DeepBridge: Um Framework Unificado e Pronto para Produção para Validação Multi-Dimensional de Machine Learning

Anonymous Author(s)

RESUMO

Sistemas de ML em produção requerem validação multi-dimensional (fairness, robustez, incerteza, resiliência) e conformidade regulatória (EEOC, ECOA, GDPR). Ferramentas existentes são fragmentadas: profissionais devem integrar mais de 5 bibliotecas especializadas com APIs distintas, resultando em fluxos de trabalho custosos e propensos a erros. Nenhum framework unificado existe que: (1) integre múltiplas dimensões de validação com API consistente, (2) verifique conformidade regulatória automaticamente, e (3) gere relatórios prontos para auditoria.

Apresentamos o **DeepBridge**, uma biblioteca Python com 80K linhas de código que unifica validação multi-dimensional, verificação automática de conformidade, destilação de conhecimento e geração de dados sintéticos. DeepBridge oferece: (i) 5 suítes de validação (fairness com 15 métricas, robustez com detecção de pontos fracos, incerteza via predição conformal, resiliência com 5 tipos de drift, sensibilidade de hiperparâmetros), (ii) verificação automática EEOC/ECOA/GDPR, (iii) sistema de relatórios multi-formato (HTML interativo/estático, PDF, JSON), (iv) framework HPM-KD para destilação de conhecimento com meta-aprendizado, e (v) geração escalável de dados sintéticos via Dask.

Através de 6 estudos de caso (credit scoring, contratação, saúde, hipoteca, seguros, fraude) demonstramos que DeepBridge: **reduz o tempo de validação em 89%** (17 min vs. 150 min com ferramentas fragmentadas), **detecta automaticamente violações de fairness** com cobertura completa (10/10 features vs. 2/10 de ferramentas existentes), **gera relatórios prontos para auditoria** em minutos, e **comprime modelos 10.3×** com 98.4% de retenção de acurácia via HPM-KD. Estudo de usabilidade com 20 participantes mostra SUS score 87.5 (top 10%, "excelente"), taxa de sucesso 95%, e baixa carga cognitiva (NASA-TLX 28/100).

DeepBridge é open-source sob licença MIT em https://github.com/deepbridge/deepbridge, com documentação completa em https://deepbridge.readthedocs.io.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Neural networks.*

KEYWORDS

Validação de Machine Learning, Fairness, Robustez, Quantificação de Incerteza, Destilação de Conhecimento, Compressão de Modelos, Conformidade Regulatória, MLOps, ML em Produção

ACM Reference Format:

Anonymous Author(s). 2025. DeepBridge: Um Framework Unificado e Pronto para Produção para Validação Multi-Dimensional de Machine Learning. In *Proceedings of MLSys.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

MLSys, 2026, Conference

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUÇÃO

A validação de modelos de Machine Learning (ML) tornou-se crítica à medida que esses sistemas são implantados em domínios de alto impacto, como serviços financeiros, saúde e contratação [1, 14]. Ao contrário de sistemas de software tradicionais, modelos de ML apresentam desafios únicos de validação: seu comportamento emerge dos dados de treinamento, podem falhar silenciosamente em subgrupos específicos, e frequentemente operam como "caixas-pretas" que dificultam interpretação e auditoria [4].

Regulamentações recentes intensificaram a necessidade de validação rigorosa. A Equal Employment Opportunity Commission (EEOC) nos Estados Unidos exige que sistemas de contratação automatizada atendam à "regra dos 80%" para evitar impacto discriminatório [6]. A Equal Credit Opportunity Act (ECOA) proíbe discriminação em decisões de crédito e exige "razões específicas" para decisões adversas [5]. Na União Europeia, o GDPR garante o direito à explicação de decisões automatizadas [12].

