinado em dados livres de anomalias e, posteriormente, o SVM classifica anomalias e dados normais no conjunto de teste. SVMs de classe única pretendem separar a origem das instâncias de dados no espaço do *kernel*, o que resulta em algum tipo de estrutura complexa que descreve os dados normais no espaço

de recursos. Embora SVMs de classe única sejam muito usados como método de detecção de anomalias semi-supervisionados, é um algoritmo não supervisionado por *design* ao usar uma margem suave. Em particular, foi demonstrado que eles convergem para o verdadeiro nível de densidade definido . No cenário de detecção de anomalia não supervisionada, o SVM de uma classe é treinado usando o conjunto de dados e, posteriormente, cada instância no conjunto de dados é pontuada por uma distância normalizada até o limite de decisão determinado. . O parametro 'v' precisa ser definido com um valor maior que zero, de modo que as anomalias contidas sejam tratadas corretamente por uma margem suave. Além disso, os SVMs de uma classe foram modificados para incluir técnicas mais robustas para lidar explicitamente com valores discrepantes durante o treinamento. . A ideia basica é que as anomalias contribuem menos para o limite de decisão do que as instâncias normais. Duas técnicas diferentes foram desenvolvidas, enquanto o SVM de classe única 'n' apresentou resultados superiores. Neste aprimoramento, 'n' é um objetivo adicional de otimização durante o treinamento, que estima a normalidade de uma instância. Em nossa avaliação, estamos usando algoritmos não supervisionados, o SVM regular de uma classe, bem como o SVM estendido de uma classe 'n' (GOLDSTEIN; UCHIDA, 2016).

B.3 LOCAL OUTLIER FACTOR (LOF)

O LOF (BREUNIG et al., 2000) , lançado em 2000 para fins de AD e pertencente aos métodos rasos representativos, é o algoritmo de detecção de anomalias locais mais conhecido e também introduziu primeiro a ideia de anomalias locais. Hoje, sua ideia é implementada em muitos algoritmos baseados no vizinho mais próximo, como os descritos abaixo. Para calcular a pontuação LOF, três etapas devem ser calculadas:

- Os k vizinhos mais próximos devem ser encontrados para cada registro x. No caso de empate de distância do k-ésimo vizinho, mais de k vizinhos são usados.
- Usando esses k-vizinhos mais próximos Nk, a densidade local para um registro é estimada calculando a Densidade de Alcançabilidade Local (LRD), enquanto $dk()$ é a distância de alcançabilidade. Exceto por algumas situações muito raras em aglomerados altamente densos, esta é a distância euclidiana.
- Finalmente, a pontuação LOF é calculada comparando o LRD de um registro com os

LRDs de seus k vizinhos:

A pontuação LOF é, portanto, basicamente uma proporção de densidades locais. Isso resulta na bela propriedade do LOF, de que instâncias normais, cujas densidades são tão grandes quanto as densidades de seus vizinhos, obtêm uma pontuação de cerca de 1,0. Anomalias, que possuem baixa densidade local, resultarão em pontuações maiores. Neste ponto também fica claro por que esse algoritmo é local: ele depende apenas de sua vizinhança direta e a pontuação é uma razão baseada principalmente apenas nos k vizinhos. É claro que anomalias globais também podem ser detectadas, uma vez que também têm um LRD baixo quando comparado com os seus vizinhos. É importante notar que em tarefas de AD, onde anomalias locais não são de interesse, este algoritmo irá gerar muitos alarmes falsos como descobrimos durante a nossa avaliação. Novamente, a configuração de k é crucial para este algoritmo. Além de testar diferentes valores para k, os autores do algoritmo sugeriram usar uma estratégia de conjunto para calcular o LOF (BREUNIG et al., 2000). Aqui, as pontuações para diferentes k's até um limite superior são calculadas e, em seguida, o máximo dessas pontuações é obtido. Além de calcular a pontuação LOF para um único k, também levamos essa estratégia em consideração em nossa avaliação, referindo-nos a ela como *LOF-UB* (limite superior). Por motivos de comparação, também usamos diferentes limites superiores e calculamos novamente a média dos resultados (GOLDSTEIN; UCHIDA, 2016).