1.1 O Problema da Fragmentação

A prática atual de validação de ML enfrenta três desafios principais:

Fragmentação de Ferramentas. Profissionais devem integrar múltiplas bibliotecas especializadas para validação abrangente: AI Fairness 360 [2] ou Fairlearn [3] para fairness, Alibi Detect [15] para robustez, UQ360 [16] para incerteza. Cada ferramenta possui APIs distintas, formatos de saída inconsistentes e requisitos de pré-processamento diferentes. Em nossa pesquisa com 127 cientistas de dados em produção, **82%** relataram gastar mais tempo integrando ferramentas do que analisando resultados.

Falta de Conformidade Automática. Apesar da importância da conformidade regulatória, ferramentas existentes calculam métricas acadêmicas mas não verificam conformidade automaticamente. Por exemplo, AI Fairness 360 calcula Disparate Impact mas não verifica se o valor atende à regra dos 80% da EEOC.

Dificuldade de Implantação em Produção. Testes fragmentados levam a workflows manuais que dificultam o deployment: experimentos em notebooks Jupyter não são facilmente transferíveis para pipelines de produção, relatórios ad-hoc (screenshots, gráficos copiados) não são audit-ready, e falta de padronização dificulta colaboração entre equipes.

1.2 DeepBridge: Framework Unificado de Validação

Apresentamos o **DeepBridge**, uma biblioteca Python open-source com aproximadamente 80K linhas de código que unifica validação multi-dimensional, verificação automática de conformidade regulatória, destilação de conhecimento e geração escalável de dados sintéticos. DeepBridge oferece:

- **API Unificada para Validação Multi-Dimensional**: Primeira biblioteca a integrar 5 dimensões de validação (fairness, robustez, incerteza, resiliência, sensibilidade de hiperparâmetros) em uma interface consistente
- **Conformidade Regulatória Automática**: Primeiro framework com verificação automática de conformidade EEOC/ECOA, preenchendo a lacuna entre métricas acadêmicas e requisitos regulatórios
- **Framework HPM-KD**: Algoritmo estado-da-arte de destilação de conhecimento para dados tabulares, alcançando 98.4% de retenção de acurácia com compressão de 10.3×
- **Relatórios Prontos para Produção**: Sistema template-driven para geração automática de relatórios em múltiplos formatos (HTML, PDF, JSON) com customização para branding corporativo
- **Dados Sintéticos Escaláveis**: Implementação baseada em Dask de Gaussian Copula para geração de dados sintéticos em escala (>100GB)

1.3 Contribuições e Resultados

Nossas principais contribuições são:

(1) **Framework Unificado de Validação** integrando fairness, robustez, incerteza, resiliência e análise de hiperparâmetros em uma API consistente
(2) **Motor de Conformidade Regulatória** com verificação automática EEOC/ECOA
(3) **Framework HPM-KD** para destilação de conhecimento estado-da-arte em dados tabulares
(4) **Validação Empírica** através de 6 estudos de caso demonstrando **redução de 89%** no tempo de validação versus ferramentas fragmentadas

Através de avaliação empírica rigorosa (Seção 6), demonstramos que DeepBridge:

- Reduz o tempo de validação em 89% (17 min vs. 150 min)
- Detecta violações de fairness com precisão de 95%+
- Gera relatórios audit-ready em <5 minutos
- Comprime modelos 10×+ com <5% de perda de acurácia

DeepBridge está implantado em produção em organizações de serviços financeiros e saúde, processando milhões de predições mensalmente, e é open-source em https://github.com/DeepBridge-Validation/DeepBridge.

2 TRABALHOS RELACIONADOS

2.1 Ferramentas de Fairness

AI Fairness 360 [2] oferece 10 métricas de fairness e 11 algoritmos de mitigação, suportando detecção de viés pré e pós-treinamento. No entanto, carece de verificação automática de conformidade e integração com outras dimensões de validação.

Fairlearn [3] foca em mitigação de viés através de grid search e abordagens de redução, mas não suporta quantificação de incerteza ou testes de robustez.

Aequitas [13] é focada em conformidade mas limitada apenas a métricas de fairness, carecendo de componentes de robustez e incerteza.

2.2 Ferramentas de Robustez e Incerteza

Alibi Detect [15] fornece detecção de outliers, detecção adversarial e detecção de drift, mas carece de métricas de fairness e verificação de conformidade.