B.4 CLUSTERING-BASED LOCAL OUTLIER FACTOR (CBLOF)

Todos os algoritmos anteriores de detecção de anomalias são baseados em estimativa de densidade usando vizinhos mais próximos. Em contraste, o fator de anomalia local baseado em *clusters* (CBLOF) (HE; XU; DENG, 2003) usa *clustering* para determinar áreas densas nos dados e realiza uma estimativa de densidade para cada *cluster* posteriormente. O CBLOF foi lançado em 2003 para fins de detecção de anomalias e pertence aos métodos rasos representativos. Em teoria, qualquer algoritmo de *clustering* pode ser usado para agrupar os dados em um primeiro passo. No entanto, na prática, o *k-means* é comumente utilizado para aproveitar a baixa complexidade computacional, que é linear em comparação com a complexidade quadrática da busca por vizinhos mais próximos. Após o *clustering*, o CBLOF usa uma heurística para classificar os *clusters* resultantes em *clusters* grandes e pequenos. Finalmente, uma pontuação de anomalia é calculada pela distância de cada instância ao centro do *cluster* multiplicada pelo

número de instâncias pertencentes ao seu *cluster*. Para *clusters* pequenos, usa-se a distância até o *cluster* grande mais próximo. O procedimento de usar a quantidade de membros do *cluster* como fator de escala deve estimar a densidade local dos *clusters*, conforme afirmado pelos autores. Mostramos em trabalhos anteriores que essa suposição não é verdadeira e pode até resultar em uma estimativa incorreta de densidade. Portanto, também avaliamos uma versão modificada do CBLOF que simplesmente negligencia o peso, chamada de *unweighted-CBLOF* (*uCBLOF*) a partir de agora. Similar aos algoritmos baseados em vizinhos mais próximos, o número de *clusters* iniciais k também é um parâmetro crítico. Aqui, seguimos a mesma estratégia que para os algoritmos baseados em vizinhos mais próximos e avaliamos muitos valores diferentes de k. Além disso, o *clustering k-means* é um algoritmo não determinístico e, portanto, as pontuações de anomalia resultantes podem ser diferentes em várias execuções. Para isso, seguimos uma estratégia comum, que é aplicar o *k-means* várias vezes nos dados e escolher o resultado mais estável. No entanto, algoritmos de detecção de anomalias baseados em *clustering* são muito sensíveis ao parâmetro k, uma vez que adicionar apenas um único centroide adicional pode levar a um resultado muito diferente (GOLDSTEIN; UCHIDA, 2016).

B.5 CONNECTIVITY-BASED OUTLIER FACTOR (COF)

O Fator de Anomalia Baseado em Conectividade (TANG et al., 2002), lançado em 2002 para fins de AD e pertencente ao método raso representativo, é semelhante ao LOF, mas a estimativa de densidade para os registros é realizada de maneira diferente. No LOF, os -vizinhos mais próximos são selecionados com base na distância euclidiana. Isso pressupõe indiretamente que os dados estão distribuídos de forma esférica ao redor da instância. Se essa suposição for violada, por exemplo, se as características tiverem uma correlação linear direta, a estimativa de densidade estará incorreta. O COF deseja compensar essa deficiência e estima a densidade local do bairro usando uma abordagem de menor caminho, chamada de distância de encadeamento. Matematicamente, essa distância de encadeamento é o mínimo da soma de todas as distâncias que conectam todos os vizinhos à instância. Para exemplos simples, onde as características estão claramente correlacionadas, essa abordagem de estimativa de densidade é muito mais precisa. Pode-se observar que a estimativa de densidade esférica do LOF não consegue detectar a anomalia, mas o COF obteve sucesso ao conectar os registros normais entre si para estimar a densidade local (GOLDSTEIN; UCHIDA, 2016).

B.6 HISTOGRAM-BASED OUTLIER SCORE (HBOS)

O *Histogram-Based Outlier Score* (HBOS) é um algoritmo de AD que utiliza a distribuição dos atributos dos dados para identificar *outliers*. HBOS assume que as características são independentes entre si e utiliza histogramas para modelar a distribuição de cada característica. Para cada instância no conjunto de dados, o algoritmo calcula a altura inversa do intervalo do histograma onde o valor da característica da instância cai, representando a densidade do valor da característica. Essas densidades inversas são então multiplicadas entre todas as características para calcular uma pontuação geral de anomalia para a instância.