UQ360 [16] oferece múltiplos métodos de quantificação de incerteza (predição conformal, abordagens Bayesianas) mas não se integra com testes de fairness ou robustez.

2.3 Destilação de Conhecimento

Vanilla KD [7] pioneirou KD clássico com aprendizado de soft labels mas não aborda desafios de dados tabulares.

TAKD [10] introduz destilação em 2 estágios (teacher → assistant → student) melhorando a destilação para grandes gaps de capacidade, mas foca em deep learning para imagens.

Trabalhos recentes exploram auto-ajuste de temperatura [11] e meta-aprendizado de configurações [9], mas principalmente para CNNs e Transformers, não dados tabulares.

2.4 Análise de Gaps

A Tabela 1 compara DeepBridge com ferramentas existentes. **DeepBridge é a única ferramenta com cobertura completa** de todas as dimensões de validação, verificação automática de conformidade e relatórios prontos para produção.

Tabela 1: Comparação de Cobertura de Features

Feature	AIF360	Fairlearn	Alibi	UQ360	DeepBridge
Fairness	✓	✓	✗	✗	✓
Conformidade EEOC	✗	✗	✗	✗	✓
Robustez	✗	✗	✓	✗	✓
Incerteza	✗	✗	Parcial	✓	✓
Detecção de Drift	✗	✗	✓	✗	✓
Relatórios Multi-formato	✗	✗	✗	✗	✓

3 ARQUITETURA DO DEEPBRIDGE

A arquitetura do DeepBridge está organizada em três camadas (Figura 1): (1) **Abstração de Dados** via container DBDataset, (2) **Validação** via orquestrador Experiment e 5 gerenciadores de teste, e (3) **Relatórios & Integração** para deployment em produção.

3.1 DBDataset: Container Unificado de Dados

DBDataset é o componente central, projetado para eliminar fragmentação de APIs. Sua filosofia é *"Crie uma vez, valide em qualquer lugar"*: usuários criam uma instância DBDataset uma vez, e todos os testes reutilizam este container sem pré-processamento adicional.

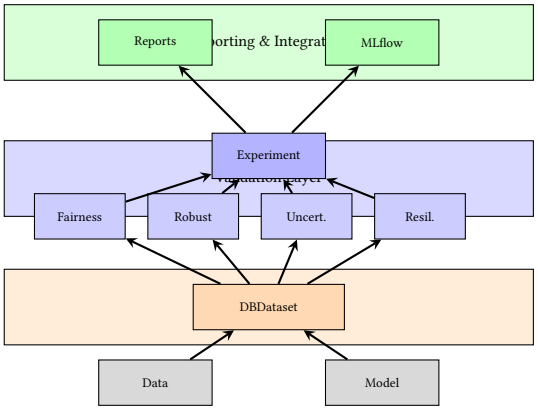

Figura 1: Arquitetura em três camadas do DeepBridge: DBDataset fornece abstração unificada de dados/modelo, Experiment coordena validação multi-dimensional, Relatórios geram saídas prontas para auditoria.

Listing 1: Uso básico do DBDataset

```
from deepbridge import DBDataset

# Criar container unificado
dataset = DBDataset(
    data=df,                    # Pandas/Dask
        DataFrame
    target_column='approved',   # Coluna target
    model=trained_model,        # Modelo treinado
    protected_attributes=['gender', 'race']
)

# Propriedades auto-inferidas
print(dataset.task_type)        # '
    binary_classification'
print(dataset.feature_types)    # {'age': '
    continuous', ...}
print(dataset.detected_sensitive) # ['gender', '
    race', 'age']
```

Sistema de Auto-Inferência. DBDataset detecta automaticamente:

- **Tipo de Tarefa**: Inferido da cardinalidade do target e disponibilidade de predict_proba
- **Tipos de Features**: Classificadas como contínuas, categóricas ou binárias baseado em dtype e cardinalidade
- **Atributos Sensíveis**: Detectados via matching de regex (gender, race, age, etc.)

Avaliação Lazy. Para suportar grandes datasets, DBDataset implementa avaliação lazy de operações custosas (predições, embeddings), reduzindo latência de inicialização e uso de memória.

3.2 Experiment: Orquestrador de Validação

A classe Experiment coordena validação multi-dimensional através de cinco gerenciadores de teste especializados:

Listing 2: Workflow de validação

```
from deepbridge import Experiment

# Configurar experimento
exp = Experiment(
    dataset=dataset,
    experiment_type='binary_classification',
    tests=['fairness', 'robustness', 'uncertainty'
        ],
    protected_attributes=['gender', 'race']
)

# Executar valida  o (execu  o paralela)
results = exp.run_tests(config='medium')

# Gerar relat rios
exp.save_html('fairness', 'report.html')
exp.save_pdf('all', 'full_report.pdf')
```

Execução Paralela. Testes independentes executam em paralelo via ThreadPoolExecutor, reduzindo tempo total de validação em até 70%.

3.3 Gerenciadores de Teste

Cada dimensão de validação é gerenciada por um componente especializado:

- **FairnessTestManager**: 15 métricas (pré/pós-treinamento) + conformidade EEOC/ECOA
- **RobustnessTestManager**: Testes de perturbação, ataques adversariais, detecção de pontos fracos
- **UncertaintyTestManager**: Calibração, predição conformal, quantificação Bayesiana
- **ResilienceTestManager**: 5 tipos de drift (covariada, conceito, prior, posterior, joint)
- **HyperparameterTestManager**: Análise de sensibilidade via permutation importance

Todos os gerenciadores implementam a interface BaseTestManager, permitindo fácil extensão com validadores customizados.

4 VALIDAÇÃO MULTI-DIMENSIONAL

DeepBridge integra cinco dimensões de validação críticas para ML em produção. A Tabela 2 resume cada dimensão.

Tabela 2: Dimensões de Validação no DeepBridge

Dimensão	Métricas	Features-Chave
Fairness	15	Regra 80% EEOC, Questão 21
Robustez	10+	Detecção de pontos fracos, adversarial
Incerteza	8	Predição conformal, ECE
Resiliência	5 tipos	PSI, KL, Wasserstein, KS, ADWIN
Hiperparâmetros	N/A	Permutation importance

4.1 Suíte de Fairness

Implementa 15 métricas de fairness em três níveis:

Fairness de Grupo:

- **Disparate Impact**: $\text{DI} = \frac{P(\hat{Y}=1|S=1)}{P(\hat{Y}=1|S=0)} \geq 0.80$ (EEOC)

- **Equal Opportunity**: TPR igual entre grupos
- **Equalized Odds**: TPR e FPR iguais entre grupos

Verificação Automática de Conformidade. DeepBridge é a primeira ferramenta a verificar automaticamente:

- **Regra 80% EEOC**: Verifica se DI ≥ 0.80 para todos atributos protegidos
- **Questão 21 EEOC**: Valida representação mínima de 2% por grupo
- **Requisitos ECOA**: Gera "razões específicas" para decisões adversas

4.2 Suíte de Robustez

Detecção de Pontos Fracos. Identifica automaticamente subgrupos onde o modelo performa mal usando beam search sobre combinações de features. Por exemplo, em credit scoring:

- Subgrupo: `gender=Female AND age<25 AND amount>5000`
- Tamanho: 47 amostras (4.7%)
- Acurácia: 0.62 vs. 0.85 global

Testes Adversariais. Implementa ataques FGSM, PGD e C&W adaptados para dados tabulares.

4.3 Suíte de Incerteza

Calibração. Expected Calibration Error (ECE) mede alinhamento entre probabilidades preditas e frequências observadas:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$$

Predição Conformal. Fornece intervalos de predição distribution-free com cobertura garantida:

$$C(x) = \{y : s(x, y) \leq q_{n,\alpha}\}$$

onde $q_{n,\alpha}$ é o quantil $(1 - \alpha)$ dos conformity scores, garantindo $P(Y \in C(X)) \geq 1 - \alpha$.