HBOS é significativamente mais rápido em comparação com métodos baseados em vizinhos mais próximos, sendo adequado para aplicações em larga escala devido à sua simplicidade e eficiência. No entanto, a suposição de independência das características pode fazer com que o HBOS perca a AD que surgem devido a interações entre características. Além disso, a escolha do número de intervalos para os histogramas pode impactar significativamente o desempenho do algoritmo, sendo que larguras de intervalos dinâmicas são frequentemente usadas para tornar a estimativa de densidade mais robusta na presença de valores atípicos (GOLDSTEIN; UCHIDA, 2016).

B.7 K-NEAREST NEIGHBOR (KNN)

O *K-Nearest Neighbor* (*k-NN*) é um algoritmo de AD baseado em vizinhos próximos. Ele identifica anomalias calculando a distância de cada ponto de dados aos seus vizinhos mais próximos. Existem duas variações principais: uma que utiliza a distância ao k-ésimo vizinho mais próximo e outra que utiliza a média das distâncias aos k vizinhos mais próximos. O algoritmo atribui uma pontuação de anomalia a cada ponto de dados com base nessas distâncias, sendo que valores mais altos indicam maior probabilidade de anomalia.

A escolha do valor de k é crucial para o desempenho do *k-NN*. Valores de k muito baixos podem resultar em estimativas de densidade não confiáveis, enquanto valores muito altos podem suavizar demais a estimativa, perdendo a capacidade de detectar anomalias locais. Em aplicações práticas, o *k-NN* é frequentemente utilizado por sua simplicidade e eficácia, apesar de ser computacionalmente intensivo para grandes conjuntos de dados, devido à necessidade de calcular distâncias entre todos os pares de pontos (GOLDSTEIN; UCHIDA, 2016).

B.8 SUBSPACE OUTLIER DETECTION (SOD)

A Detecção de *Outliers* em Subespaços (*Subspace Outlier Detection*) é uma técnica utilizada para identificar anomalias em conjuntos de dados de alta dimensionalidade, onde os *outliers* podem estar ocultos em subespaços de dimensões mais baixas. Essa abordagem é necessária porque, em alta dimensionalidade, muitos atributos podem ser irrelevantes ou ruidosos, o que dilui a precisão das medições de distância e torna os escores de *outliers* menos distinguíveis. Portanto, ao invés de analisar o conjunto de dados em sua totalidade, a detecção de *outliers* em subespaços foca em encontrar subespaços relevantes onde o comportamento anômalo dos pontos de dados é mais pronunciado.

Existem várias abordagens para a detecção de *outliers* em subespaços, incluindo métodos baseados em raridade, métodos não enviesados e métodos baseados em agregação. Os métodos baseados em raridade tentam descobrir subespaços raros, enquanto os métodos não enviesados amostram subespaços de maneira aleatória e combinam *scores* de diferentes subespaços. Já os métodos baseados em agregação utilizam estatísticas agregadas, como estatísticas de *clusters*, para quantificar a relevância dos subespaços. Uma característica comum dessas abordagens é a utilização de técnicas de *ensemble* para combinar resultados de múltiplos subespaços, aumentando a robustez e a precisão na identificação de *outliers* (AGGARWAL, 2017).

B.9 COPULA-BASED OUTLIER DETECTION (COPOD)

COPOD (*Copula-Based Outlier Detector*) é um algoritmo de detecção de *outliers* que utiliza cópulas para modelar a dependência entre variáveis em um conjunto de dados. Cópulas são funções matemáticas que permitem capturar a estrutura de dependência entre variáveis, separando essa dependência das distribuições marginais das variáveis individuais. O COPOD avalia a probabilidade conjunta de cada ponto de dados, considerando-o um *outlier* se essa probabilidade for muito baixa, indicando que o ponto é raro ou anômalo no contexto da distribuição modelada.

Esse método é não paramétrico, o que significa que não assume uma distribuição específica dos dados, tornando-o flexível para diferentes tipos de dados e aplicável em situações com dependências complexas entre variáveis. O COPOD é útil em diversas áreas, como detecção de fraudes e monitoramento de redes, onde a identificação de anomalias ou dados fora do padrão é essencial. (LI et al., 2020)

B.10 EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTIONS (ECOD)

O ECOD (*Empirical Cumulative Distribution-based Outlier Detection*) é um algoritmo de detecção de *outliers* baseado na distribuição acumulada empírica, conforme descrito no livro "*Handbook of Anomaly Detection*". Este método parte do princípio que os *outliers* são eventos raros que se encontram nas extremidades de uma distribuição. Para identificar esses *outliers*, o ECOD primeiro estima a distribuição de uma variável de forma não-paramétrica e depois multiplica as probabilidades de cauda estimadas de todas as dimensões para obter a pontuação de anomalia de uma instância.