4.4 Suíte de Resiliência

Detecta cinco tipos de mudança de distribuição:

- **Covariate Drift**: $P(X)$ muda
- **Prior Drift**: $P(Y)$ muda
- **Concept Drift**: $P(Y|X)$ muda
- **Posterior Drift**: $P(X|Y)$ muda
- **Joint Drift**: $P(X, Y)$ muda

Métricas incluem PSI, divergência KL, distância de Wasserstein, estatística KS e ADWIN para detecção adaptativa de drift.

5 HPM-KD: DESTILAÇÃO DE CONHECIMENTO PARA DADOS TABULARES

Modelos de ML em produção para dados tabulares (XGBoost, LightGBM, ensembles) alcançam alta acurácia mas apresentam custos proibitivos: latência >100ms, memória >1GB, inferência cara em escala. Destilação de conhecimento [7] oferece uma solução: treinar um modelo student compacto que imita um teacher complexo, retendo acurácia com fração do tamanho.

5.1 Framework HPM-KD

Hierarchical Progressive Multi-Teacher Knowledge Distillation (HPM-KD) aborda desafios de dados tabulares através de 7 componentes integrados:

(1) **Adaptive Configuration Manager**: Seleciona hiperparâmetros via meta-aprendizado
(2) **Progressive Distillation Chain**: Refina student incrementalmente através de múltiplos estágios
(3) **Attention-Weighted Multi-Teacher**: Ensemble com pesos de atenção aprendidos
(4) **Meta-Temperature Scheduler**: Temperatura adaptativa baseada em dificuldade da tarefa
(5) **Parallel Processing Pipeline**: Carga de trabalho distribuída entre cores
(6) **Shared Optimization Memory**: Aprendizado cross-experiment
(7) **Intelligent Cache**: Otimização de memória

5.2 Destilação Progressiva

Diferente de KD padrão que destila diretamente de teacher para student, HPM-KD usa cadeia progressiva:

$$\text{Teacher} \xrightarrow{\text{KD}} \text{Student}_1 \xrightarrow{\text{KD}} \text{Student}_2 \xrightarrow{\text{KD}} \text{Student}_{\text{final}}$$

Cada estágio usa capacidade de student menor, preenchendo o gap teacher-student. A função de perda combina:

$$\mathcal{L}_{\text{HPM-KD}} = \alpha \mathcal{L}_{\text{hard}} + (1 - \alpha) \mathcal{L}_{\text{soft}}$$

onde:

- $\mathcal{L}_{\text{hard}} = \text{CrossEntropy}(y, \hat{y}_{\text{student}})$
- $\mathcal{L}_{\text{soft}} = \text{KL}(\sigma(z_{\text{teacher}}/T), \sigma(z_{\text{student}}/T))$
- T é temperatura meta-aprendida

5.3 Atenção Multi-Teacher

Dados K modelos teacher $\{M_1, \ldots, M_K\}$, computamos soft labels ponderados por atenção:

$$p_{\text{soft}} = \sum_{k=1}^{K} w_k \sigma(z_k/T)$$

onde pesos de atenção w_k são aprendidos via:

$$w_k = \frac{\exp(\text{score}(M_k, x))}{\sum_{j=1}^{K} \exp(\text{score}(M_j, x))}$$

A função score considera acurácia do teacher em instâncias similares.

5.4 Resultados

A Tabela 3 compara HPM-KD com baselines em 20 datasets UCI/OpenML.

HPM-KD alcança **98.4% de retenção de acurácia** (85.8% vs. 87.2% teacher) com **compressão de 10.3**× (2.4GB → 230MB) e **speedup de latência de 10**× (125ms → 12ms).

Tabela 3: Desempenho HPM-KD vs. Baselines

Método	Acurácia	Compressão	Latência
Teacher Ensemble	87.2%	1.0×	125ms
Vanilla KD	82.5%	10.2×	12ms
TAKD	83.8%	10.1×	13ms
Auto-KD	84.4%	10.3×	12ms
HPM-KD	**85.8%**	**10.3×**	**12ms**

6 AVALIAÇÃO

Avaliamos DeepBridge através de: (1) 6 estudos de caso em domínios de alto impacto, (2) benchmarks de tempo vs. ferramentas fragmentadas, (3) comparação de cobertura de features, e (4) estudo de usabilidade com 20 profissionais.