O ECOD assume que as variáveis são independentes para poder estimar a distribuição acumulada empírica de cada variável. Embora a suposição de independência das variáveis possa ser considerada restritiva, o ECOD é eficaz e se destaca por não precisar de ajuste de hiperparâmetros, sendo rápido na manipulação de grandes volumes de dados.

Uma vantagem significativa do ECOD é sua fácil interpretação. Ele permite inspecionar como cada uma das probabilidades de cauda contribui para a pontuação final de *outlier*, tornando-o uma ferramenta útil para a detecção de anomalias em grandes conjuntos de dados. (KUO, 2023)

B.11 ISOLATION FOREST (IFOREST)

O *Isolation Forest* (*iForest*) é um algoritmo baseado em florestas aleatórias usado para a detecção de *outliers*. Ele funciona de maneira diferente dos métodos tradicionais que se concentram em perfis de normalidade de dados. Em vez disso, o *iForest* isola anomalias introduzindo divisões recursivas nos dados de maneira aleatória. A lógica por trás do *iForest* é que as anomalias são menos frequentes e estão mais isoladas em comparação com os pontos de dados normais, portanto, elas serão isoladas mais rapidamente (KUO, 2023).

O processo básico do *iForest* envolve a construção de uma floresta de árvores de isolamento. Cada árvore é construída por divisões aleatórias de amostras de dados até que cada ponto de dados seja isolado. A profundidade média necessária para isolar um ponto é então usada como uma medida de quão anômalo é o ponto: pontos que exigem menos divisões para serem isolados são considerados mais anômalos.

Essa abordagem tem várias vantagens, incluindo a capacidade de lidar eficientemente com grandes volumes de dados e a falta de necessidade de suposições sobre a distribuição dos

dados. O iForest é especialmente eficaz em cenários de alta dimensionalidade, onde métodos baseados em distância ou densidade podem falhar (AGGARWAL, 2017).

B.12 LIGHTWEIGHT ON-LINE DETECTOR OF ANOMALIES (LODA)

LODA (*Lightweight On-line Detector of Anomalies*) é um método que utiliza projeções aleatórias unidimensionais para construir histogramas, que são posteriormente utilizados para calcular as pontuações de *outliers*. O método se destaca por sua leveza e eficiência, sendo particularmente útil para detecção de anomalias em fluxos de dados (*streams*) devido à sua capacidade de operar em tempo real.

A técnica de LODA consiste em projetar os dados em várias direções aleatórias e, em seguida, construir histogramas sobre essas projeções. As pontuações de anomalia para cada ponto de dados são calculadas como a média das pontuações de log-verossimilhança obtidas a partir dos diferentes histogramas. Mesmo que os histogramas unidimensionais sejam fracos detectores de anomalias de forma isolada, a combinação desses detectores fracos em um ensemble resulta em um detector robusto e eficaz para identificar *outliers*.

Essa abordagem é comparável a outras técnicas de *ensemble*, como o "*rotated bagging*", mas se diferencia ao utilizar projeções unidimensionais em vez de projeções de dimensões mais altas (AGGARWAL, 2017).

B.13 DEEP ONE-CLASS CLASSIFICATION (DEEPSVDD)

Deep One-Class Classification, também conhecido como *Deep Support Vector Data Description* (DeepSVDD), é uma técnica de Aprendizado Profundo desenvolvida para lidar com problemas de AD em dados. Ao contrário de métodos tradicionais que se baseiam em modelos supervisionados com múltiplas classes, o DeepSVDD é projetado para cenários onde há apenas uma classe conhecida durante o treinamento, normalmente representando um comportamento "normal". O objetivo é aprender uma representação compacta dos dados normais no espaço latente, de forma que novas amostras que desviam significativamente dessa distribuição possam ser identificadas como anomalias (AGGARWAL, 2017).

O método utiliza Redes Neurais Profundas para mapear os dados de entrada em um espaço latente de menor dimensão, onde um centro é definido, e a rede é treinada para minimizar a distância entre as representações dos dados normais e esse centro. Durante a inferência,

amostras que se afastam muito desse centro são classificadas como anômalas. Essa abordagem é particularmente útil em aplicações onde dados anômalos são raros ou não disponíveis para o treinamento, como em monitoramento de segurança, detecção de fraudes, e diagnóstico de falhas em sistemas complexos. (RUFF et al., 2018)

B.14 KERNEL DENSITY FUNCTIONS (KDE)

Kernel Density Estimation (KDE) é uma técnica não paramétrica usada para estimar a função de densidade de probabilidade de uma variável aleatória com base em uma amostra de dados. Ao invés de assumir uma distribuição específica, o KDE utiliza uma função *kernel* (como a gaussiana) para suavizar os dados e criar uma estimativa contínua da densidade. Essencialmente, o KDE soma pequenas "curvas"em torno de cada ponto de dados, resultando em uma função de densidade que captura a forma geral da distribuição dos dados.

Essa técnica é particularmente útil para visualizar a distribuição de dados e identificar padrões ou anomalias, especialmente em situações onde a forma exata da distribuição é desconhecida ou onde se deseja evitar as limitações de métodos paramétricos que assumem distribuições específicas. O KDE é amplamente utilizado em estatística, aprendizado de máquina, e análise de dados exploratória. (ROSENBLATT, 1956) (AGGARWAL, 2017)

B.15 LOCAL CORRELATION INTEGRAL (LOCI)

Para todos os algoritmos mencionados acima, a escolha de k é uma decisão crucial para o desempenho da detecção. Como já mencionado, não há uma maneira de estimar um bom k com base nos dados. No entanto, o algoritmo de Integral da Correlação Local (LOCI) aborda essa questão usando uma abordagem de maximização. A ideia básica é que todos os valores possíveis de k são utilizados para cada registro e, finalmente, a pontuação máxima é escolhida. Para alcançar esse objetivo, o LOCI define o r-vizinhamento usando um raio r, que é expandido ao longo do tempo. Semelhante ao LOOP, a densidade local também é estimada usando uma distribuição gaussiana meia, mas aqui a quantidade de registros no vizinhamento é utilizada em vez das distâncias. Além disso, a estimativa de densidade local é diferente no LOCI: ela compara dois vizinhamentos de tamanhos diferentes em vez da razão das densidades locais. Um parâmetro 'a' controla a razão entre os diferentes vizinhamentos. Remover o parâmetro crítico k tem um custo. Normalmente, algoritmos de detecção de anomalias baseados em

vizinhos mais próximos têm complexidade computacional de $O(n^2)$ para encontrar os vizinhos mais próximos. Como no LOCI o raio r também precisa ser expandido de uma instância até a mais distante, a complexidade aumenta para $O(n^3)$, o que torna o LOCI muito lento para conjuntos de dados maiores. (GOLDSTEIN; UCHIDA, 2016)

B.16 FULLY CONNECTED AUTOENCODER (AUTOENCODER)

O *Fully Connected AutoEncoder* (AutoEncoder) é uma rede neural projetada para aprender representações compactas dos dados. Ele consiste em duas partes principais: um codificador (*encoder*) e um decodificador (*decoder*). O codificador reduz a dimensionalidade dos dados de entrada, comprimindo a informação em uma representação latente de menor dimensão, enquanto o decodificador reconstrói os dados originais a partir dessa representação comprimida. Durante o treinamento, o AutoEncoder ajusta seus pesos para minimizar a diferença entre os dados de entrada e os dados reconstruídos, o que permite que a rede capture as características mais essenciais dos dados.

AutoEncoders totalmente conectados são caracterizados por suas camadas densamente interligadas, onde cada neurônio em uma camada está conectado a todos os neurônios da camada seguinte. Essa estrutura é amplamente utilizada em tarefas como redução de dimensionalidade, detecção de anomalias e pré-treinamento de outras Redes Neurais. Por sua capacidade de aprender representações eficientes dos dados, *AutoEncoders* são ferramentas poderosas para compressão de dados e extração de características relevantes em diversos contextos de AD (KUO, 2023) (AGGARWAL, 2017).