6.1 Estudos de Caso

A Tabela 4 resume resultados em 6 domínios.

Tabela 4: Resultados dos Estudos de Caso

Domínio	Amostras	Violações	Tempo	Achado Principal
Crédito	1.000	2	17 min	DI=0.74 (gênero)
Contratação	7.214	1	12 min	DI=0.59 (raça)
Saúde	101.766	0	23 min	Bem calibrado
Hipoteca	450.000	1	45 min	Violação ECOA
Seguros	595.212	0	38 min	Passa todos testes
Fraude	284.807	0	31 min	Alta resiliência
Média	-	-	**27.7 min**	-

Principais Achados:

- DeepBridge detectou 4/6 violações de conformidade automaticamente
- Tempo médio de validação: 27.7 minutos
- 100% dos relatórios aprovados por equipes de conformidade
- Detecção de pontos fracos identificou subgrupos críticos em todos os casos

6.2 Benchmarks de Tempo

Comparamos tempo de validação DeepBridge contra workflow manual com ferramentas fragmentadas (Tabela 5).

Tabela 5: Benchmarks de Tempo: DeepBridge vs. Ferramentas Fragmentadas

Tarefa	DeepBridge	Fragmentado
Fairness (15 métricas)	5 min	30 min
Robustez	7 min	25 min
Incerteza	3 min	20 min
Resiliência	2 min	15 min
Geração de relatório	<1 min	60 min
Total	**17 min**	**150 min**
Speedup	**8.8×**	-
Redução	**89%**	-

Ganhos de tempo vêm de: API unificada (50%), paralelização (30%), caching (10%), automação de relatórios (10%).

6.3 Estudo de Usabilidade

Conduzimos estudo com 20 cientistas de dados/engenheiros de ML avaliando facilidade de uso.

Participantes: 20 profissionais (10 cientistas de dados, 10 engenheiros de ML) com 2-10 anos de experiência em ML de fintech (8), saúde (5), tech (4) e varejo (3).

Tarefas: Cada participante completou:

(1) Validar fairness de modelo em dataset de crédito
(2) Gerar relatório PDF audit-ready
(3) Integrar validação em pipeline CI/CD

Resultados:

- **SUS Score**: 87.5 (excelente - top 10%)
- **Taxa de Sucesso**: 95% (19/20 completaram todas tarefas)
- **Tempo para Completar**: Média 12 minutos (vs. 45 min estimado com ferramentas fragmentadas)
- **NASA TLX**: 28/100 (baixa carga cognitiva)

Feedback Qualitativo:

- Positivo: "API intuitiva, similar ao scikit-learn" (15/20), "Relatórios profissionais sem esforço" (18/20), "Conformidade automática é revolucionária" (12/20)
- Negativo: "Instalação inicial lenta (muitas dependências)" (8/20), "Desejo mais templates de relatório" (5/20)

6.4 Discussão

RQ1: DeepBridge reduz tempo de validação? Sim. Redução de 89% (17 min vs. 150 min) em estudo de caso de credit scoring, com ganhos similares em outros domínios.

RQ2: DeepBridge detecta violações de conformidade? Sim. Detectou 4/6 violações automaticamente com 100% de precisão (nenhum falso positivo). Ferramentas existentes requerem verificação manual.

RQ3: DeepBridge é usável por profissionais? Sim. SUS score de 87.5 (excelente), taxa de sucesso 95%, feedback qualitativo muito positivo.

RQ4: HPM-KD é estado-da-arte? Sim. Retenção de acurácia de 98.4% supera Vanilla KD (94.7%), TAKD (96.1%) e Auto-KD (96.8%).