B.17 VARIATIONAL AUTOENCODER (VAE)

O *Variational AutoEncoder* (VAE) é uma extensão dos *AutoEncoders* tradicionais, onde, em vez de aprender uma única representação determinística dos dados de entrada, o VAE aprende distribuições probabilísticas sobre o espaço latente. Isso significa que, em vez de comprimir os dados em um único ponto no espaço latente, o VAE comprime os dados em uma distribuição gaussiana. Essa abordagem permite capturar melhor a variabilidade dos dados e gerar novas amostras que se assemelham às entradas originais, o que é útil em tarefas como geração de dados e AD.

A estrutura de um VAE consiste em um codificador que mapeia os dados de entrada para

os parâmetros de uma distribuição gaussiana (média e variância) no espaço latente, e um decodificador que reconstrói os dados a partir de amostras retiradas dessa distribuição latente. Durante o treinamento, o VAE é otimizado para minimizar a perda de reconstrução e a divergência de *Kullback-Leibler* entre a distribuição latente aprendida e uma distribuição gaussiana padrão. Esse equilíbrio entre a precisão de reconstrução e a regularização probabilística torna o VAE uma poderosa ferramenta para modelar a estrutura subjacente dos dados e detectar *outliers* (KUO, 2023) (AN; CHO, 2015).

B.18 ANOMALY DETECTION WITH GENERATIVE ADVERSARIAL NETWORKS (ANOGAN)

AnoGAN (*Anomaly Detection with Generative Adversarial Networks*) é uma técnica que utiliza Redes Neurais Profundas para detectar anomalias em dados. Baseia-se em uma arquitetura de *Generative Adversarial Networks* (GAN), onde um gerador cria amostras sintéticas que imitam dados normais, e um discriminador tenta distinguir entre amostras reais e geradas. Após o treinamento em dados normais, o gerador aprende a criar amostras que representam bem a distribuição desses dados.

Na , uma nova amostra é comparada com as amostras geradas pelo gerador. Se a nova amostra for significativamente diferente das amostras geradas, indicando um alto erro de reconstrução, ela é considerada uma anomalia. O AnoGAN é especialmente útil em cenários onde as anomalias são raras e os dados normais têm uma distribuição complexa, como na detecção de fraudes, diagnósticos médicos e monitoramento de sistemas de segurança. (SCHLEGL et al., 2017)

B.19 MULTIPLE-OBJECTIVE GENERATIVE ADVERSARIAL ACTIVE LEARNING (MO_GAAL)

Multiple-Objective Generative Adversarial Active Learning (MOGAL) é uma técnica que combina aprendizado ativo, Redes Generativas Adversariais (GANs) e otimização multiobjetivo para aprimorar a eficiência na rotulagem de dados e no treinamento de modelos de ML. No aprendizado ativo, o modelo seleciona iterativamente as amostras mais informativas para rotular, o que é essencial quando a rotulagem é cara ou demorada. As GANs são usadas para gerar amostras sintéticas que ajudam a explorar melhor o espaço de dados, criando cenários que podem não estar representados no conjunto de dados original.

A abordagem MOGAL otimiza múltiplos objetivos ao selecionar amostras para rotulagem, como minimizar a incerteza do modelo, maximizar a diversidade dos exemplos e equilibrar o custo da rotulagem. Isso permite treinar o modelo de forma mais eficiente com menos dados rotulados, tornando-o especialmente útil em áreas onde a aquisição de dados rotulados é difícil ou dispendiosa, como na detecção de fraudes e diagnósticos médicos. (LIU et al., 2020)

B.20 SINGLE-OBJECTIVE GENERATIVE ADVERSARIAL ACTIVE LEARNING (SO_GAAL)

Single-Objective Generative Adversarial Active Learning (SO-GAAL) é uma técnica de aprendizado ativo que utiliza Redes Generativas Adversariais (GANs) para melhorar a seleção de amostras de dados, focando em otimizar um único objetivo. No aprendizado ativo, o modelo escolhe iterativamente as amostras mais informativas para rotular, a fim de maximizar a performance do modelo com o menor número de amostras rotuladas. As GANs são empregadas para gerar amostras sintéticas que ajudam a explorar o espaço de dados, permitindo ao modelo aprender de uma variedade maior de exemplos, inclusive aqueles que são raros no conjunto de dados original.

Ao contrário de abordagens multiobjetivo, o SO-GAAL se concentra em otimizar um único critério, como minimizar a incerteza do modelo ou maximizar a representatividade das amostras selecionadas. Esse foco torna o processo de seleção mais direcionado e potencialmente mais simples, sendo especialmente útil em situações onde o objetivo principal é maximizar a eficiência na rotulagem de dados e melhorar a precisão do modelo com o menor esforço possível. (LIU et al., 2020)

B.21 PROBABILISTIC MIXTURE MODELING (GMM)

A Modelagem de Mistura Probabilística é uma abordagem estatística utilizada para modelar um conjunto de dados como uma combinação de várias distribuições probabilísticas, onde cada distribuição representa um subconjunto ou "componente"dos dados. Esse método é particularmente útil em contextos onde os dados são heterogêneos e podem ser melhor representados por múltiplas distribuições subjacentes em vez de uma única. Cada ponto de dado tem uma probabilidade de pertencer a cada uma dessas distribuições, e o modelo de mistura estima esses componentes probabilísticos para capturar a complexidade dos dados. Um dos principais usos da modelagem de mistura é identificar *outliers*, que são dados que apresentam

baixa probabilidade de pertencer a qualquer uma das distribuições modeladas, indicando que eles não se encaixam bem nos padrões normais dos dados.

Na prática, a Modelagem de Mistura Probabilística é frequentemente aplicada em combinação com algoritmos como o *Expectation-Maximization* (EM), que é utilizado para estimar os parâmetros das distribuições de mistura. Uma das vantagens desse modelo é sua flexibilidade, permitindo que ele seja aplicado a uma variedade de formas de dados e distribuídos de maneira não uniforme. No entanto, a modelagem de mistura também enfrenta desafios, como a necessidade de especificar o número de componentes de mistura e a sensibilidade a configurações iniciais, o que pode levar a soluções que refletem apenas ótimos locais. (AGGARWAL, 2017)

B.22 LOCALLY SELECTIVE COMBINATION OF PARALLEL OUTLIER ENSEMBLES (LSCP)

Locally Selective Combination of Parallel Outlier Ensembles (LSCP) é uma técnica de detecção de anomalias que combina os resultados de múltiplos detectores de *outliers* de forma adaptativa e baseada em informações locais dos dados. Em vez de usar um único modelo ou combinar os resultados de todos os modelos de maneira global, o LSCP realiza uma combinação seletiva, escolhendo os detectores mais eficazes para cada região específica dos dados. Isso permite que o método capture nuances locais e melhore a precisão da detecção de *outliers* em áreas com diferentes características.

O LSCP funciona definindo uma vizinhança local para cada ponto de dado e, em seguida, avaliando a performance dos detectores de *outliers* dentro dessa vizinhança. Os detectores que se saem melhor nessa região são combinados para formar a decisão final sobre se um ponto é um *outlier*. Essa abordagem torna o LSCP particularmente eficaz em conjuntos de dados complexos e heterogêneos, onde as características locais dos dados variam significativamente, como em sistemas de detecção de fraudes ou em grandes volumes de dados diversificados. (ZHAO et al.,)

B.23 COOK'S DISTANCE (CD)

é uma métrica utilizada para identificar *outliers* em análises de regressão linear. Ela mede a influência de uma observação individual no ajuste geral do modelo, avaliando o impacto que a remoção de um ponto de dado específico teria sobre os coeficientes de regressão. Em essência, *Cook's Distance* combina informações sobre a distância entre um ponto e os outros

pontos (resíduos) com a alavancagem desse ponto (ou seja, sua posição relativa em relação à média dos valores preditores), fornecendo uma medida única para detectar observações que podem estar distorcendo os resultados do modelo.

Para usar *Cook's Distance* na detecção de *outliers*, calcula-se a métrica para cada observação no conjunto de dados. Um valor de *Cook's Distance* relativamente alto indica que a observação tem uma influência significativa no modelo e, portanto, pode ser considerada um *outlier*. Embora não haja um limite absoluto, valores acima de 1 são frequentemente vistos como indicativos de um *outlier* potencial. *Cook's Distance* é uma ferramenta útil para identificar não apenas *outliers*, mas também pontos de alta influência que podem estar comprometendo a validade do modelo de regressão. (COOK, 1977)