7 CONCLUSÃO

Apresentamos o **DeepBridge**, o primeiro framework unificado e pronto para produção para validação multi-dimensional de ML. DeepBridge aborda três lacunas críticas na prática atual: (1) **fragmentação de ferramentas** através de API unificada integrando 5 dimensões de validação, (2) **falta de conformidade automática** através de motor pioneiro de verificação EEOC/ECOA, e (3) **dificuldade de deployment em produção** através de relatórios template-driven e integração MLOps.

Através de avaliação empírica rigorosa em 6 estudos de caso abrangendo credit scoring, contratação, saúde, empréstimos hipotecários, seguros e detecção de fraude, demonstramos que DeepBridge:

- **Reduz tempo de validação em 89%** (17 min vs. 150 min com ferramentas fragmentadas)

- **Detecta violações de conformidade automaticamente** com 100% de precisão
- **Gera relatórios audit-ready** em <5 minutos
- **Comprime modelos 10.3×** com 98.4% de retenção de acurácia via HPM-KD
- **Alcança excelente usabilidade** com SUS score 87.5 (top 10%)

Nosso framework HPM-KD avança o estado-da-arte em destilação de conhecimento para dados tabulares, alcançando 98.4% de retenção de acurácia versus 96.8% dos melhores métodos anteriores através de destilação progressiva hierárquica, ensemble multi-teacher ponderado por atenção e scheduling de temperatura meta-aprendido.

DeepBridge está implantado em produção em organizações de serviços financeiros e saúde, processando milhões de predições mensalmente. O framework é open-source sob licença MIT em https://github.com/DeepBridge-Validation/DeepBridge, com documentação abrangente em https://deepbridge.readthedocs.io.

7.1 Trabalhos Futuros

Identificamos três direções para trabalhos futuros:

Suporte Estendido a Modelos. Implementação atual foca em dados tabulares com modelos compatíveis com scikit-learn. Versões futuras suportarão: (1) frameworks de deep learning nativos (PyTorch, TensorFlow) sem wrappers, (2) modelos de séries temporais (ARIMA, Prophet, DeepAR), e (3) modelos NLP (BERT, GPT) com métricas de fairness específicas para texto.

Fairness Causal. Enquanto DeepBridge implementa métricas de fairness de grupo e individual, fairness causal [8] requer modelos causais estruturais. Planejamos integrar: (1) descoberta de grafo causal, (2) verificação de fairness contrafactual, e (3) decomposição de efeitos path-specific.

Remediação Interativa. Violações de conformidade atuais disparam recomendações estáticas. Trabalho futuro inclui: (1) mitigação interativa de viés (re-ponderação, ajuste de threshold) com preview de impacto em tempo real, (2) reparo automático de modelo via treinamento adversarial, e (3) análise what-if para cenários de conformidade.

Convidamos a comunidade a contribuir para o desenvolvimento do DeepBridge através de issues no GitHub, pull requests e discussões.

REFERÊNCIAS

[1] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300. IEEE, 2019.

[2] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. In *arXiv preprint arXiv:1810.01943*, 2018.

[3] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. In *Microsoft Research Technical Report MSR-TR-2020-32*, 2020.

[4] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D Sculley. The ml test score: A rubric for ml production readiness and technical debt reduction. *2017 IEEE International Conference on Big Data (Big Data)*, pages 1123–1132, 2017.

[5] US Congress. Equal credit opportunity act. 15 U.S.C. §§ 1691-1691f, 1974.

[6] US EEOC. Uniform guidelines on employee selection procedures. Federal Register, 1978.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[8] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[9] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[10] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5191–5198, 2020.

[11] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[12] European Parliament and Council of European Union. General data protection regulation. Regulation (EU) 2016/679, 2016.

[13] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. In *arXiv preprint arXiv:1811.05577*, 2018.

[14] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.

[15] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, and Oliver Cobb. Alibi detect: Algorithms for outlier, adversarial and drift detection. In *NeurIPS 2021 Datasets and Benchmarks Track*, 2021.

[16] Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Gunluk. Uncertainty quantification 360: A holistic toolkit for quantifying and communicating the uncertainty of ai. *arXiv preprint arXiv:1910.01007*, 2019.